ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Συγκριτική ανάλυση και μελέτη ουρών σε cloud περιβάλλον

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΔΗΜΗΤΡΗ ΑΓΓΕΛΙΝΑ

Επιβλέπουσα : Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π.

Αθήνα, Απρίλιος 2016

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Συγκριτική ανάλυση και μελέτη ουρών σε cloud περιβάλλον

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΔΗΜΗΤΡΗ ΑΓΓΕΛΙΝΑ

Επιβλέπουσα : Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 7η Απριλίου 2016.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Θεοδώρα Βαρβαρίγου	Βασίλειος Λούμος	Εμμανουήλ Βαρβαρίγος
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Απρίλιος 2016

(Υπογραφή)

.....

ΔΗΜΤΡΗΣ ΑΓΓΕΛΙΝΑΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η ανάπτυξη των τεχνολογιών πληροφορικής έχει οδηγήσει στη ραγδαία αύξηση της συλλογής τεράστιων όγκων δεδομένων. Αυτά τα δεδομένα ονομάζονται big data. Μαζί με τις ευκαιρίες που προσφέρουν, λόγω της φύσης τους χρειάζεται να αντιμετωπιστούν μία σειρά από προκλήσεις. Τέτοιες προκλήσεις έχουν να κάνουν με την ανάλυση, τη συλλογή, την επιμέλεια, την αναζήτηση, το μοίρασμα, την αποθήκευση, τη μεταφορά, την απεικόνιση και άλλες. Μία τεχνολογία που μπορεί να τις αντιμετωπίσει είναι το cloud computing. Με τον όρο cloud computing αναφερόμαστε σε ένα είδος υπολογισμού βασισμένο στο Internet που παρέχει κοινόχρηστους υπολογιστικούς πόρους και δεδομένα σε υπολογιστές και άλλες συσκευές κατά παραγγελία.

Οι υπηρεσίες cloud είναι πάρα πολλές σε πλήθος και έχουν πολλές διαφορετικές ιδιότητες. Επίσης υπάρχουν πολλοί πάροχοι τέτοιων υπηρεσιών που τις προσφέρουν με διαφορετικά χαρακτηριστικά. Έτσι, για ένα χρήστη, η επιλογή ενός συνόλου υπηρεσιών cloud γίνεται πολύ δύσκολη. Δημιουργείται λοιπόν η ανάγκη ενός marketplace cloud υπηρεσιών. Το marketplace θα πρέπει να είναι σε θέση να καταλήξει στο καταλληλότερο σύνολο υπηρεσιών για το χρήστη.

Ο σκοπός της διπλωματικής εργασίας είναι η δημιουργία, η μελέτη και η βελτίωση ενός τέτοιου marketplace. Υλοποιήθηκε ένα marketplace με τέσσερα βασικά στάδια επεξεργασίας. Αρχικά έγινε μία υλοποίηση του marketplace που επεξεργάζεται τα δεδομένα με σειριακό τρόπο, με το κάθε στάδιο να ξεκινά όταν τελειώνει το προηγούμενο. Στη συνέχεια υλοποιήθηκε ένα βελτιωμένο marketplace με παράλληλη επεξεργασία κάνοντας χρήση ουρών μηνυμάτων, έχοντας όλα τα στάδια να λειτουργούν ταυτόχρονα. Μετά από σύγκριση των δύο αρχιτεκτονικών συμπεραίνεται ότι η παράλληλη βελτιώνει σημαντικά την επίδοση του marketplace.

Λέξεις Κλειδιά: big data, cloud computing, υπηρεσίες cloud, marketplace, σειριακή επεξεργασία, παράλληλη επεξεργασία, ουρές μηνυμάτων, RabbitMQ, AMQP, MongoDB, Python

Abstract

The development of computer technologies has led to the rapid increase of collecting huge volumes of data. These data are called big data. Apart from the opportunities big data offer, there are a number of challenges that has to be faced. These challenges include analysis, collection, search, sharing, storage, transfer, visualization and more. A technology that can deal with these challenges is cloud computing. With the term cloud computing we refer to an Internet based kind of computing that provides shared computing resources and data to computer and other devices, on demand.

There is a very large number of different cloud services and they have various characteristics. Also there are many cloud service providers that offer different features. Thus, the selection of a cloud services set becomes very difficult for a user. Therefore the need for a cloud service marketplace becomes clear. The marketplace should be able to conclude to the cloud service set most suitable for the user.

The purpose of this thesis is the creation, study and improvement of such a marketplace. A marketplace of four basic stages of processing was implemented. Initially, a marketplace which processes data with a serial way was implemented. With each stage begging its process when the previous stage ends its operation. Next an improved marketplace is implemented, with parallel processing, using message queues and having all stages working simultaneously. After comparing the two architectures, it is concluded that the parallel one improves significantly the performance of the marketplace.

Keywords: big data, cloud computing, cloud services, marketplace, serial processing, parallel processing, message queues, RabbitMQ, AMQP, MongoDB, Python

Ευχαριστίες

Πρώτα απ’ όλα, θέλω να ευχαριστήσω την επιβλέπουσα της διπλωματικής εργασίας μου, Καθηγήτρια κ. Θεοδώρα Βαρβαρίγου, για την πολύτιμη βοήθεια και καθοδήγησή της κατά τη διάρκεια της δουλειάς μου.

Επίσης, είμαι ευγνώμων στον υποψήφιο διδάκτορα κ. Βρεττό Μουλό για την προσεκτική ανάγνωση της εργασίας μου και για τις πολύτιμες υποδείξεις τους.

Τέλος, θέλω να εκφράσω την ευγνωμοσύνη στους γονείς μου, Βασίλη και Κωνσταντίνα για την ολόψυχη αγάπη και υποστήριξή τους όλα αυτά τα χρόνια.

Πίνακας περιεχομένων

1

Εισαγωγή

Καθώς η ταχύτητα της αύξησης της πληροφορίας ξεπερνάει το νόμο του Moore, τα big data έχουν τραβήξει τεράστιο ενδιαφέρον από ερευνητές και επιχειρήσεις. Τα big data προβάλουν καινούριες προκλήσεις που έχουν σχέση με τη συλλογή τους, την αποθήκευσή τους, τη διαχείρισή τους και την ανάλυσή τους. Μία από τις κυριότερες τεχνολογίες που βοηθά στην αντιμετώπιση αυτών των προκλήσεων είναι το cloud computing. Παρέχονται στους χρήστες υπολογιστικοί πόροι και δεδομένα μέσω του Internet χωρίς να χρειάζεται να χτίσουν οι ίδιοι κάποια υπολογιστική υποδομή.

Οι υπηρεσίες cloud είναι πολλές σε πλήθος, έχουν πολλά χαρακτηριστικά και προσφέρονται από πολλούς πάροχους. Ένα marketplace για υπηρεσίες cloud διευκολύνει το χρήστη στο να επιλέξει τις κατάλληλες. Αρκετοί πάροχοι έχουν marketplace για τις υπηρεσίες που προσφέρουν. Στην παρούσα εργασία υλοποιείται ένα marketplace υπηρεσιών cloud από πολλούς διαφορετικούς πάροχους. Ο χρήστης εισάγει ένα σύνολο υπηρεσιών που χρειάζεται μαζί με τις απαιτήσεις που έχει και το marketplace επιστρέφει τις βέλτιστες επιλογές. Υλοποιείται ένα marketplace με αρχιτεκτονική σειριακής επεξεργασίας και για τη βελτίωσή του υλοποιείται ξανά με αρχιτεκτονική παράλληλης επεξεργασίας με χρήση ουρών μηνυμάτων.

Το πρώτο κεφάλαιο της διπλωματική είναι το παρόν και αποτελεί μία εισαγωγή. Το δεύτερο κεφάλαιο αναφέρεται στα big data. Το τρίτο κεφάλαιο περιγράφει το cloud computing. Το τέταρτο κεφάλαιο αναφέρεται στις ουρές μηνυμάτων. Στο πέμπτο κεφάλαιο περιγράφεται

αναλυτικά το πρόβλημα του marketplace υπηρεσιών cloud. Περιγράφονται τα συστατικά που θα χρησιμοποιηθούν και αναλύεται η αρχιτεκτονική του συστήματος. Στο έκτο κεφάλαιο γίνεται ανάλυση των δύο διαφορετικών αρχιτεκτονικών και περιγραφή των παραμέτρων χρήσης για το πρόβλημα που δομήθηκε καθώς και του κώδικα. Τέλος, στο έβδομο κεφάλαιο παρατίθενται τα στατιστικά και μοντελιστικά αποτελέσματα. Γίνεται περιγραφή των εικόνων, διαγραμμάτων και μετρήσεων. Παρουσιάζονται τα συγκριτικά αποτελέσματα και τα συμπεράσματα.

2

Big Data

Με τον όρο big data εννοούμε σύνολα δεδομένων τα οποία είναι τόσο μεγάλα ή πολύπλοκα που οι παραδοσιακές εφαρμογές επεξεργασίας δεδομένων είναι ανεπαρκής. Στις προκλήσεις των big data περιλαμβάνεται η ανάλυση, η συλλογή, η επιμέλεια, η αναζήτηση, το μοίρασμα, η αποθήκευση, η μεταφορά, η απεικόνιση, το querying και οι πληροφορίες ιδιωτικότητας. Ο όρος συχνά αναφέρεται απλά στη χρήση αναλύσεων πρόβλεψης ή συγκεκριμένες άλλες προχωρημένες μεθόδους εξαγωγής αξίας από τα δεδομένα. Σπάνια αναφέρεται σε ένα συγκεκριμένο μέγεθος ενός συνόλου δεδομένων. Η ακρίβεια στα big data μπορεί να οδηγήσει στη δημιουργία αποφάσεων με περισσότερη σιγουριά και ως εκ τούτου καλύτερες αποφάσεις μπορούν να έχουν σαν αποτέλεσμα μεγαλύτερη αποτελεσματικότητα, μείωση του κόστους και περιορισμένο ρίσκο.

Τα σύνολα δεδομένων αυξάνονται ραγδαίως επειδή συλλέγονται αυξανόμενα από φθηνά και πολλά μέσα. Τέτοια μέσα είναι κινητές συσκευές information-sensing, απομακρυσμένοι αισθητήρες, καταγραφές λογισμικού, κάμερες, μικρόφωνα, radio-frequency identification readers και ασύρματα αισθητήρια δίκτυα.

Συστήματα διαχείρισης συσχετιστικών βάσεων δεδομένων και πακέτα στατιστικών και οραματισμού συχνά αντιμετωπίζουν δυσκολία στη διαχείριση των bid data. Η δουλειά αντ' ατού απαιτεί παράλληλο λογισμικό να τρέχει σε δεκάδες, εκατοντάδες ή ακόμα και χιλιάδες servers. Τι θεωρείται big data διαφέρει αναλόγως των δυνατοτήτων των χρηστών και των εργαλείων και οι δυνατότητες επέκτασης κάνουν τα big data ένα κινούμενο στόχο. Για

κάποιους οργανισμούς το να αντιμετωπίζουν εκατοντάδες gigabytes δεδομένων για πρώτη φορά μπορεί να πυροδοτήσει μια ανάγκη για αναθεώρηση των επιλογών διαχείρισης δεδομένων. Για άλλους μπορεί να χρειαστεί δεκάδες ή εκατοντάδες terabytes δεδομένων πριν το μέγεθος των δεδομένων να γίνει σημαντικός παράγοντας για θεώρηση.

2.1 Ορισμός και ιδιότητες

Τα big data είναι μια αφηρημένη έννοια. Εκτός από τις μεγάλες μάζες δεδομένων, έχει επίσης και άλλες ιδιότητες, οι οποίες καθορίζουν τη διαφορά μεταξύ αυτών και των μεγάλων δεδομένων ή τον πολύ μεγάλων δεδομένων. Επί του παρόντος, παρόλο που η σημασία των big data έχει ευρέως αναγνωριστεί, οι άνθρωποι ακόμα έχουν διαφορετικές απόψεις για τον ορισμό τους. Γενικά, ο όρος big data σημαίνει σύνολα δεδομένων που δε μπορούν να αντιληφθούν, αποκτηθούν, διαχειριστούν και επεξεργαστούν από παραδοσιακά IT και software/hardware εργαλεία μέσα σε αποδεκτό χρόνο. Λόγω διαφορετικών ανησυχιών, επιστημονικές και τεχνολογικές επιχειρήσεις, ερευνητές, αναλυτές δεδομένων και τεχνικοί έχουν διαφορετικούς ορισμούς για τα big data. Οι ακόλουθοι ορισμοί μπορούν να βοηθήσουν στην καλύτερη κατανόηση της βαθυστόχαστης κοινωνικής, οικονομικής και τεχνολογικής έννοιας των big data.

Το 2010, ο οργανισμός Apache Hadoop όρισε τα big data ως σύνολα δεδομένων που δε μπορούν να συλληφθούν, διαχειριστούν και επεξεργαστούν από γενικούς υπολογιστές μέσα σε ένα αποδεκτό εύρος. Τα big data σημαίνουν τέτοια σύνολα δεδομένων που δε μπορούν να αποκτηθούν, αποθηκευτούν και διαχειριστούν από κλασικό λογισμικό βάσεων δεδομένων. Αυτός ο ορισμός περιλαμβάνει δύο έννοιες. Πρώτον, ο όγκος των συνόλων δεδομένων που συνιστούν το μέτρο για τα big data αλλάζει και μπορεί να αυξάνεται με τον καιρό ή με τις τεχνολογικές προοδεύσεις. Δεύτερον, ο όγκος των συνόλων δεδομένων που συνιστούν το μέτρο για τα big data σε διάφορες εφαρμογές είναι διαφορετικός για την καθεμία από αυτές. Σήμερα, τα big data γενικά κυμαίνονται από μερικά TB σε μερικά PB. Ο όγκος ενός συνόλου δεδομένων δεν είναι το μόνο κριτήριο για τα big data. Η ολοένα αυξανόμενη κλίμακα των δεδομένων και η διαχείρισή της που δε μπορεί να γίνει με παραδοσιακές τεχνολογίες βάσεων δεδομένων είναι τα επόμενα χαρακτηριστικά κλειδιά.

2.2 Το μοντέλο 3V

Στην πραγματικότητα, τα big data έχουν οριστεί από το 2001. Ο Doug Laney, ένας αναλυτής του META όρισε σε μία ερευνητική αναφορά τις προκλήσεις και τις ευκαιρίες που επέφεραν τα αυξημένα δεδομένα με ένα μοντέλο με 3 V's, δηλαδή την αύξηση του Volume (όγκος), Velocity (ταχύτητα) και του Variety (ποικιλία). Παρόλο που ένα τέτοιο μοντέλο δεν είχε

αρχικά χρησιμοποιηθεί για να ορίσει τα big data, η Gartner και πολλές άλλες επιχειρήσεις όπως η IBM και κάποια ερευνητικά τμήματα της Microsoft χρησιμοποίησαν το 3Vs μοντέλο για να περιγράψουν τα big data μέσα στα επόμενα 10 χρόνια. Στο 3Vs μοντέλο, το Volume σημαίνει ότι με δημιουργία και τη συλλογή μεγάλων μαζών δεδομένων, η κλίμακα των δεδομένων γίνεται ολοένα και περισσότερο μεγάλη. Το Velocity σημαίνει τα χρονοδιαγράμματα των big data, πιο ειδικά, η συλλογή και η ανάλυση, πρέπει να γίνουν γρήγορα και έγκαιρα ώστε να μεγιστοποιηθεί η χρήση της εμπορικής αξίας των big data. Το Variety υποδεικνύει τους διάφορους τύπους των δεδομένων, που περιλαμβάνουν ημιδομημένα και αδόμητα δεδομένα όπως ήχος, βίντεο, ιστοσελίδες και κείμενο καθώς και παραδοσιακά δομημένα δεδομένα.

2.3 Το μοντέλο 4V

Ωστόσο, άλλοι έχουν διαφορετικές απόψεις, συμπεριλαμβανομένων και της IDC, μία από τις ηγετικές εταιρείες με επιρροή στα big data και στα ερευνητικά πεδία τους. Το 2011, μία αναφορά της IDC όρισε τα big data σαν τεχνολογίες που περιγράφουν μια καινούρια γενιά τεχνολογιών και αρχιτεκτονικών, σχεδιασμένες να εξάγουν οικονομικά αξία από πολύ μεγάλους όγκους δεδομένων σε μία ευρεία ποικιλία δεδομένων, με το να χρησιμοποιούν υψηλής ταχύτητας συλλογή, ανακάλυψη και ανάλυση. Με αυτό τον ορισμό, τα χαρακτηριστικά των big data μπορούν να συνοψισθούν σε 4V's, δηλαδή Volume(μεγάλος όγκος), Variety(ποικίλες λεπτομέρειες), Velocity(ταχεία δημιουργία) και Value(τεράστια αξία αλλά πολύ μικρή πυκνότητα). Αυτός ο ορισμός αναγνωρίστηκε ευρέως αφού επισημαίνει τη σημασία και την αναγκαιότητα των big data, για παράδειγμα εξερευνεί τις τεράστιες κρυφές αξίες. Αυτός ο ορισμός υποδεικνύει το πιο κρίσιμο πρόβλημα των big data, το οποίο είναι πώς να ανακαλύψεις αξίες από σύνολα δεδομένων με τεράστια κλίμακα, διαφόρων τύπων και ταχύτατης δημιουργίας.

Επιπρόσθετα, το NIST ορίζει τα big data ως τα δεδομένα που ο όγκος τους, η ταχύτητα απόκτησής τους ή η αναπαράστασή τους περιορίζει την ικανότητα χρησιμοποίησης παραδοσιακών συσχετιστικών μεθόδων για τη διεξαγωγή αποτελεσματικής ανάλυσης. Υποδεικνύει ότι πρέπει να αναπτυχθούν και να χρησιμοποιηθούν αποτελεσματικοί μέθοδοι ή τεχνολογίες για να αναλυθούν και να επεξεργαστούν τα big data.[1]

2.4 Το μοντέλο 5V

Με την ενσωμάτωση ανόμοιων συστημάτων δεδομένων εισάγεται κα το 5ο V, το Veracity. Δηλαδή η ορθότητα και η ακρίβεια της πληροφορίας. Πίσω από κάθε πρακτική διαχείρισης κρύβεται η θεωρία της ποιότητας των δεδομένων, της διακυβέρνησής τους και της

διαχείρισης των μέτα-δεδομένων, μαζί με θεωρήσεις για ιδιωτικότητα και νομικά ζητήματα. Τα big data χρειάζεται να ενσωματωθούν σε ολόκληρο το τοπίο της πληροφορίας και όχι να βλέπονται σαν μία ανεξάρτητη προσπάθεια ή ένα κρυφό πρόγραμμα που γίνεται από κάποιους ειδικούς σε αυτά.

Έτσι συνοψίζοντας το 5Vs μοντέλο έχουμε τα παρακάτω:

Volume. Η ποσότητα των δημιουργημένων και αποθηκευμένων δεδομένων. Το μέγεθος των δεδομένων καθορίζει την αξία και τις εν δυνάμει πληροφορίες και αν μπορούν πραγματικά να θεωρηθούν big data ή όχι.

Variety. Ο τύπος και η φύση των δεδομένων. Αυτό βοηθάει στην ανάλυση των δεδομένων και στην αποτελεσματική χρήση της πληροφορίας που έχει σαν αποτέλεσμα.

Velocity. Η ταχύτητα με την οποία τα δεδομένα δημιουργούνται και επεξεργάζονται για να καλυφθούν οι απαιτήσεις και οι προκλήσεις που εμφανίζονται στο μονοπάτι της ανάπτυξης και της εξέλιξης.

Value. Η αξία που δημιουργείται από την απόκτηση, ανάλυση και επεξεργασία των big data.

Veracity. Η ορθότητα ή η αξιοπιστία των δεδομένων. Η ποιότητα των αποκτημένων δεδομένων μπορεί να διαφέρει σημαντικά, επηρεάζοντας την ακριβή ανάλυση. Τα big data και οι σημερινές τεχνολογίες επιτρέπουν την επεξεργασία τέτοιων δεδομένων. Συχνά ο όγκος των δεδομένων αναπληρώνει την έλλειψη ποιότητας ή ακρίβειας. [2][3]

2.5 Ευκαιρίες των Big Data

Πρόσφατα, αρκετές κυβερνητικές υπηρεσίες των Ηνωμένων Πολιτειών, όπως το εθνικό ινστιτούτο υγείας (NIH) και το εθνικό ίδρυμα επιστήμης (NSF), διαβεβαιώνουν ότι οι χρησιμότητες των big data στη λήψη αποφάσεων επηρεάζουν βαθιά τις μελλοντικές τους εξελίξεις. Συνεπώς, προσπαθούν να αναπτύξουν big data τεχνολογίες και τεχνικές ώστε να διευκολύνουν το έργο τους μετά την μεγάλης κλίμακας κυβερνητική πρωτοβουλία για τα big data. Αυτή η πρωτοβουλία είναι πολύ βοηθητική για το χτίσιμο καινούριων δυνατοτήτων για την εκμετάλλευση γνώσης και τη διευκόλυνση της λήψης αποφάσεων.

Είναι γνωστό ότι οι γέφυρες μεταξύ των big data και της κρυμμένης πληροφορίας σε αυτά είναι πολύ σημαντικές σε όλες τις περιοχές της εθνικής προτεραιότητας. Αυτή η πρωτοβουλία επίσης θα θέσει τα θεμέλια για συμπληρωματικές δραστηριότητες στα big data, όπως προγράμματα υποδομής, πλατφόρμες ανάπτυξης και τεχνικές για την επίλυση πολύπλοκων, οδηγούμενων από δεδομένα προβλήματα στις επιστήμες και στη μηχανική. Τέλος θα λειτουργήσουν στην πράξη και θα ωφελήσουν την κοινωνία.

Σύμφωνα με αναφορά από το McKinsey ινστιτούτο, η αποτελεσματική χρήση των big data έχει τα βασικά πλεονεκτήματα του να μεταμορφώνει οικονομίες και να παραδίδει ένα

καινούριο κύμα παραγωγικής ανάπτυξης. Η εκμετάλλευση πολύτιμης πληροφορίας από τα big data θα γίνει η ο βασικός ανταγωνισμός μεταξύ των σημερινών επιχειρήσεων και θα δημιουργήσει καινούριους ανταγωνιστές που θα είναι σε θέση να προσελκύσουν υπαλλήλους οι οποίοι θα έχουν τις κρίσιμες ικανότητες πάνω στα big data. Υπάρχουν πολλά πλεονεκτήματα στον επιχειρηματικό τομέα που μπορούν να αποκτηθούν από τη συλλογή των big data, όπως αυξανόμενη επιχειρηματική αποδοτικότητα, ενημερωτική στρατηγική κατεύθυνση, ανάπτυξη καλύτερων υπηρεσιών για τους πελάτες, αναγνώριση και ανάπτυξη καινούριων προϊόντων και υπηρεσιών, αναγνώριση νέων πελατών και αγορών και άλλα.

2.6 Προκλήσεις των Big Data

Οι ευκαιρίες πάντα ακολουθούνται από προκλήσεις. Από τη μία μεριά, τα big data φέρνουν πολλές ελκυστικές ευκαιρίες. Από την άλλη μεριά, πρέπει να αντιμετωπιστούν πολλές προκλήσεις. Όταν γίνεται χειρισμός προβλημάτων με big data, οι δυσκολίες βρίσκονται στη συλλογή των δεδομένων, στην αποθήκευση, στην αναζήτηση, στο μοίρασμα, στην ανάλυση και στην απεικόνιση. Μία πρόκληση που υπάρχει στην αρχιτεκτονική των υπολογιστών για δεκαετίες, είναι ότι υπάρχει υψηλές επιδόσεις στην επεξεργασία (CPU-heavy) αλλά φτωχές επιδόσεις στην είσοδο/έξοδο (I/O-poor). Αυτή η ανισορροπία στο σύστημα περιορίζει την ανάπτυξη των big data.

Η επίδοση των επεξεργαστών διπλασιάζεται κάθε 18 μήνες σύμφωνα με το νόμο του Moore. Και η επίδοση των σκληρών δίσκων επίσης διπλασιάζεται με τον ίδιο ρυθμό. Όμως, η ταχύτητα των δίσκων έχει ελάχιστα βελτιωθεί την τελευταία δεκαετία. Ως συνέπεια αυτής της ανισορροπίας οι τυχαίες ταχύτητες I/O έχουν βελτιωθεί μέτρια ενώ οι ακολουθιακές ταχύτητες I/O αυξάνονται αργά. Ταυτόχρονα, οι πληροφορίες αυξάνεται με εκθετικό ρυθμό αλλά η εξέλιξη στις μεθόδους επεξεργασίας της πληροφορίας είναι σχετικά πιο αργές. Σε πολλές σημαντικές εφαρμογές των big data, οι τεχνικές και τεχνολογίες αιχμής δε μπορούν ιδεατά να λύσουν το πραγματικά προβλήματα, ειδικά για ανάλυση πραγματικού χρόνου.

Οι προκλήσεις στην ανάλυση των big data περιλαμβάνουν τη μη συνοχή και μη πληρότητα στα δεδομένα, τη σταθερότητα, τα χρονοδιαγράμματα και την ασφάλεια. Σαν προϋπόθεση για την ανάλυση τους, τα δεδομένα πρέπει να είναι καλά δομημένα. Ωστόσο, θεωρώντας ποικίλα σύνολα δεδομένων σε big data προβλήματα, είναι ακόμα μία μεγάλη πρόκληση να προταθούν αποτελεσματικοί τρόποι για αναπαράσταση, πρόσβαση και ανάλυση αδόμητων ή ημιδομημένων δεδομένων σε επόμενες μελέτες. Καθώς τα μεγέθη των συνόλων δεδομένων είναι πολύ μεγάλα, κάποιες φορές μερικά GBs και η προέλευσή τους από ετερογενής πηγές, κάνουν τις σύγχρονες βάσεις δεδομένων ευάλωτες σε ασυνεχή, ελλιπή, με θόρυβο δεδομένα. Έτσι, ένας αριθμός από τεχνικές επεξεργασίας δεδομένων, όπως καθαρισμός δεδομένων,

ενσωμάτωση δεδομένων, μεταμόρφωση και μείωση δεδομένων, εφαρμόζονται για να αφαιρέσουν το θόρυβο και να διορθώσουν τις ασυνέχειες.

2.6.1 Συλλογή και αποθήκευση των δεδομένων

Τα σύνολα δεδομένων αυξάνονται σε μέγεθος γιατί συλλέγονται ολοένα και περισσότερο από πανταχού παρόν κινητές συσκευές με αισθητήρες για πληροφορίες, εναέριες αισθητήριες τεχνολογίες, απομακρυσμένους αισθητήρες, καταγραφές λογισμικού, κάμερες, μικρόφωνα και ούτω καθ' εξής. Κάθε μέρα δημιουργούνται 2.5 πεντακισεκατομμύρια bytes δεδομένων, και αυτός ο αριθμός εξακολουθεί να αυξάνεται εκθετικά. Η παγκόσμια τεχνολογική δυνατότητα αποθήκευσης της πληροφορίας διπλασιάζεται περίπου κάθε τρία χρόνια από τη δεκαετία του 1980. Σε πολλά πεδία, όπως τα οικονομικά και τα ιατρικά δεδομένα συχνά διαγράφονται ακριβώς επειδή δεν υπάρχει αρκετός χώρος για να αποθηκευτούν αυτά τα δεδομένα. Αυτά τα χρήσιμα δεδομένα δημιουργούνται και συλλέγονται με μεγάλο κόστος αλλά τελικά αγνοούνται. Η μαζική αποθήκευση απαιτεί πειραματικές βάσεις δεδομένων, αποθήκευση σε πίνακες για επιστημονικούς υπολογισμούς μεγάλης κλίμακας και λαμβάνονται τεράστια αρχεία εξόδου.

Τα big data έχουν αλλάξει τον τρόπο που συλλέγουμε και αποθηκεύουμε τα δεδομένα, συμπεριλαμβανομένων συσκευές αποθήκευσης δεδομένων, αρχιτεκτονικές αποθήκευσης δεδομένων, μηχανισμοί πρόσβασης δεδομένων. Καθώς χρειάζονται περισσότερα μέσα αποθήκευσης και μεγαλύτερες ταχύτητες I/O για να αντιμετωπιστούν οι προκλήσεις, δεν υπάρχει αμφιβολία ότι θα χρειαστούν μεγάλες καινοτομίες. Πρώτον, η προσβασιμότητα των big data είναι η πρώτη προτεραιότητα της διαδικασίας ανακάλυψης της γνώσης. Τα big data πρέπει να προσπελαύνονται εύκολα και άμεσα για περεταίρω ανάλυση, καταργώντας πλήρως ή εν μέρει τον περιορισμό CPU-heavy αλλά I/O-poor. Επιπρόσθετα, οι υπό ανάπτυξη τεχνολογίες αποθήκευσης, όπως solid state drive (SSD) και phase-change memory (PCM), μπορούν να βοηθήσουν σε αυτές τις δυσκολίες αλλά δεν είναι καθόλου αρκετό. Μία σημαντική αλλαγή είναι επίσης επικείμενη. Αυτή της μεταμορφωτικής αλλαγής των παραδοσιακών υποσυστημάτων εισόδου-εξόδου. Όπως είναι γνωστό, οι HDDs έχουν πολύ πιο αργή τυχαία I/O επίδοση από ότι ακολουθιακή I/O επίδοση και οι μηχανές επεξεργασίας μορφοποιούσαν τα δεδομένα τους και σχεδίαζαν τις επεξεργαστικές μεθόδους αναζήτησης σύμφωνα με αυτόν τον περιορισμό. Αλλά οι HDDs ολοένα και αντικαθίστανται από SSDs σήμερα, και άλλες τεχνολογίες όπως το PCM είναι επίσης στη γωνία. Οι σύγχρονες τεχνολογίες αποθήκευσης δε μπορούν να έχουν την ίδια υψηλή επίδοση και για την τυχαία I/O και για την ακολουθιακή I/O ταυτόχρονα, κάτι που επιβάλει την αναθεώρηση του σχεδιασμού των υποσυστημάτων αποθήκευσης για τα συστήματα επεξεργασίας big data.

Οι πιο συχνά χρησιμοποιούμενες αρχιτεκτονικές αποθήκευσης μέχρι τώρα είναι η direct-attached storage (DAS), network-attached storage (NAS) και storage area network (SAN). Ωστόσο, όλες αυτές οι υπάρχουσες αρχιτεκτονικές έχουν σημαντικά μειονεκτήματα και περιορισμούς όταν έχουν να αντιμετωπίσουν κατανεμημένα συστήματα μεγάλης κλίμακας. Επιθετικός συγχρονισμός και throughput ανά server είναι οι βασικές απαιτήσεις για εφαρμογές σε υπολογιστικά συμπλέγματα υψηλής κλίμακας, και τα σημερινά συστήματα αποθήκευσης στερούνται και τα δύο. Η βελτιστοποίηση της πρόσβασης των δεδομένων είναι ένας δημοφιλής τρόπος για τη βελτίωση της επίδοσης του υπολογισμού δεδομένων, αυτές οι τεχνικές περιλαμβάνουν αντιγραφή δεδομένων, μετακίνηση, κατανομή, και παράλληλη πρόσβαση.

2.6.2 Μετάδοση των δεδομένων

Η αποθήκευση των δεδομένων στο cloud χρησιμοποιείται ευρέως μετά την ανάπτυξη των cloud τεχνολογιών. Είναι γνωστό ότι η χωρητικότητα σε bandwidth του δικτύου είναι το bottleneck στο cloud και στα κατανεμημένα συστήματα, ειδικά όταν ο όγκος της επικοινωνίας είναι πολύ μεγάλος. Από την άλλη μεριά, η αποθήκευση των δεδομένων στο cloud οδηγεί σε προβλήματα ασφάλειας καθώς υπάρχουν απαιτήσεις για έλεγχο της ακεραιότητας των δεδομένων.

2.6.3 Επιμέλεια των δεδομένων

Η επιμέλεια των δεδομένων στοχεύει στην ανακάλυψη και την ανάκτηση των δεδομένων, στη διασφάλιση της ποιότητας των δεδομένων, στην πρόσθεση αξίας, στη διαχείριση, στη διατήρηση και την αναπαράσταση. Τα υπάρχοντα εργαλεία διαχείρισης βάσεων δεδομένων δεν είναι σε θέση να επεξεργαστούν big data που αναπτύσσονται τόσο σε μέγεθος και πολυπλοκότητα. Αυτή η κατάσταση θα συνεχιστεί καθώς τα πλεονεκτήματα από την εκμετάλλευση των big data επιτρέπουν στους ερευνητές να αναλύουν επιχειρηματικές τάσεις, να αποτρέψουν ασθένειες και να πολεμήσουν το έγκλημα. Παρόλο που το μέγεθος των big data συνεχίζει και αυξάνεται εκθετικά, η τωρινή δυνατότητα εργασίας με τα αυτά είναι σχετικά στα χαμηλά επίπεδα των petabytes, exabytes και zettabytes δεδομένων. Η κλασική προσέγγιση διαχείρισης δομημένων δεδομένων περιλαμβάνουν δύο μέρη, ένα είναι ένα σχήμα για αποθήκευση του συνόλου δεδομένων και το άλλο είναι μία συσχετιστική βάση δεδομένων για ανάκτηση των δεδομένων. Για διαχείριση συνόλων δεδομένων μεγάλης κλίμακας με ένα δομημένο τρόπο, τα data warehouses και τα data marts είναι δύο δημοφιλής προσεγγίσεις. Το data warehouse είναι ένα σύστημα συσχετιστικής βάσης δεδομένων που χρησιμοποιείται για την αποθήκευση και ανάλυση των δεδομένων, καθώς και για την αναφορά των αποτελεσμάτων στους χρήστες. Το data mart είναι βασισμένο πάνω σε ένα data

warehouse και διευκολύνει την πρόσβαση και την ανάλυση του data warehouse. Το data warehouse είναι κυρίως υπεύθυνο για την αποθήκευση των δεδομένων που προέρχονται από λειτουργικά συστήματα. Η προ-επεξεργασία των δεδομένων είναι απαραίτητη πριν αποθηκευτούν. Τέτοια προ-επεξεργασία μπορεί να είναι ο καθαρισμός των δεδομένων, η μεταμόρφωση και η κατηγοριοποίηση τους. Μετά την προ-επεξεργασία, τα δεδομένα είναι διαθέσιμα για υψηλότερου επιπέδου συναρτήσεις εξόρυξης δεδομένων. Τα data warehouse και marts είναι βασισμένα σε SQL συστήματα βάσεων δεδομένων.

Οι NoSQL βάσεις δεδομένων είναι μία σύγχρονη προσέγγιση για σχεδιασμό βάσεων δεδομένων για διαχείριση μεγάλου όγκου κατανεμημένων δεδομένων. Οι NoSQL βάσεις δεδομένων δεν αποφεύγουν την SQL. Παρόλο που είναι αλήθεια ότι κάποια NoSQL συστήματα είναι πλήρως μη συσχετιστικά, άλλα απλά αποφεύγουν επιλεγμένες συσχετιστικές λειτουργικότητες.

Για την αποθήκευση και τη διαχείριση αδόμητων δεδομένων ή μη συσχετιστικών δεδομένων, ένα NoSQL σύστημα απασχολεί συγκεκριμένες προσεγγίσεις. Πρώτον, η αποθήκευση και η διαχείριση των δεδομένων χωρίζονται σε δύο ανεξάρτητα μέρη. Αυτό είναι αντίθετο με τα τις συσχετιστικές βάσεις δεδομένων οι οποίες προσπαθούν να ανταποκριθούν στα ζητήματα και στις δύο πλευρές ταυτόχρονα. Αυτός ο σχεδιασμός δίνει στα συστήματα NoSQL βάσεων δεδομένων πολλά πλεονεκτήματα. Στο κομμάτι της αποθήκευσης επικεντρώνεται στην κλιμάκωση της αποθήκευσης δεδομένων με υψηλή επίδοση. Στο κομμάτι της διαχείρισης παρέχει μηχανισμούς πρόσβασης χαμηλού επιπέδου με τους οποίους το έργο της διαχείρισης των δεδομένων μπορεί να υλοποιηθεί στο επίπεδο της εφαρμογής, αντί να υπάρχει λογική διαχείρισης δεδομένων σκορπισμένη στην SQL ή γλώσσες ειδικές για τη διαδικασία αποθήκευσης σε βάσεις δεδομένων. Γι' αυτό το λόγο, τα NoSQL συστήματα είναι πολύ ευέλικτα για μοντελοποίηση δεδομένων και είναι εύκολο στις αναβαθμίσεις αναπτυγμένων εφαρμογών.

Οι περισσότερες NoSQL βάσεις δεδομένων έχουν μια σημαντική ιδιότητα. Είναι συνήθως ανεξάρτητες από σχήμα (schema-free). Όντως, το μεγαλύτερο πλεονέκτημα των schema-free βάσεων δεδομένων είναι ότι επιτρέπουν στις εφαρμογές να τροποποιούν γρήγορα τη δομή των δεδομένων χωρίς να χρειάζεται να ξανά γραφτούν πίνακες της βάσης. Επιπρόσθετα, κατέχει μεγαλύτερη ευελιξία όταν τα δομημένα δεδομένα είναι αποθηκευμένα ετερογενώς. Στο επίπεδο διαχείρισης των δεδομένων, στα δεδομένα επιβάλλεται να είναι ακέραια και έγκυρα.

2.6.4 Ανάλυση των δεδομένων

Η πρώτη εντύπωση των big data είναι ο όγκος τους, οπότε η μεγαλύτερη και πιο σημαντική πρόκληση είναι η κλιμάκωση αντιμετωπίζονται ζητήματα ανάλυσης των big data. Τις

τελευταίες λίγες δεκαετίες οι ερευνητές έδιναν περισσότερη προσοχή σε αλγόριθμους επιταχυνόμενης ανάλυσης για να αντιμετωπίσουν τους αυξανόμενους όγκους δεδομένων και να επιταχύνουν τους επεξεργαστές ακολουθώντας το νόμο του Moore. Για το τελευταίο, είναι απαραίτητο να αναπτυχθούν δειγματοληπτικές, on-line και πολύ-αποφασιστικές μέθοδοι ανάλυσης. Από τη μεριά των αναλυτικών τεχνικών για big data, οι αυξητικοί αλγόριθμοι δεν έχουν καλή ιδιότητα κλιμάκωσης για όλους τους αλγόριθμους εκμάθησης. Καθώς το μέγεθος των δεδομένων κλιμακώνεται πολύ πιο γρήγορα από της ταχύτητες των επεξεργαστών, υπάρχει μία φυσική δραματική μετατόπιση προς την τεχνολογία επεξεργαστών. Παρόλο που η συχνότητα του κύκλου ρολογιού των επεξεργαστών διπλασιάζεται σύμφωνα με το νόμο του Moore, οι ταχύτητες του ρολογιού έχουν μείνει σημαντικά πίσω. Εναλλακτικά, οι επεξεργαστές ενσωματώνονται με αυξημένους αριθμούς από πυρήνες. Αυτή η μετατόπιση στους επεξεργαστές οδηγεί στην ανάπτυξη του παράλληλου υπολογισμού.

Για τις big data εφαρμογές σε πραγματικό χρόνο, όπως πλοήγηση, κοινωνικά δίκτυα, οικονομικά, βιοϊατρική, αστρονομία, έξυπνα συστήματα μεταφοράς και Internet of Things, τα χρονοδιαγράμματα είναι η πρώτη προτεραιότητα. Η εγγύηση της απόκρισης των χρονοδιαγραμμάτων όταν ο όγκος των δεδομένων που πρέπει να επεξεργαστεί είναι πολύ μεγάλος είναι ακόμα μία μεγάλη πρόκληση. Είναι ορθό να ειπωθεί ότι τα big data όχι μόνο έχουν δημιουργήσει πολλές προκλήσεις και έχουν αλλάξει τις κατευθύνσεις της ανάπτυξης του hardware αλλά επίσης και στις αρχιτεκτονικές του λογισμικού.

Η ασφάλεια στα δεδομένα έχει αναδειχθεί με μεγάλα ζητήματα. Σημαντικά προβλήματα ασφάλειας περιλαμβάνουν την προστασία ασφάλειας δεδομένων, την προστασία πνευματικών δικαιωμάτων, την προστασία της προσωπικής ιδιωτικότητας και την προστασία εμπορικών μυστικών και οικονομικών πληροφοριών. Οι περισσότερες αναπτυγμένες και υπό ανάπτυξη χώρες έχουν ήδη νόμους σχετικούς με την προστασία των δεδομένων για να ενισχύσουν την ασφάλεια. Για εφαρμογές σχετικές με big data, τα προβλήματα ασφάλειας δεδομένων είναι πιο πολύπλοκα για διάφορους λόγους. Πρώτον, το μέγεθος των big data είναι πάρα πολύ μεγάλο, διοχετεύοντας τις προσεγγίσεις για προστασία. Δεύτερον, οδηγεί επίσης σε πολύ βαρύτερο φόρτο εργασίας της ασφάλειας. Διαφορετικά, τα περισσότερα big data αποθηκεύονται με ένα κατανεμημένο τρόπο και οι απειλές από τα δίκτυα μπορούν επίσης να επιβαρύνουν τα προβλήματα.

2.6.5 Απεικόνιση των δεδομένων

Το κύριο αντικείμενο της απεικόνισης των δεδομένων είναι η αναπαράσταση της γνώσης πιο διαισθητικά και αποδοτικά με τη χρήση διαφορετικών γραφημάτων. Για την εύκολη μετάδοση της πληροφορίας με το να παρέχεται η κρυμμένη γνώση στα πολύπλοκα και μεγάλης κλίμακας σύνολα δεδομένων, είναι απαραίτητες η καλαίσθητη μορφή τους και η

λειτουργικότητα. Η πληροφορία που έχει γίνει αφηρημένη σε κάποιες σχηματικές μορφές, συμπεριλαμβανομένων ιδιοτήτων ή μεταβλητών των μονάδων της πληροφορίας, είναι επίσης πολύτιμη για ανάλυση δεδομένων. Αυτό ο τρόπος είναι πολύ πιο διορατικός από πολύπλοκες προσεγγίσεις.

Για εφαρμογές big data, είναι ιδιαίτερα δύσκολο να διεξαχθεί η απεικόνιση των δεδομένων εξ αιτίας του πολύ μεγάλου μεγέθους και της υψηλής διάστασης των big data. Ωστόσο, τα σύγχρονα εργαλεία απεικόνισης των big data έχουν κυρίως φτωχές επιδόσεις στις λειτουργικότητες, την κλιμάκωση και τον χρόνο απόκρισης. Η αβεβαιότητα μπορεί να οδηγήσει σε μεγάλες προκλήσεις αναφορικά με την αποτελεσματική απεικόνιση και μπορούν να φανερωθούν σε οποιοδήποτε στάδιο της απεικονιστικής αναλυτικής διαδικασίας. Καινούρια frameworks για τη μοντελοποίηση της αβεβαιότητας και την κατηγοριοποίηση της εξέλιξης της αβεβαιότητας της πληροφορίας είναι πολύ απαραίτητη για την αναλυτική διαδικασία.

Η έλλειψη ταλέντων θα αποτελέσει σημαντικό περιορισμό στη σύλληψη αξίας από τα big data. Τα big data εκτιμάται ότι πολύ γρήγορα θα αποτελέσουν καθοριστικό παράγοντα ανταγωνισμού σε πολλούς τομείς. Παρόλα αυτά, αυτός ο κλάδος απαιτεί βαθιές αναλυτικές θέσεις εργασίας αλλά τα big data θα ξεπεράσουν τις προμήθειες που παράγονται στις σημερινές τάσεις στις θέσεις εργασίας. Επιπλέον, αυτού του είδους το ανθρώπινο δυναμικό είναι πιο δύσκολο να εκπαιδευτεί. Συνήθως χρειάζονται πολλά χρόνια για να εκπαιδευτούν big data αναλυτές οι οποίοι πρέπει να έχουν ουσιαστικές μαθηματικές ικανότητες και σχετικά επαγγελματική γνώση.

3

Cloud Computing

Με το cloud computing αναφερόμαστε σε ένα είδος υπολογισμού βασισμένο στο Internet που παρέχει κοινόχρηστους υπολογιστικούς πόρους και δεδομένα σε υπολογιστές και άλλες συσκευές κατά παραγγελία. Οι λύσεις για cloud computing και αποθήκευση παρέχουν στους χρήστες και τις επιχειρήσεις ποικίλες δυνατότητες για αποθήκευση και επεξεργασία των δεδομένων τους σε τρίτα κέντρα δεδομένων. [5] Βασίζεται στην κοινή χρήση των πόρων με στόχο να επιτυγχάνεται συνοχή και οικονομία κλίμακας, παρόμοια με μια υπηρεσία (όπως το ηλεκτρικό δίκτυο) πάνω σε δίκτυο. Θεμέλια του cloud computing είναι η ευρύτερη έννοια της συγκεντρωμένης υποδομής και κοινόχρηστων υπηρεσιών.

Η παρούσα διαθεσιμότητα σε δίκτυα υψηλής χωρητικότητας, υπολογιστές με μικρό κόστος και συσκευές αποθήκευσης καθώς και την εκτενή υιοθέτηση της εικονοποίησης του hardware, την προσανατολισμένη στις υπηρεσίες αρχιτεκτονική και αυτόνομη και ωφέλιμη επεξεργασία έχει οδηγήσει στην ανάπτυξη του cloud computing. [6]

Το cloud computing έχει γίνει μια υπηρεσία υψηλής ζήτησης λόγω των πλεονεκτημάτων της υψηλής υπολογιστικής δύναμης, του φθηνού κόστους των υπηρεσιών, της υψηλής απόδοσης, της κλιμάκωσης, της προσβασιμότητας καθώς και της διαθεσιμότητας.

Το cloud computing είναι ένα μοντέλο το οποίο δίνει τη δυνατότητα για πανταχού παρόν, εύκολη, κατά παραγγελία δικτυακή πρόσβαση σε μία κοινή δεξαμενή παραμετροποιήσιμων πόρων (π.χ. δίκτυα, servers, αποθήκευση, εφαρμογές και υπηρεσίες) που μπορούν πολύ γρήγορα να εφοδιαστούν και να ελευθερωθούν με ελάχιστη προσπάθεια διαχείρισης ή

αλληλεπίδραση από τον πάροχο υπηρεσιών. Αυτό το μοντέλο cloud αποτελείται από πέντε ουσιώδη χαρακτηριστικά, τρία μοντέλα υπηρεσιών και τέσσερα μοντέλα παράταξης. [4]

3.1 Βασικά χαρακτηριστικά

3.1.1 Software as a Service (Saas)

Η δυνατότητα που παρέχεται στον καταναλωτή είναι να χρησιμοποιεί τις εφαρμογές του πάροχου που λειτουργούν σε μια cloud υποδομή. Οι εφαρμογές είναι προσβάσιμες από διάφορες συσκευές πελατών διαμέσου μίας διεπαφής πελάτη, όπως ένας web browser (π.χ. web-based email), ή μια διεπαφή προγράμματος. Ο καταναλωτής δε διαχειρίζεται ούτε ελέγχει τη βασική cloud υποδομή, συμπεριλαμβανομένου του δικτύου, των servers, των λειτουργικών συστημάτων, της αποθήκευσης ή ακόμα και ατομικές δυνατότητες εφαρμογών, με πιθανή εξαίρεση περιορισμένες, συγκεκριμένες για το χρήστη, ρυθμίσεις της εφαρμογής.

3.1.2 Software as a Service (Saas)

Η δυνατότητα που παρέχεται στον καταναλωτή είναι να αναπτύσσει στην cloud υποδομή εφαρμογές φτιαγμένες από καταναλωτή ή αποκτημένες εφαρμογές που δημιουργήθηκαν χρησιμοποιώντας γλώσσες προγραμματισμού, βιβλιοθήκες, υπηρεσίες και εργαλεία που υποστηρίζονται από τον πάροχο. Ο καταναλωτής δε διαχειρίζεται ούτε ελέγχει τη βασική cloud υποδομή, συμπεριλαμβανομένου του δικτύου, των servers, των λειτουργικών συστημάτων, της αποθήκευσης αλλά έχει έλεγχο πάνω στις αναπτυγμένες εφαρμογές και πιθανόν στις ρυθμίσεις του περιβάλλοντος που φιλοξενεί την εφαρμογή.

3.1.3 Software as a Service (Saas)

Η δυνατότητα που παρέχεται στον καταναλωτή είναι να εφοδιάζεται με πόρους επεξεργασίας, αποθήκευσης, δικτύων και άλλους θεμελιώδους υπολογιστικούς πόρους όπου ο καταναλωτής είναι σε θέση να αναπτύξει και να τρέξει αυθαίρετο λογισμικό, το οποίο μπορεί να περιλαμβάνει λειτουργικά συστήματα και εφαρμογές. Ο καταναλωτής δε διαχειρίζεται ούτε ελέγχει τη βασική cloud υποδομή αλλά έχει έλεγχο πάνω στα λειτουργικά συστήματα, την αποθήκευση και τις αναπτυγμένες εφαρμογές και πιθανόν περιορισμένο έλεγχο στην επιλογή συστατικών δικτύου (π.χ. host firewalls).

3.2 Μοντέλα παράταξης

3.2.1 Ιδιωτικό cloud

Η υποδομή cloud παρέχεται για αποκλειστική χρήση από ένα οργανισμό που περιλαμβάνει πολλαπλούς καταναλωτές (π.χ. business units). Μπορεί να κατέχεται, να διαχειρίζεται και να λειτουργείται από τον οργανισμό, από τρίτο ή από κάποιο συνδυασμό αυτών και μπορεί να υπάρχει εντός ή εκτός των εγκαταστάσεων του οργανισμού.

3.2.2 Κοινοτικό cloud

Η υποδομή cloud παρέχεται για αποκλειστική χρήση από μια συγκεκριμένη κοινότητα καταναλωτών από οργανισμούς που έχουν κοινές ανησυχίες (π.χ. αποστολή, απαιτήσεις ασφάλειας, πολιτική, εκτιμήσεις συμμόρφωσης). Μπορεί να κατέχεται, να διαχειρίζεται και να λειτουργείται από έναν ή περισσότερους από τους οργανισμούς της κοινότητας, από τρίτο ή από κάποιο συνδυασμό αυτών και μπορεί να υπάρχει εντός ή εκτός των εγκαταστάσεων του οργανισμού.

3.2.3 Δημόσιο cloud

Η υποδομή cloud παρέχεται για ανοιχτή χρήση από το γενικό κοινό. Μπορεί να κατέχεται, να διαχειρίζεται και να λειτουργείται από ένα επιχειρηματικό οργανισμό, ένα ακαδημαϊκό οργανισμό, ένα κυβερνητικό οργανισμό ή ένα συνδυασμό αυτών. Υπάρχει στις εγκαταστάσεις του πάροχου του cloud.

3.2.4 Υβριδικό cloud

Η υποδομή cloud είναι μια σύνθεση δύο ή περισσότερων διακριτών υποδομών cloud (ιδιωτικό, κοινοτικό ή δημόσιο) που παραμένουν μοναδικές οντότητες αλλά συνδέονται μεταξύ τους από πρότυπη ή ιδιόκτητη τεχνολογία που επιτρέπει τη φορητότητα των δεδομένων και τον εφαρμογών (π.χ. ισορρόπηση φορτίου μεταξύ υποδομών cloud). [4]

3.3 Αρχιτεκτονική Cloud

Όταν γίνεται λόγος για ένα υπολογιστικό σύστημα cloud, βοηθάει να χωριστεί σε δύο κατηγορίες. Το front end και το back end. Αυτά συνδέονται μεταξύ τους μέσω ενός δικτύου, συνήθως το Internet. Το front end είναι η πλευρά που βλέπει ο χρήστης του υπολογιστή ή ο πελάτης. Το back end είναι η cloud πλευρά του συστήματος.

Το front end περιλαμβάνει τον υπολογιστή του πελάτη (ή το δίκτυο υπολογιστών) και την εφαρμογή που απαιτείται για είναι προσβάσιμο το υπολογιστικό σύστημα cloud. Δεν έχουν όλα τα υπολογιστικά συστήματα cloud την ίδια διεπαφή χρήστη. Υπηρεσίες όπως προγράμματα web-based e-mail χρησιμοποιούν υπάρχοντες φυλλομετρητές. Άλλα συστήματα έχουν μοναδικές εφαρμογές που παρέχουν στους πελάτες πρόσβαση στο δίκτυο.

Στο back end του συστήματος υπάρχουν διάφοροι υπολογιστές, servers και συστήματα αποθήκευσης δεδομένων που δημιουργούν το "cloud" (σύννεφο) των υπολογιστικών υπηρεσιών. Στη θεωρία, ένα υπολογιστικό σύστημα cloud πρακτικά περιλαμβάνει κάθε πρόγραμμα υπολογιστή που μπορεί κανείς να φανταστεί, από επεξεργασία δεδομένων μέχρι video games. Συνήθως κάθε εφαρμογή θα έχει το δικό της dedicated server.

Ένας κεντρικός server διαχειρίζεται το σύστημα, παρακολουθεί την κίνηση και τις απαιτήσεις των πελατών ώστε να εξασφαλίζει ότι όλα τρέχουν ομαλά. Ακολουθείται ένα σύνολο από κανόνες που ονομάζονται πρωτόκολλα και χρησιμοποιείται ειδικό είδος λογισμικού που ονομάζεται ενδιάμεσο λογισμικό. Το ενδιάμεσο λογισμικό επιτρέπει στους δικτυωμένους υπολογιστές να επικοινωνούν μεταξύ τους. Την περισσότερη ώρα, οι servers δεν χρησιμοποιούνται πλήρως, αυτό σημαίνει ότι υπάρχει αχρησιμοποίητη υπολογιστική δύναμη που πηγαίνει χαμένη. Είναι εφικτό να ξεγελαστεί ένας φυσικός server και να νομίζει ότι στην πραγματικότητα είναι πολλαπλοί servers, ο καθένας με το δικό του ανεξάρτητο λειτουργικό σύστημα. Αυτή η τεχνική ονομάζετε server virtualization. Με το να μεγιστοποιείται η έξοδος του κάθε ατομικού sever, η τεχνική αυτή μειώνει την ανάγκη για περισσότερα φυσικά μηχανήματα.

3.4 Ασφάλεια και ιδιωτικότητα

Το cloud computing τοποθετεί ανησυχίες σχετικά με την ιδιωτικότητα διότι ο πάροχος της υπηρεσίας μπορεί να έχει πρόσβαση στα δεδομένα που βρίσκονται στο cloud ανά πάσα στιγμή. Θα μπορούσε κατά λάθος ή σκόπιμα να τροποποιήσει ή ακόμα και να διαγράψει πληροφορίες. Πολλοί πάροχοι cloud μπορούν να μοιράζονται πληροφορίες σε τρίτους αν χρειάζεται για νομικούς λόγους χωρίς καν την ύπαρξή εντάλματος. Αυτό είναι επιτρεπτό στην πολιτική ιδιωτικότητας τους, με την οποία οι χρήστες πρέπει να συμφωνήσουν πριν αρχίσουν να χρησιμοποιούν τις υπηρεσίες cloud. Λύσεις για την ιδιωτικότητα περιλαμβάνουν πολιτική νομοθεσία καθώς και επιλογές στο χρήστη για το πώς αποθηκεύονται τα δεδομένα. Οι χρήστες μπορούν να κρυπτογραφήσουν τα δεδομένα που επεξεργάζονται ή να τα αποθηκεύσουν μέσα στο cloud ώστε να αποτραπεί μη εξουσιοδοτημένη πρόσβαση σε αυτά.

Σύμφωνα με το Cloud Security Alliance, οι τρείς δημοφιλέστερες απειλές για το cloud είναι οι ανασφαλείς διεπαφές και API's, η απώλεια και διαρροή δεδομένων και η αποτυχία του

hardware, οι οποίες είναι υπεύθυνες για το 29%, 25% και 10% αντίστοιχα για όλες τις διακοπές του cloud για θέματα ασφάλειας. Μαζί, αυτές σχηματίζουν τις αδυναμίες της κοινόχρηστης τεχνολογίας. Σε μία cloud πλατφόρμα που μοιράζεται από πολλούς χρήστες υπάρχει η πιθανότητα πληροφορίες που ανήκουν σε διαφορετικούς χρήστες να βρίσκονται στο ίδιο server δεδομένων. Γι' αυτό το λόγο, διαρροή πληροφοριών μπορεί να συμβεί κατά λάθος όταν πληροφορίες ενός χρήστη δοθούν σε άλλο χρήστη.

4

Ουρές μηνυμάτων

Στην επιστήμη της πληροφορικής, οι ουρές μηνυμάτων είναι συστατικά μηχανικής λογισμικού που χρησιμοποιούνται για την επικοινωνία μεταξύ διεργασιών ή την επικοινωνία μεταξύ νημάτων της ίδιας διεργασίας. Χρησιμοποιούν μία ουρά για την επικοινωνία μέσω μηνυμάτων, το πέρασμα του ελέγχου ή του περιεχομένου. Ομάδες συστημάτων επικοινωνίας παρέχουν παρόμοια είδη λειτουργικότητας.

Το παράδειγμα της ουράς μηνυμάτων είναι παρόμοιο με το publisher/subscriber μοτίβο και τυπικά είναι ένα μέρος ενός μεγαλύτερου συστήματος ενδιάμεσου λογισμικού βασισμένο σε μηνύματα. Τα περισσότερα συστήματα μηνυμάτων υποστηρίζουν τόσο το μοντέλο publisher/subscriber όσο και το μοντέλο των ουρών μηνυμάτων στο API τους.

4.1 Επισκόπιση

Οι ουρές μηνυμάτων παρέχουν ένα πρωτόκολλο ασύγχρονης επικοινωνίας, που σημαίνει ότι ο αποστολέας και ο παραλήπτης του μηνύματος δε χρειάζεται να αλληλεπιδρούν με την ουρά μηνυμάτων την ίδια στιγμή. Τα μηνύματα που τοποθετούνται στην ουρά αποθηκεύονται μέχρι να ανακτηθούν από τον παραλήπτη τους. Οι ουρές μηνυμάτων έχουν ασαφή ή σαφή όρια για το μέγεθος των δεδομένων που μπορούν να μεταδοθούν σε ένα μήνυμα και τον αριθμό των μηνυμάτων που μπορούν να παραμένουν στην ουρά.

Πολλές υλοποιήσεις των ουρών μηνυμάτων λειτουργούν εσωτερικά, μέσα σε ένα λειτουργικό σύστημα ή μέσα σε μία εφαρμογή. Αυτές οι ουρές υπάρχουν μόνο για τους σκοπούς αυτού του συστήματος.

Άλλες υλοποιήσεις επιτρέπουν το πέρασμα των μηνυμάτων μεταξύ διαφορετικών υπολογιστικών συστημάτων, και την εν δυνάμει σύνδεση πολλαπλών εφαρμογών και πολλαπλών λειτουργικών συστημάτων. Αυτά τα συστήματα ουρών μηνυμάτων τυπικά παρέχουν βελτιωμένη λειτουργικότητα ελαστικότητας ώστε να διασφαλίζεται ότι τα μηνύματα δεν θα χαθούν σε περίπτωση αποτυχίας του συστήματος.

Οι υλοποιήσεις υπάρχουν σαν ιδιόκτητο λογισμικό, παρέχονται σαν υπηρεσία, λογισμικό ανοιχτού κώδικα ή λύσεις βασισμένες στο υλικό.

Τα περισσότερα λειτουργικά συστήματα πραγματικού χρόνου ενθαρρύνουν τη χρήση των ουρών μηνυμάτων ως τον κύριο μηχανισμό επικοινωνίας μεταξύ των διεργασιών ή των νημάτων. Αυτό έχει σαν αποτέλεσμα τη σφιχτή ενσωμάτωση μεταξύ του περάσματος των μηνυμάτων και της χρονοδρομολόγησης του επεξεργαστή και είναι ο κύριος λόγος για τη χρήση των λειτουργικών συστημάτων πραγματικού χρόνου σε εφαρμογές πραγματικού χρόνου.

4.2 Χρήση

Σε μία τυπική υλοποίηση ουράς μηνυμάτων, ένας διαχειριστής συστήματος εγκαθιστά και τροποποιεί το λογισμικό της ουράς μηνυμάτων και καθορίζει μία ουρά μηνυμάτων. Διαφορετικά γίνεται εγγραφή σε μια υπηρεσία ουρών μηνυμάτων.

Στη συνέχεια μία εφαρμογή εγγράφει μία ρουτίνα λογισμικού που περιμένει μηνύματα που τοποθετούνται στην ουρά.

Δευτερεύουσες και μεταγενέστερες εφαρμογές μπορούν να συνδέονται στην ουρά και να μεταφέρουν μηνύματα σε αυτή.

Το λογισμικό που διαχειρίζεται την ουρά αποθηκεύει τα μηνύματα μέχρι να συνδεθεί μια εφαρμογή παραλήπτης και τότε καλεί τη ρουτίνα λογισμικού που έχει εγγραφεί. Στη συνέχεια η εφαρμογή παραλήπτης επεξεργάζεται το μήνυμα με κατάλληλο τρόπο.

Συχνά υπάρχει διάφορες επιλογές για τα συγκεκριμένα σχέδια του περάσματος των μνημάτων, όπως φαίνεται παρακάτω.

Durability - Αντοχή. Τα μηνύματα μπορούν να κρατούνται στη μνήμη, να γράφονται στο δίσκο ή ακόμα να παραπέμπονται σε ένα σύστημα διαχείρισης βάσης δεδομένων, όταν η ανάγκη για αξιοπιστία υποδεικνύει μια λύση απαιτητική σε πόρους.

Security policies– Πολιτικές ασφάλειας. Ποιες εφαρμογές θα πρέπει να έχουν πρόσβαση σε συγκεκριμένα μηνύματα.

Message purging policies – Πολιτικές εκκαθάρισης μηνυμάτων. Οι ουρές ή τα μηνύματα μπορούν να έχουν χρόνο ζωής.

Message filtering – Φιλτράρισμα μηνυμάτων. Κάποια συστήματα παρέχουν φιλτράρισμα των δεδομένων ώστε ένας subscriber να μπορεί να μόνο να δει μηνύματα που ταιριάζουν με κάποια προκαθορισμένα κριτήρια ενδιαφέροντος.

Delivery policies – Πολιτικές παράδοσης. Χρειάζεται να εγγυηθεί ότι ένα μήνυμα θα παραδοθεί τουλάχιστον μία φορά ή όχι περισσότερες από μία φορές.

Routing policies – Πολιτικές δρομολόγησης. Σε ένα σύστημα με πολλούς servers ουρών, ποιοι servers θα πρέπει να δέχονται ένα μήνυμα ή τα μηνύματα μίας ουράς.

Batching policies – Πολιτικές δεσμίδων. Θα πρέπει τα μηνύματα να παραδίδονται άμεσα ή θα πρέπει το σύστημα να περιμένει λίγο και να προσπαθεί να παραδίδει πολλά μηνύματα με τη μία.

Queueing criteria – Κριτήρια ουράς. Πότε ένα μήνυμα πρέπει να θεωρείται ότι έχει μπει στην ουρά. Πότε μία ουρά έχει ένα μήνυμα. Πότε ένα μήνυμα έχει προωθηθεί τουλάχιστον σε μία απομακρυσμένη ουρά ή σε όλες τις ουρές.

Receipt notification – Ειδοποίηση παράδοσης. Ένας publisher μπορεί να χρειάζεται να ξέρει πότε ένας ή όλοι η subscribers έχουν παραλάβει ένα μήνυμα.

Αυτοί όλοι είναι παράγοντες που μπορούν να έχουν σημαντική επίδραση στα σχέδια συναλλαγών, στην αξιοπιστία του συστήματος και στην αποδοτικότητα του συστήματος.

4.3 Πρότυπα και πρωτόκολλα

Ιστορικά, οι ουρές μηνυμάτων χρησιμοποιούσαν ιδιωτικά, κλειστά πρωτόκολλα, περιορίζοντας την ικανότητα για διαφορετικά λειτουργικά συστήματα ή γλώσσες προγραμματισμού να αλληλεπιδρούν σε ένα ετερογενές σύνολο από περιβάλλοντα.

Έχουν αναδυθεί τρία πρότυπα που χρησιμοποιούνται σε ανοιχτού κώδικα υλοποιήσεις ουρών μηνυμάτων.

Advanced Message Queueing Protocol (AMQP). Πρωτόκολλο ουρών μηνυμάτων πλούσιο σε χαρακτηριστικά και ιδιότητες.

Streaming Text Oriented Messaging Protocol (STOMP). Πρωτόκολλο ουρών μηνυμάτων απλό και προσανατολισμένο σε κείμενο.

MQTT. Πρωτόκολλο ουρών μηνυμάτων ελαφρύ, ειδικό για ενσωματωμένες συσκευές.

Αυτά τα πρωτόκολλα είναι σε διαφορετικά στάδια προτυποποίησης και υιοθέτησης. Τα δύο πρώτα λειτουργούν στο ίδιο επίπεδο όπως το HTTP, ενώ το MQTT πρωτόκολλο λειτουργεί σε επίπεδο TPC/IP.

Κάποιες ιδιωτικές υλοποιήσεις επίσης χρησιμοποιούν το HTTP πρωτόκολλο για να παρέχουν υπηρεσίες ουρών μηνυμάτων. Αυτό συμβαίνει γιατί είναι πάντα εφικτό τεθεί σε επίπεδα η ασύγχρονη συμπεριφορά (κάτι το οποίο αποτελεί προϋπόθεση για τις ουρές μηνυμάτων) πάνω σε ένα σύγχρονο πρωτόκολλο χρησιμοποιώντας σχέδια αίτησης-απάντησης. Ωστόσο, σε αυτή την περίπτωση, τέτοιες υλοποιήσεις περιορίζονται από το βασικό πρωτόκολλο και μπορεί να μην είναι σε θέση να προσφέρουν πλήρη πιστότητα ή ένα σύνολο επιλογών που απαιτούνται στο πέρασμα των μηνυμάτων που αναφέρθηκε παραπάνω.

4.4 Το AMQP πρωτόκολλο

Το Advanced Message Queuing Protocol (AMQP) είναι ένα πρωτόκολλο ανοιχτής προτυποποίησης σε επίπεδο εφαρμογής για ενδιάμεσο λογισμικό προσανατολισμένο στα μηνύματα. Τα καθοριστικά χαρακτηριστικά του AMQP είναι ο προσανατολισμός στα μηνύματα, η ουρές, η δρομολόγηση, η αξιοπιστία και η ασφάλεια.

Το AMQP επιτάσσει τη συμπεριφορά του πάροχου μηνυμάτων και του πελάτη στο βαθμό που οι υλοποιήσεις από διαφορετικούς πωλητές είναι διαλειτουργικές. Κατά τον ίδιο τρόπο που τα SMTP, HTTP, FTP κ.τ.λ. έχουν δημιουργήσει διαλειτουργικά συστήματα. Προηγούμενες προτυποποιήσεις του ενδιάμεσου λογισμικού έχουν συμβεί στο επίπεδο του API και είχαν επικεντρωθεί στην προτυποποίηση της αλληλεπίδρασης του προγραμματιστή με διαφορετικές υλοποιήσεις ενδιάμεσου λογισμικού, αντί να γίνει εστίαση στην παροχή διαλειτουργικότητας μεταξύ πολλαπλών υλοποιήσεων. Το AMQP είναι ένα wire-level πρωτόκολλο. Ένα wire-level πρωτόκολλο είναι μια περιγραφή της μορφής των δεδομένων που στέλνονται μέσω του δικτύου σαν ένα ρεύμα από οκτάδες. Συνεπώς, κάθε εργαλείο που μπορεί να δημιουργεί και να ερμηνεύει μηνύματα που συμμορφώνονται με αυτή τη μορφή δεδομένων μπορεί να διαλειτουργεί με κάθε άλλο συμμορφωμένο εργαλείο, ανεξάρτητα από τη γλώσσα υλοποίησης.[8]

Το AMQP είναι ένα πρωτόκολλο δυαδικό, σε επίπεδο εφαρμογής, σχεδιασμένο να υποστηρίζει αποτελεσματικά ένα μεγάλο εύρος ποικίλων εφαρμογών μηνυμάτων και επικοινωνιακών μοτίβων. Παρέχει ροή ελέγχου, επικοινωνία προσανατολισμένη στα μηνύματα με εγγυήσεις παράδοσης μηνυμάτων, πιστοποίηση και κρυπτογράφηση.

Ο προσδιορισμός του AMQP καθορίζεται από διάφορα επίπεδα.

1. Ένα τύπο του συστήματος.

2. Ένα συμμετρικό, ασύγχρονο πρωτόκολλο για τη μεταφορά των μηνυμάτων από τη μία διεργασία στην άλλη.
3. Μία πρότυπη, επεκτάσιμη μορφή μηνύματος.
4. Ένα σύνολο από πρότυπες αλλά επεκτάσιμες ιδιότητες μηνυμάτων.

4.5 Περιγραφή του AMQP

4.5.1 Τύπος συστήματος

Το AMQP καθορίζει ένα αυτό-περιγραφόμενο σχήμα κωδικοποίησης που επιτρέπει τη διαλειτουργική αναπαράσταση ενός μεγάλου εύρους συχνών χρησιμοποιούμενων τύπων. Επίσης επιτρέπει σε τυποποιημένα δεδομένα να υποσημειώνονται με επιπρόσθετες σημασίες.

Ο τύπος συστήματος χρησιμοποιείται για να καθορίζει τη μορφή ενός μηνύματος επιτρέποντας στα πρότυπα και εκτεταμένα δεδομένα να εκφραστούν και να κατανοηθούν με την επεξεργασία οντοτήτων. Επίσης χρησιμοποιείται για τον καθορισμό των πρωτόγονων επικοινωνιών μέσω των οποίων τα μηνύματα ανταλλάσσονται μεταξύ τέτοιων οντοτήτων.[9]

4.5.2 Μονάδα δεδομένων και το πρωτόκολλο link

Η βασική μονάδα δεδομένων στο AMQP είναι το frame. Υπάρχουν εννιά καθορισμένα σώματα από AMQP frames που χρησιμοποιούνται για την εκκίνηση, τον έλεγχο και τον τερματισμό της μετάδοσης μηνυμάτων.

- Open
- Begin
- Attach
- Transfer
- Flow
- Disposition
- Detach
- End
- Close

Το πρωτόκολλο link βρίσκεται στην καρδιά του AMQP.

Ένα attach frame σώμα στέλνεται για να ξεκινήσει ένας καινούριος σύνδεσμος (link) και ένα detach frame για να τερματιστεί. Οι σύνδεσμοι εγκαθίστανται ώστε να λαμβάνονται και να στέλνονται μηνύματα.

Τα μηνύματα στέλνονται μέσω ενός συνδέσμου χρησιμοποιώντας το transfer frame. Τα μηνύματα σε ένα σύνδεσμο έχουν ροή προς μία μόνο κατεύθυνση.

Οι μεταφορές υπόκεινται σε ένα σχήμα ελέγχου ροής και διαχειρίζονται χρησιμοποιώντας flow frames. Αυτό επιτρέπει σε μία διεργασία να προστατεύει τον εαυτό της από τον κατακλυσμό της από πολύ μεγάλο όγκο μηνυμάτων ή πιο απλά επιτρέπει σε ένα subscribing σύνδεσμο να τραβάει μηνύματα όπως και όταν είναι επιθυμητό.

Κάθε μεταφερόμενο μήνυμα πρέπει τελικά να τακτοποιηθεί. Η τακτοποίηση εξασφαλίζει ότι ο αποστολέας και ο παραλήπτης συμφωνούν στην κατάσταση της μεταφοράς, παρέχοντας εγγυήσεις αξιοπιστίας. Οι αλλαγές στην κατάσταση και την τακτοποίηση για μία μεταφορά ή ένα σύνολο μεταφορών, γίνονται γνωστές στους αποστολείς/παραλήπτες χρησιμοποιώντας το disposition frame. Με αυτό τον τρόπο μπορούν να επιβληθούν διάφορες εγγυήσεις αξιοπιστίας.

Πολλαπλοί σύνδεσμοι, και προς τις δύο κατευθύνσεις, μπορούν να ομαδοποιηθούν μαζί σε μία συνεδρία (session). Μία συνεδρία είναι αμφίδρομη, ακολουθιακή συνομιλία μεταξύ δύο συνομιλητών που αρχίζει με ένα begin frame και τερματίζει με ένα end frame. Μία σύνδεση μεταξύ δύο συνομιλητών μπορεί να έχει πολλαπλές συνεδρίες που πολυπλέκονται πάνω σε αυτή, όντας η κάθε μία λογικά ανεξάρτητη. Οι συνδέσεις αρχίζουν με ένα open frame στο οποίο εκφράζονται οι δυνατότητες του αποστολέα, και τερματίζονται με το close frame.

4.5.3 Μορφή μηνύματος

Το AMQP καθορίζει το ‘γυμνό’ μήνυμα, το κομμάτι του μηνύματος που δημιουργείται από την εφαρμογή αποστολέα. Αυτό θεωρείται αμετάβλητο καθώς το μήνυμα μεταφέρεται μεταξύ μίας ή περισσότερων διεργασιών.

Διασφαλίζοντας ότι το μήνυμα που στέλνεται από την εφαρμογή είναι αμετάβλητο επιτρέπει την κρυπτογράφηση των μηνυμάτων και εξασφαλίζει ότι οι έλεγχοι για ακεραιότητα παραμένουν έγκυροι. Το μήνυμα μπορεί να υποσημειωθεί από ενδιάμεσους κατά τη μετάβαση αλλά κάθε τέτοια υποσημείωση παραμένει ξεχωριστή από το αμετάβλητο ‘γυμνό’ μήνυμα. Οι υποσημειώσεις μπορούν να προστεθούν πριν ή μετά το γυμνό μήνυμα.

Η επικεφαλίδα (header) είναι ένα πρότυπο σύνολο από υποσημειώσεις σχετικές με την παράδοση που μπορούν να ζητηθούν ή να υποδεικνύουν για ένα μήνυμα και περιλαμβάνουν το χρόνο ζωής, την αντοχή και την προτεραιότητα.

Το γυμνό μήνυμα από μόνο του είναι δομημένο σαν μία λίστα επιλογών από πρότυπες ιδιότητες (message id, user id, creation time, reply to, subject, correlation id, group id κ.τ.λ.), μία λίστα επιλογών με συγκεκριμένες ιδιότητες για την εφαρμογή και ένα σώμα στο οποίο το AMQP αναφέρεται ως δεδομένα εφαρμογής.

Οι ιδιότητες προσδιορίζονται στο AMQP τύπο συστήματος ως υποσημειώσεις. Τα δεδομένα εφαρμογής μπορούν να έχουν οποιοδήποτε μορφή, και να είναι σε οποιαδήποτε κωδικοποίηση που επιλέγει η εφαρμογή. Μία επιλογή είναι να χρησιμοποιηθεί ο AMQP τύπος συστήματος για να σταλθούν δομημένα αυτό-περιγραφικά δεδομένα.

4.5.4 Δυνατότητες μηνυμάτων

Το πρωτόκολλο link μεταφέρει μηνύματα μεταξύ δύο κόμβων αλλά υποθέτει πολύ λίγα ως προς τι είναι αυτοί οι κόμβοι ή πως έχουν υλοποιηθεί.

Αυτοί οι κόμβοι είναι μία κατηγορία κλειδί και χρησιμοποιείται σαν σημείο συνάντησης μεταξύ αποστολέων και παραληπτών μηνυμάτων. Ο AMQP προσδιορισμός ονομάζει αυτούς τους κόμβους ως κόμβους κατανομής και κωδικοποιεί κάποιες κοινές συμπεριφορές. Αυτές περιλαμβάνουν.

- Κάποια πρότυπα συμπεράσματα για μεταφορές, μέσω των οποίων οι παραλήπτες των μηνυμάτων μπορούν για παράδειγμα να αποδέχονται ή να απορρίπτουν μηνύματα.
- Ένα μηχανισμό για να ζητά ή να υποδεικνύει ένα από τα δύο βασικά μοτίβα κατανομής, τους ανταγωνιστικούς και τους μη ανταγωνιστικούς καταναλωτές μηνυμάτων.
- Την ικανότητα να δημιουργούνται κόμβοι κατά παραγγελία.
- Την ικανότητα εκκαθαρίζεται ένα σύνολο μηνυμάτων ενδιαφέροντος για ένα παραλήπτη μέσω φίλτρων.

Παρόλο που το AMQP μπορεί να χρησιμοποιηθεί σε απλά peer-to-peer συστήματα, ο καθορισμός αυτού του framework για δυνατότητες μηνυμάτων επιτρέπει επιπρόσθετα την διαλειτουργικότητα με τους διαμεσολαβητές μηνυμάτων σε μεγαλύτερα, πιο πλούσια δίκτυα μηνυμάτων. Το καθορισμένο framework καλύπτει βασικές συμπεριφορές αλλά επιτρέπει επεκτάσεις που μπορούν να εξελιχθούν και μπορούν περαιτέρω να κωδικοποιηθούν και να προτυποποιηθούν.

4.6 Σύγχρονη και ασύγχρονη επικοινωνία

Πολλά από τα πιο ευρέως γνωστά πρωτόκολλα επικοινωνίας σε χρήση λειτουργούν σύγχρονα. Το HTTP πρωτόκολλο, που χρησιμοποιείται στον παγκόσμιο ιστό και στις διαδικτυακές υπηρεσίες, προσφέρει ένα προφανές παράδειγμα στο οποίο ένας χρήστης στέλνει μία αίτηση για μία ιστοσελίδα και μετά περιμένει για μία απάντηση.

Ωστόσο, υπάρχουν σενάρια στα οποία η σύγχρονη συμπεριφορά δεν είναι κατάλληλη. Παραδείγματα ασύγχρονης επικοινωνίας υπάρχουν σε συστήματα ειδοποίησης γεγονότων και σε συστήματα publish/subscribe. Μία εφαρμογή μπορεί να χρειάζεται να ειδοποιήσει μία

άλλη ότι ένα γεγονός συνέβη, αλλά δε χρειάζεται να περιμένει για μία απάντηση. Σε συστήματα publish/subscribe, μία εφαρμογή ‘δημοσιεύει’ πληροφορίες για ένα αριθμό πελατών που τις διαβάζουν. Και για τα δύο παραπάνω παραδείγματα δεν έχει νόημα ο αποστολέας της πληροφορίας να χρειάζεται να περιμένει αν για παράδειγμα ένας από τους παραλήπτες έχει πρόβλημα.

5

Περιγραφή του προβλήματος - Cloud Marketplace

Οι υπηρεσίες cloud είναι υπηρεσίες διαθέσιμες για τους χρήστες κατά παραγγελία μέσω του διαδικτύου από πάροχους cloud computing servers σε αντίθεση με cloud computing servers που βρίσκονται που βρίσκονται στα κτίρια της ίδιας της εταιρείας. Αυτές οι υπηρεσίες είναι σχεδιασμένες για να παρέχουν εύκολη και κλιμακωτή πρόσβαση σε εφαρμογές και πόρους. Είναι πλήρως διαχειρίσιμες από έναν πάροχο υπηρεσιών cloud.

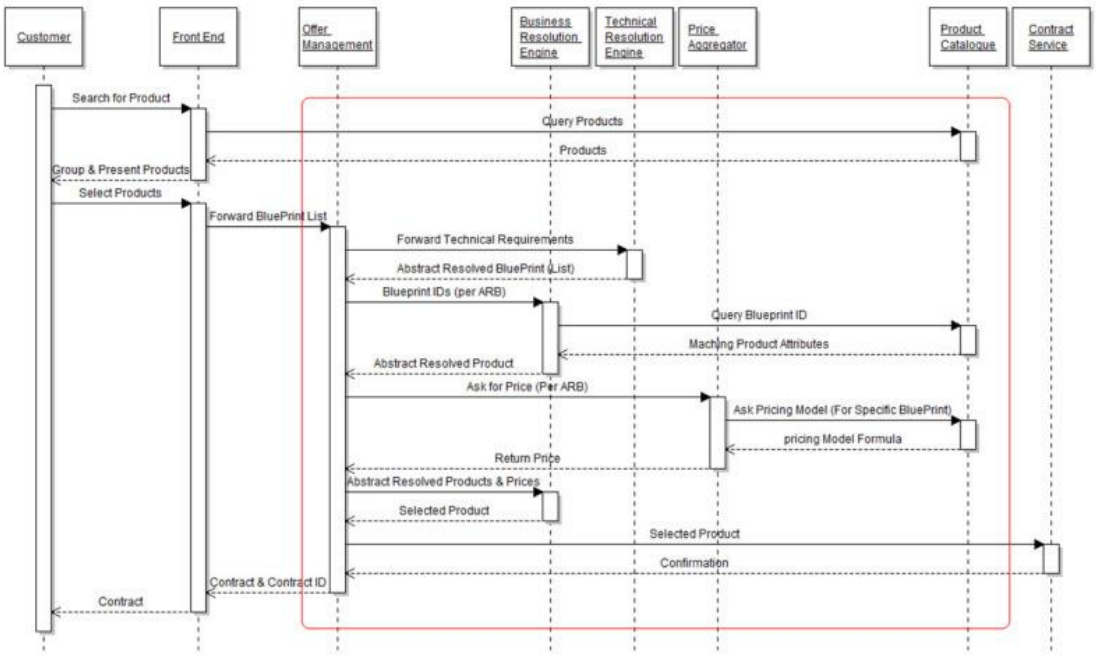
Οι υπηρεσίες cloud μπορούν δυναμικά να κλιμακώνονται ώστε να καλύπτουν τις ανάγκες των χρηστών και επειδή ο πάροχος των υπηρεσιών προμηθεύει το υλικό και το λογισμικό που χρειάζεται για την υπηρεσία, δεν υπάρχει ανάγκη για μία εταιρεία να παρέχει δικούς της πόρους ή να διανείμει IT προσωπικό για να διαχειρίζεται τις υπηρεσίες.

Έτσι οι υπηρεσίες cloud έχουν εξελιχθεί πολύ και έχουν γίνει πολύ δημοφιλείς. Υπάρχουν όμως πάρα πολλές διαφορετικές υπηρεσίες και όλες έχουν πολλές διαφορετικές ιδιότητες και ποικίλα χαρακτηριστικά. Γι' αυτούς τους λόγους η επιλογή ενός συνόλου υπηρεσιών γίνεται πολύ δύσκολη διαδικασία. Οι πάροχοι των υπηρεσιών cloud έχουν αναπτύξει marketplaces για την αγορά των υπηρεσιών που θέλει ο κάθε χρήστης. Οι χρήστες όμως θέλουν να αγοράζουν διαφορετικές υπηρεσίες cloud από διαφορετικούς πάροχους. Γεννάται έτσι η ανάγκη της δημιουργίας ενός ευρύτερου marketplace υπηρεσιών cloud που θα περιλαμβάνει όλων των ειδών των υπηρεσιών από όλους τους πάροχους που τις προσφέρουν. Θα πρέπει να είναι σε θέση να παίρνει σαν είσοδο το σύνολο των υπηρεσιών που χρειάζεται ο χρήστης μαζί με τις απαιτήσεις που θέτει. Σαν έξοδο θα πρέπει να δίνει στο χρήστη το βέλτιστο σύνολο με

τις υπηρεσίες cloud που έχει ζητήσει που ταυτόχρονα να ικανοποιεί τις απαιτήσεις του χρήστη.

Ένα τέτοιο marketplace έχει υλοποιηθεί με χρήση του 4CaaSt framework. Το 4CaaSt marketplace είναι ένα περιβάλλον για την παροχή cloud υπηρεσιών μέσω ενός δυναμικού οικοσυστήματος. Οι πάροχοι των υπηρεσιών χρησιμοποιούν τις λειτουργίες του marketplace για να ορίσουν νέες προσφορές σε προϊόντα, πληροφορίες σχετικά με την επιχειρηματικότητα και υπηρεσίες ενώ οι χρήστες μπορούν να θέσουν συγκεκριμένες αιτήσεις για υπηρεσίες και να κλείσουν συμβόλαιο με πάροχους. Οι μοναδικές προσφορές του marketplace εστιάζονται στην αλληλεπίδραση μεταξύ των τεχνικών και επιχειρηματικών πλευρών των υπηρεσιών αντιμετωπίζοντας τις προκλήσεις που σχετίζονται με μοντέλα τιμολόγησης σε κατανεμημένα περιβάλλοντα. Το marketplace ενσωματώνει ένα προσομοιωτή τιμολόγησης για τα προσφερόμενα προϊόντα, επιτρέποντας στους πάροχους να αξιολογούν τα μοντέλα τιμολόγησης μαζί με το προϊόν και να το συνδέσουν με τα τεχνικά και επιχειρηματικά χαρακτηριστικά. Σύμφωνα με τη διαδικασία της προσομοίωσης, το marketplace προτείνει συγκεκριμένα μοντέλα τιμολόγησης και τα υλοποιεί για κάθε ένα από τα προϊόντα με ένα δυναμικό τρόπο.

Στη συνέχεια φαίνεται το ακολουθιακό διάγραμμα του 4CaaSt marketplace για τη διαδικασία της αναζήτησης και της επιλογής προϊόντος.



Εικόνα 1 4CaaSt marketplace - Διαδικασία αναζήτησης και επιλογής προϊόντος

Ιδιαίτερο ενδιαφέρον παρουσιάζουν τα μοντέλα τιμολόγησης (price models). Το μοντέλο τιμολόγησης ενός προϊόντος καθορίζει πόσο χρεώνεται ένας καταναλωτής, ανάλογα με το πόσο έχει χρησιμοποιηθεί αυτό το προϊόν. Μία από τις βασικές καινοτομίες του 4CaaSt marketplace είναι η ιδέα των μικτών υπηρεσιών, που βασίζονται σε υπηρεσίες από τρίτους που προσφέρονται από διαφορετικούς πάροχους. Τα πλεονεκτήματα αυτού του αυτόματου συνδυασμού των υπηρεσιών από διαφορετικούς πάροχους περιλαμβάνουν το μειωμένο κόστος, την περισσότερη λειτουργικότητα και τη μεγαλύτερη ευελιξία στην επιλογή των υπηρεσιών. Ωστόσο, αυτή η αυτόματη δημιουργία μικτών υπηρεσιών οδηγεί σε ένα πολύ μεγάλο αριθμό από συνδυασμούς υπηρεσιών, ο καθένας εκ των οποίων πρέπει να τιμολογηθεί κατάλληλα. Για να οριστεί το μοντέλο τιμολόγησης για μία μικτή υπηρεσία, ο πάροχος πρέπει να εξετάσει διάφορους παράγοντες όπως το κόστος της ανάπτυξης και της λειτουργίας της μικτής υπηρεσίας και το κόστος που προκύπτει από τη χρήση υπηρεσιών από τρίτους. Η ικανότητα του marketplace να δημιουργεί αυτόματα μικτέ υπηρεσίες οδηγεί στην αλλαγή του κόστους του τελευταίου παράγοντα με το πέρασμα του χρόνου, χωρίς την παρέμβαση του πάροχου της μικτής υπηρεσίας. Για να αποτραπεί το να γίνουν παράλογες αποφάσεις τιμολόγησης και να γίνει αποφυγή της χειροκίνητης δουλειάς στο χειρισμό των εσόδων μέσα σε ένα μεγάλο αριθμό από μικτές υπηρεσίες, το marketplace μπορεί να αυτόματα να συναθροίσει αρκετά μοντέλα τιμολόγησης σε ένα συσωρευτικό μοντέλο τιμολόγησης. Έτσι, ο πάροχος της μικτής υπηρεσίας μπορεί να αναλύσει ποια κόστη προέρχονται από τη χρήση υπηρεσιών από τρίτους και μπορεί να προσδιορίσει κατάλληλα το μοντέλο τιμολόγησης για τη μικτή υπηρεσία. Το marketplace μπορεί αυτόματα να ενημερώσει τον πάροχο μιας μικτής υπηρεσίας για αλλαγές στις συνθήκες, όπως για καινούριες προσφορές για υπηρεσίες από τρίτους, και προτείνει ένα ανανεωμένο μοντέλο τιμολόγησης.

Στην παρούσα εργασία μελετήθηκε η περίπτωση ενός απλοποιημένου marketplace. Η περιγραφή του γίνεται στις επόμενες ενότητες.

5.1 Τα συστατικά του marketplace

Σε αυτό το marketplace ο χρήστης μπορεί να εισάγει τις τεχνικές και τις επιχειρηματικές απαιτήσεις που έχει για το σύνολο των υπηρεσιών που θέλει. Το σύστημα επιστρέφει ένα σύνολο πακέτων με τις επιθυμητές υπηρεσίες που είναι συμβατά με τις απαιτήσεις του χρήστη. Τα πακέτα είναι βαθμολογημένα σύμφωνα με ένα αλγόριθμο αξιολόγησης.

Για να το πετύχει αυτό, το σύστημα έχει τέσσερεις βασικές μονάδες και ως εκ τούτου χωρίζει τη διαδικασία σε τέσσερα στάδια. Οι μονάδες αυτές είναι η Technical Resolution Engine, η Business Resolution Engine, ο Price Aggregator και η Final Resolution Engine. Η κάθε μονάδα επεξεργάζεται δεδομένα και δημιουργεί καινούρια.

Το σύνολο των υπηρεσιών cloud καθώς και οι ιδιότητες και τα χαρακτηριστικά τους αποθηκεύονται σε μία βάση δεδομένων. Επίσης η κάθε μονάδα του συστήματος παίρνει τα δεδομένα που χρειάζεται από τη βάση δεδομένων και αποθηκεύει σε αυτή τα δεδομένα που παράγει.

Η επικοινωνία με το χρήστη γίνεται μέσω μίας διεπαφής front end. Ο χρήστης εισάγει το σύνολο με τις υπηρεσίες cloud και τις απαιτήσεις στη διεπαφή. Στο τέλος, το σύστημα μέσω της διεπαφής αυτής, επιστρέφει στο χρήστη το αποτέλεσμα.

5.2 Τα στάδια του marketplace

Η διαδικασία επιλογής των συμβατών υπηρεσιών cloud σύμφωνα με τις απαιτήσεις του χρήστη και της τελικής αξιολόγησης, όπως αναφέρθηκε παραπάνω, χωρίζεται σε τέσσερα στάδια. Πρώτον, υπάρχει το στάδιο ανάλυσης των τεχνικών απαιτήσεων του χρήστη (Technical Resolution). Δεύτερον, υπάρχει το στάδιο ανάλυσης των επιχειρηματικών απαιτήσεων του χρήστη (Business Resolution). Τρίτον, υπάρχει το στάδιο άθροισης του συνολικού κόστους των υπηρεσιών (Price Aggregator). Τέλος, υπάρχει το στάδιο της τελικής ανάλυσης και αξιολόγησης (Final Resolution).

5.2.1 Technical resolution

Σε αυτό το στάδιο, το σύστημα διαβάζει τις τεχνικές απαιτήσεις του χρήστη. Στη συνέχεια αναζητεί στη βάση δεδομένων τις υπηρεσίες και φτιάχνει ένα σύνολο από πακέτα με όλους τους συνδυασμούς των υπηρεσιών που έχει ζητήσει ο χρήστης και ικανοποιούν αυτές τις απαιτήσεις. Τα πακέτα αυτά περιέχουν πληροφορία για τα τεχνικά χαρακτηριστικά και για τα επιχειρηματικά χαρακτηριστικά. Τέλος, αποθηκεύονται τα πακέτα στη βάση δεδομένων και προωθείται η πληροφορία στο επόμενο στάδιο.

5.2.2 Business resolution

Στο στάδιο αυτό, το σύστημα διαβάζει τις επιχειρηματικές απαιτήσεις του χρήστη. Για κάθε πακέτο υπηρεσιών cloud, που έχει παραχθεί από το προηγούμενο στάδιο, για κάθε υπηρεσία από αυτές υπάρχουν διαφορετικοί πάροχοι που τις προσφέρουν με διαφορετικά χαρακτηριστικά και ιδιότητες. Το σύστημα λοιπόν, δημιουργεί για κάθε τέτοια υπηρεσία όλους τους δυνατούς συνδυασμούς και εξάγει ένα σύνολο πακέτων με υπηρεσίες cloud. Τα πακέτα αυτά περιέχουν πληροφορία για τα τεχνικά χαρακτηριστικά και για τα επιχειρηματικά χαρακτηριστικά. Τέλος, αποθηκεύονται τα πακέτα στη βάση δεδομένων και προωθείται η πληροφορία στο επόμενο στάδιο.

5.2.3 *Price aggregator*

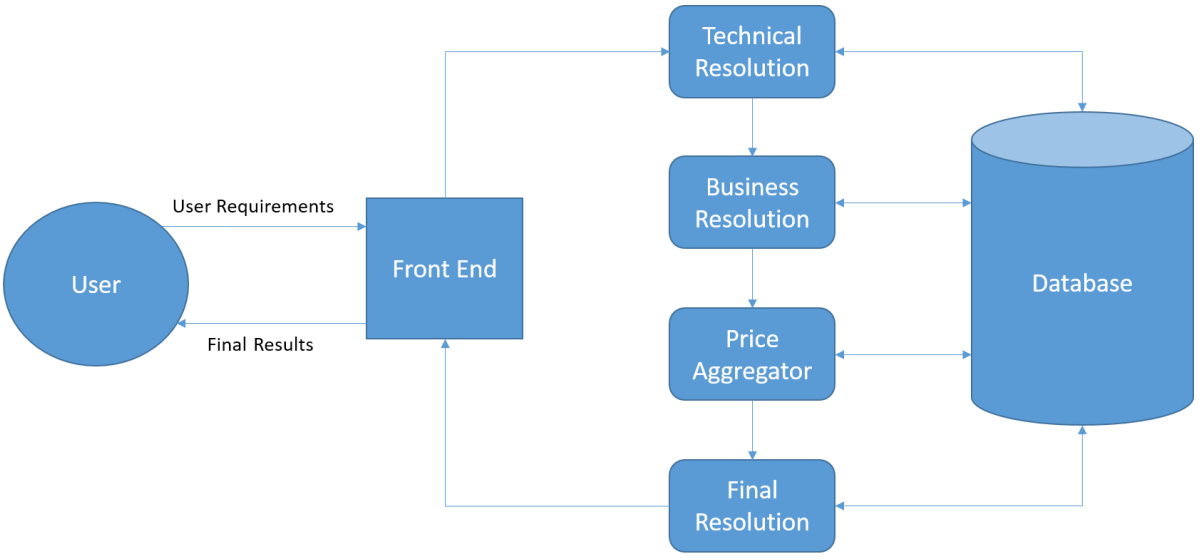
Στο στάδιο αυτό, το σύστημα διαβάζει τις απαιτήσεις του χρήστη ως προς το κόστος των υπηρεσιών cloud. Έπειτα για κάθε πακέτο υπηρεσιών που έχει παραχθεί από το προηγούμενο στάδιο αθροίζεται το κόστος κάθε υπηρεσίας. Απορρίπτονται τα πακέτα αυτά που δε συμφωνούν στο συνολικό κόστος με τις απαιτήσεις που έχει θέσει ο χρήστης. Τα πακέτα που δεν απορρίφθηκαν περιέχουν τώρα πληροφορία για τα τεχνικά χαρακτηριστικά, για τα επιχειρηματικά χαρακτηριστικά και για το συνολικό κόστος. Τέλος, αποθηκεύονται τα πακέτα στη βάση δεδομένων και προωθείται η πληροφορία στο επόμενο στάδιο.

5.2.4 *Final resolution*

Στο τελικό στάδιο, το σύστημα εξετάζει τα πακέτα υπηρεσιών cloud που παρήχθησαν από το προηγούμενο στάδιο. Διαβάζει τι ιδιότητες και τα χαρακτηριστικά κάθε υπηρεσίας και τις βάζει ως είσοδο σε ένα αλγόριθμο αξιολόγησης. Ο αλγόριθμος βγάζει ένα βαθμό αξιολόγησης για κάθε ιδιότητα/χαρακτηριστικό. Σε αυτό το στάδιο απορρίπτονται τα πακέτα με υπηρεσίες που δεν ικανοποιούν τις απαιτήσεις που έχει θέσει ο χρήστης. Για τα πακέτα που δεν απορρίπτεται υπολογίζεται ένας συνολικός βαθμός αξιολόγησης σύμφωνα με μία συνάρτηση. Έτσι παράγονται πακέτα με υπηρεσίες cloud που ικανοποιούν τις απαιτήσεις του χρήστη και είναι σε φθίνουσα σειρά με κριτήριο το βαθμό αξιολόγησής τους. Τα πακέτα αποθηκεύονται στη βάση δεδομένων.

5.3 *Αρχιτεκτονική του marketplace*

Στο παρακάτω σχήμα απεικονίζεται η αρχιτεκτονική του συστήματος marketplace για υπηρεσίες cloud.



Εικόνα 2 Αρχιτεκτονική του marketplace

Ο χρήστης, μέσω της διεπαφής front end, εισάγει στο σύστημα τα δεδομένα. Αυτά αποτελούνται από το σύνολο των διαφορετικών υπηρεσιών cloud που θέλει να αγοράσει ο χρήστης. Τις τεχνικές απαιτήσεις για τις υπηρεσίες οι οποίες μπορεί να αφορούν για παράδειγμα αρχιτεκτονική x86 ή x64, SQL ή NoSQL βάση δεδομένων, Windows ή Linux λειτουργικό σύστημα και άλλα. Τις επιχειρηματικές απαιτήσεις για τις υπηρεσίες οι οποίες μπορεί να αφορούν τη διαθεσιμότητα, την αξιοπιστία, την ασφάλεια και άλλα. Το μέγιστο συνολικό κόστος για τις υπηρεσίες.

Το front end επικοινωνεί με το technical resolution. Το technical resolution επικοινωνεί με το business resolution, αυτό με τη σειρά του με τον price aggregator, και αυτός με το final resolution. Η κάθε μονάδα από αυτές τις τέσσερεις έχουν αλληλεπίδραση με τη βάση δεδομένων. Ανακτούν δεδομένα από αυτή για την είσοδό τους και αποθηκεύουν σε αυτή την έξοδό τους.

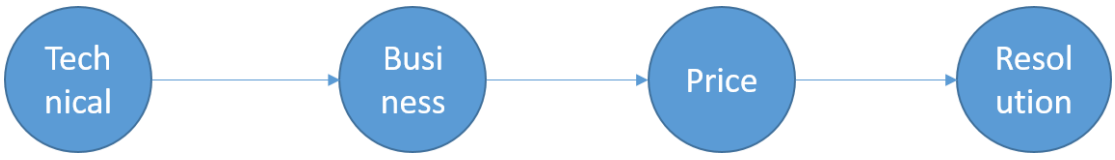
Στην τελική φάση, τα αποτελέσματα από το final resolution προωθούνται στο front end. Μέσω του front end ο χρήστης παίρνει τελικά τη λίστα με τα σύνολα υπηρεσιών cloud που καλύπτουν τις απαιτήσεις του. Το σύστημα βρίσκει τις βέλτιστες επιλογές και παρουσιάζει τα αποτελέσματα βαθμολογημένα, αξιολογώντας τα χαρακτηριστικά και τις ιδιότητες των υπηρεσιών και των πάροχων τους, σύμφωνα με τις απαιτήσεις του χρήστη.

6

Η υλοποίηση του marketplace

Στην παρούσα διπλωματική εργασία υλοποιούνται δύο διαφορετικές αρχιτεκτονικές σχετικά με την επεξεργασία των δεδομένων στα τέσσερα στάδια που έχουν αναφερθεί για το marketplace υπηρεσιών cloud. Πως αυτά τα στάδια λειτουργούν και πως επικοινωνούν μεταξύ τους.

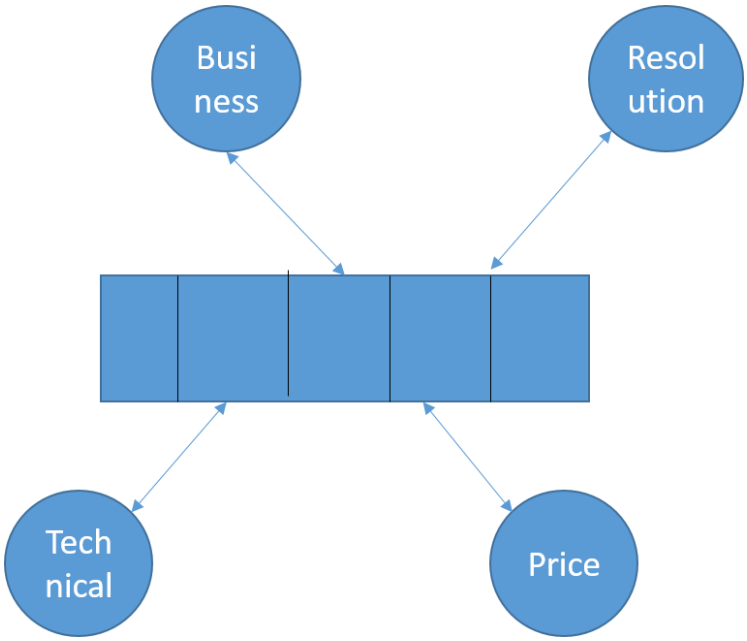
Στην πρώτη περίπτωση γίνεται σειριακή επεξεργασία των δεδομένων από το ένα στάδιο στο επόμενο. Όταν ένα στάδιο τελειώσει την επεξεργασία, τότε και μόνο τότε το επόμενο στάδιο δέχεται σαν είσοδο την έξοδο του προηγούμενου σταδίου και αρχίζει την επεξεργασία των δεδομένων. Αυτό έχει σαν αποτέλεσμα όλα τα στάδια να παραμένουν ανενεργά εκτός από ένα που λειτουργεί. Συνεπώς η συνολική διαδικασία παίρνει αρκετό χρόνο για να εκτελεστεί.



Εικόνα 3 Αρχιτεκτονική σειριακής επεξεργασίας

Στη δεύτερη περίπτωση γίνεται παράλληλη επεξεργασία των δεδομένων από τα στάδια. Το κάθε στάδιο όταν έχει δεδομένα προς επεξεργασία αρχίζει άμεσα να λειτουργεί. Αυτό γίνεται με τη χρησιμοποίηση μιας κύριας ουράς από την οποία όλες οι μονάδες παίρνουν κατάλληλα μηνύματα για να αρχίζουν τη λειτουργία τους. Με την παράλληλη επεξεργασία δεν μένει

κανένα στάδιο ανενεργό, αντίθετα λειτουργούν όλα ταυτόχρονα. Παρόλο που ο χρόνος εκτέλεσης κάθε σταδίου παραμένει ίδιος, με αυτό τον τρόπο μειώνεται σημαντικά ο χρόνος εκτέλεσης της συνολικής διαδικασίας.



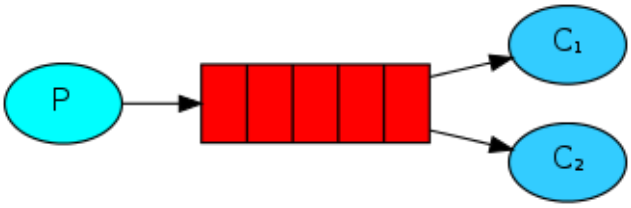
Εικόνα 4 Αρχιτεκτονική παράλληλης επεξεργασίας

6.1 Η ουρά μηνυμάτων RabbitMQ

Η επικοινωνία μεταξύ των σταδίων γίνεται με ουρές μηνυμάτων. Στην παρούσα υλοποίηση χρησιμοποιήθηκε η ουρά μηνυμάτων RabbitMQ. Η RabbitMQ είναι λογισμικό ανοιχτού κώδικα και υλοποιεί το AMQP πρωτόκολλο. Στη RabbitMQ αυτός που αποστέλλει ένα μήνυμα λέγεται παραγωγός (producer, P) ενώ αυτός που λαμβάνει ένα μήνυμα λέγεται καταναλωτής (consumer, C). Τα μηνύματα αποθηκεύονται σε ουρές μηνυμάτων. Ένας παραγωγός λοιπόν στέλνει ένα μήνυμα σε μία ουρά και ένας καταναλωτής λαμβάνει το μήνυμα από την ουρά.

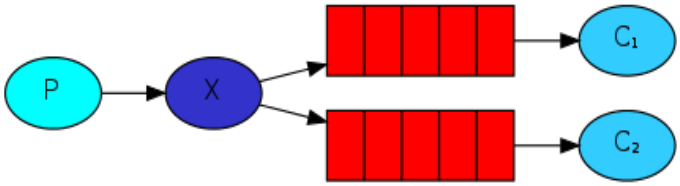
Εικόνα 5 Ουρά με ένα παραγωγό και ένα καταναλωτή

Από μία ουρά όμως μπορούν να λαμβάνουν μηνύματα πολλοί καταναλωτές. Η RabbitMQ υποστηρίζει αυτή τη δυνατότητα.



Εικόνα 6 Ουρά με ένα παραγωγό και δύο καταναλωτές

Στην περίπτωση όμως που χρειάζεται να υπάρχουν πολλοί καταναλωτές μηνυμάτων από την ίδια ουρά, αλλά ο κάθε καταναλωτής να λαμβάνει ένα υποσύνολο μηνυμάτων με συγκεκριμένα χαρακτηριστικά, είναι αναγκαία κάποιου είδους δρομολόγηση των μηνυμάτων. Στη RabbitMQ αυτό γίνεται με το exchange. Στην πραγματικότητα ένας παραγωγός δεν στέλνει μηνύματα σε μία ουρά αλλά σε ένα exchange. Το exchange είναι αυτό που ξέρει τι να κάνει το μήνυμα. Λαμβάνει μηνύματα από παραγωγούς και μπορεί να τα απορρίψει ή να τα στείλει σε μία ή περισσότερες ουρές. Μία ουρά 'δένεται' με ένα exchange, που σημαίνει ότι λαμβάνει μηνύματα από αυτό το exchange. Σε ένα exchange μπορούν να είναι δεμένες πολλές ουρές.



Εικόνα 7 Δρομολόγηση μηνυμάτων σε ουρές

Υπάρχουν διάφοροι τύποι exchange όπως direct, fanout, topic. Το topic exchange δρομολογεί τα μηνύματα στις κατάλληλες ουρές με βάση το θέμα (topic) του μηνύματος. Όταν μία ουρά δένεται με ένα exchange δηλώνεται και το topic. Έτσι το exchange ξέρει που να στείλει τα μηνύματα ανάλογα με το topic τους. Όταν αποστέλλεται ένα μήνυμα, στην πραγματικότητα δεν αποστέλλεται σε μία ουρά αλλά σε ένα exchange. Κατά την αποστολή του μηνύματος λοιπόν δηλώνεται το όνομα του exchange.

6.2 Η βάση δεδομένων MongoDB

Στο κάθε στάδιο ανακτώνται δεδομένα από τη βάση δεδομένων και αποθηκεύονται δεομένα σε αυτή. Η βάση δεδομένων που χρησιμοποιήθηκε είναι η MongoDB. Η MongoDB είναι μία διαπλατφορμική βάση δεδομένων προσανατολισμένη στα έγγραφα. Ανήκει στην κατηγορία

των μη σχεσιακών βάσεων δεδομένων. Αποφεύγει την παραδοσιακή δομή των σχεσιακών βάσεων δεδομένων που είναι βασισμένες στους πίνακες, τύπους εγγράφων όπως JSON με δυναμικά σχήματα κάνοντας την ενσωμάτωση των δεδομένων σε συγκεκριμένους τύπους εφαρμογών πιο εύκολη και πιο γρήγορη. Αντί για πίνακες, χρησιμοποιεί συλλογές (collections). Η MongoDB είναι ελεύθερο λογισμικό, ανοιχτού κώδικα.

6.3 Η γλώσσα προγραμματισμού και η μορφή των αρχείων

Για την υλοποίηση του κάθε σταδίου επεξεργασίας χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python.

Οι αρχικές απαιτήσεις του χρήστη εισάγονται με ένα αρχείο JSON το οποίο αποθηκεύεται στη βάση δεδομένων. Επίσης η λίστα με τις υπηρεσίες cloud και τα χαρακτηριστικά τους είναι σε μορφή JOSN αρχείων αποθηκευμένα στη βάση δεδομένων. Γενικότερα η επικοινωνία του κάθε σταδίου με τη βάση δεδομένων γίνεται μέσω αρχείων JSON.

6.4 Οι υπηρεσίες cloud

Οι υπηρεσίες cloud που παρέχονται στο παράδειγμα του marketplace είναι μία βάση δεδομένων, ένα λειτουργικό σύστημα, ένα σύστημα διαχείρισης περιεχομένου (CMS), και ένα δίκτυο.

Τα τεχνικά χαρακτηριστικά για κάθε υπηρεσία cloud παρατίθενται παρακάτω.

Βάση δεδομένων

- Όνομα: MySQL 5.3 Cluster, Apache Derby, MariaDB, PostgreSQL, IBM DB2, Amazon SimpleDB, Microsoft SQL Server, IBM Informix, Cassandra, MongoDB, MonetDB, CouchDB, DynamoDB, Azure Table Storage, Aerospike, Voldemort, Neo4J, FatDB, CortexDB, Trinity
- Αρχιτεκτονική: x86, x64
- Τύπος: SQL, NoSQL

Λειτουργικό σύστημα

- Διανομή: Debian, RedHat, 8.1, 10, Yosemite, Mavericks, Snow Leopard, OpenSuSe 42.1
- Τύπος: Linux, Windows, MacOS X
- Αρχιτεκτονική: x86, x64

CMS

- Όνομα: Drupal 7, Drupal 8 beta 1, Drupal 9 beta 1, Joomla 3, WordPress 3.2.1, WordPress 3.2
- Αρχιτεκτονική: x86, x64

Δίκτυο

- Όνομα: ADSL Connection, VDSL Connection, T3 Connection, ADSL2+ Connection
- Τύπος: copper, fiber

Τα επιχειρηματικά χαρακτηριστικά των υπηρεσιών cloud παρατίθενται παρακάτω.

Βάση δεδομένων

- Πάροχος: Oracle Provider, Amazon, SalesForge, Cloudera, VMWare, RackSpace, RedHat, IBM, Microsoft, Citrix, FaceBook, 10Gen, BlueHost, OpenShift, Google
- Διαθεσιμότητα: Ακέραιος θετικός αριθμός
- Ασφάλεια: Βαθμολογημένη κλίμακα ακέραιων αριθμών από το 1 μέχρι το 5
- Φήμη: Βαθμολογημένη κλίμακα ακέραιων αριθμών από το 1 μέχρι το 5
- Αξιοπιστία: Βαθμολογημένη κλίμακα ακέραιων αριθμών από το 1 μέχρι το 5
- Εμπιστευτικότητα: Ναι, Όχι
- Τιμή: Πραγματικός θετικός αριθμός

Λειτουργικό σύστημα

- Πάροχος: Amazon, RedHat, SalesForge, Microsoft, VMWare
- Διαθεσιμότητα: Ακέραιος θετικός αριθμός
- Ασφάλεια: Βαθμολογημένη κλίμακα ακέραιων αριθμών από το 1 μέχρι το 5
- Φήμη: Βαθμολογημένη κλίμακα ακέραιων αριθμών από το 1 μέχρι το 5
- Αξιοπιστία: Βαθμολογημένη κλίμακα ακέραιων αριθμών από το 1 μέχρι το 5
- Εμπιστευτικότητα: Ναι, Όχι
- Τιμή: Πραγματικός θετικός αριθμός

CMS

- Πάροχος: Drupal Inc, HostGator, BlueHost, FatCow, iPage
- Διαθεσιμότητα: Ακέραιος θετικός αριθμός
- Ασφάλεια: Βαθμολογημένη κλίμακα ακέραιων αριθμών από το 1 μέχρι το 5
- Φήμη: Βαθμολογημένη κλίμακα ακέραιων αριθμών από το 1 μέχρι το 5
- Αξιοπιστία: Βαθμολογημένη κλίμακα ακέραιων αριθμών από το 1 μέχρι το 5

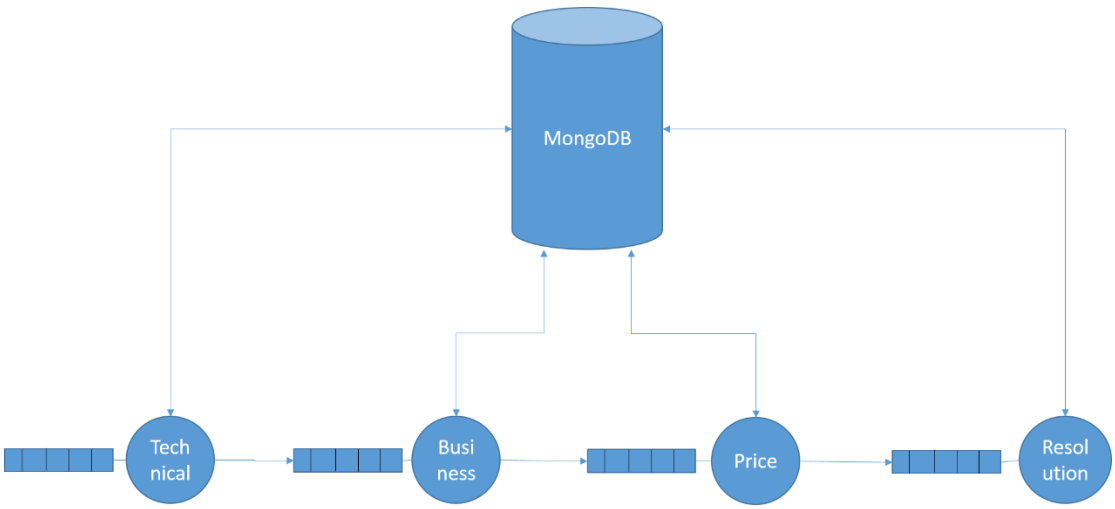
- Εμπιστευτικότητα: Ναι, Όχι
- Τιμή: Πραγματικός θετικός αριθμός

Δίκτυο

- Πάροχος: CosmOTE, T-Mobile, Orange, Vodafone, Telefonica
- Διαθεσιμότητα: Ακέραιος θετικός αριθμός
- Ασφάλεια: Βαθμολογημένη κλίμακα ακέραιων αριθμών από το 1 μέχρι το 5
- Φήμη: Βαθμολογημένη κλίμακα ακέραιων αριθμών από το 1 μέχρι το 5
- Αξιοπιστία: Βαθμολογημένη κλίμακα ακέραιων αριθμών από το 1 μέχρι το 5
- Εμπιστευτικότητα: Ναι, Όχι
- Τιμή: Πραγματικός θετικός αριθμός

6.5 *Αρχιτεκτονική της σειριακής επεξεργασίας*

Στο παρακάτω σχήμα παρουσιάζεται η αρχιτεκτονική της πρώτης περίπτωσης του marketplace. Το σχήμα αυτό αποτελείται από τέσσερεις μονάδες επεξεργασίας. Το technical, το business, το price και το resolution. Η κάθε μονάδα είναι δεμένη με μία ουρά από την οποία λαμβάνει μηνύματα. Τέλος υπάρχει και η βάση δεδομένων.



Εικόνα 8 Αρχιτεκτονική σειριακής επεξεργασίας

Όπως έχει αναφερθεί, η διαδικασία που ακολουθείται είναι σειριακή. Δηλαδή, όταν αρχικά λειτουργεί το technical, οι υπόλοιπες μονάδες περιμένουν. Όταν τελειώσει ο technical στέλνει μήνυμα στην ουρά του business, ο business καταναλώνει το μήνυμα και αρχίζει τη

λειτουργία. Με τον ίδιο τρόπο συνεχίζεται η λειτουργία του συστήματος μέχρι να τελειώσει και ο resolution.

Η κάθε μονάδα στέλνει μήνυμα στην ουρά της επόμενης. Ως εκ τούτου, δεν υπάρχει η ανάγκη για δρομολόγηση των μηνυμάτων. Ωστόσο, το exchange υπάρχει στη RabbitMQ. Όταν δε δηλωθεί το όνομα του exchange κατά την αποστολή του μηνύματος θεωρείται το default exchange που απλά προωθεί το μήνυμα στην ουρά που έχει δεθεί με αυτό το exchange. Σε αυτή την περίπτωση δηλαδή το exchange δεν παίζει κάποιο ρόλο και γι' αυτό το λόγο έχει παραληφθεί από το σχήμα.

Η πληροφορία για τις διαθέσιμες cloud υπηρεσίες, για τα technical και business χαρακτηριστικά τους και για το κόστος τους είναι αποθηκευμένη σε αρχεία μορφής JSON στη βάση δεδομένων.

(ListOfTechnicalResolutionBlueprints.json, ListOfBusinessResolutionProducts.json). Επίσης σε μορφή JOSN είναι αποθηκευμένες οι απαιτήσεις του χρήστη.

(UserRequirements.json).

Η κάθε μονάδα αποτελείται από ένα python script. Ακολουθεί περιγραφή του κώδικα.

6.6 Περιγραφή του κώδικα για την περίπτωση της σειριακής επεξεργασίας

6.6.1 Technical resolution

Γίνεται σύνδεση με τη RabbitMQ και ανοίγεται κανάλι επικοινωνίας. Δηλώνεται η ουρά business.

Γίνεται σύνδεση με το MongoClient και με τη βάση δεδομένων. Φέρνει από τη βάση τα έγγραφα ListOfTechnicalResolutionBlueprints.json, UserRequirements.json.

Για κάθε ένα στοιχείο της λίστας των υπηρεσιών με τα technical χαρακτηριστικά ελέγχει αν ικανοποιούνται οι απαιτήσεις του χρήστη και αν ισχύει αυτή η συνθήκη τότε δημιουργεί ένα καινούριο αρχείο JSON που αρχικά έχει τις απαιτήσεις του χρήστη και στη συνέχεια τα συγκεκριμένα τεχνικά χαρακτηριστικά τις υπηρεσίας. Με αυτό τον τρόπο δημιουργείται ένα αρχείο για κάθε δυνατό συνδυασμό ενός συνόλου διαφορετικών υπηρεσιών που έχει ζητήσει ο χρήστης, οι οποίες καλύπτουν τις απαιτήσεις του. Κάθε αρχείο που δημιουργείται αποθηκεύεται σε συγκεκριμένο collection (technical resolution collection) της βάσης δεδομένων.

Όταν τελειώσει αυτή η διαδικασία ο technical στέλνει μήνυμα στην ουρά business.

6.6.2 Business resolution

Γίνεται σύνδεση με τη RabbitMQ και ανοίγεται κανάλι επικοινωνίας. Δηλώνεται η ουρά business και η ουρά price.

Γίνεται σύνδεση με το MongoClient και με τη βάση δεδομένων. Φέρνει από τη βάση τα έγγραφα ListOfBusinessResolutionBlueprints.json, UserRequirements.json.

Περιμένει για μηνύματα και όταν καταναλώσει μήνυμα από την ουρά business κάνει ένα query στη βάση δεδομένων και φέρνει από το technical resolution collection τα αποτελέσματα από το προηγούμενο στάδιο. Για κάθε ένα σύνολο υπηρεσιών cloud δημιουργεί όλους τους συνδυασμούς με τα επιχειρηματικά χαρακτηριστικά.

Για κάθε μία υπηρεσία cloud υπάρχουν διάφοροι πάροχοι που την προσφέρουν με διαφορετικά επιχειρηματικά χαρακτηριστικά ο καθένας από αυτούς. Έτσι, για κάθε υπηρεσία ενός συνόλου υπηρεσιών που καλύπτουν τα τεχνικά χαρακτηριστικά δημιουργούνται όλοι οι συνδυασμοί με τους διαφορετικούς πάροχους με τα επιχειρηματικά χαρακτηριστικά. Αυτό γίνεται για όλες τις υπηρεσίες του συνόλου.

Όπως και στο προηγούμενο στάδιο, όταν φτιάχνεται ένα σύνολο με τις υπηρεσίες, δημιουργείται ένα JSON αρχείο και μαζί με τις απαιτήσεις και τα τεχνικά χαρακτηριστικά προστίθενται τα επιχειρηματικά χαρακτηριστικά. Σημειώνεται ότι στα επιχειρηματικά χαρακτηριστικά συμπεριλαμβάνεται και το κόστος κάθε υπηρεσίας, το οποίο είναι διαφορετικό από πάροχο σε πάροχο. Κάθε αρχείο που δημιουργείται αποθηκεύεται σε συγκεκριμένο collection (business resolution collection) στη βάση δεδομένων.

Κάθε φορά που γίνεται επεξεργασία ενός εγγράφου από το technical resolution collection, μετά την επεξεργασία διαγράφεται από το τη βάση δεδομένων. Όταν τελειώσει αυτή η διαδικασία, το technical resolution collection δεν περιέχει κανένα έγγραφο. Ο business στέλνει μήνυμα στην ουρά price.

6.6.3 Price aggregator

Γίνεται σύνδεση με τη RabbitMQ και ανοίγεται κανάλι επικοινωνίας. Δηλώνεται η ουρά η ουρά price και η ουρά resolution.

Γίνεται σύνδεση με το MongoClient και με τη βάση δεδομένων. Φέρνει από τη βάση το έγγραφο UserRequirements.json.

Περιμένει για μηνύματα και όταν καταναλώσει μήνυμα από την ουρά price κάνει ένα query στη βάση δεδομένων και φέρνει από το business resolution collection τα αποτελέσματα από το προηγούμενο στάδιο. Αυτό που κάνει είναι να ελέγχει αν το σύνολο των υπηρεσιών κοστίζει περισσότερο από το μέγιστο κόστος που έχει οριστεί στις απαιτήσεις του χρήστη.

Για κάθε σύνολο υπηρεσιών cloud που έχει παραχθεί από τον business, προσθέτει το κόστος κάθε ξεχωριστής υπηρεσίας. Αν το συνολικό κόστος δεν ικανοποιεί τις απαιτήσεις τότε απορρίπτει αυτό το σύνολο. Σε περίπτωση που το συνολικό κόστος είναι μικρότερο ή ίσο με αυτό ορισμένο στις απαιτήσεις τότε δημιουργείται ένα αρχείο JSON με τις απαιτήσεις του χρήστη, το σύνολο των υπηρεσιών, τα τεχνικά χαρακτηριστικά, τα επιχειρηματικά χαρακτηριστικά και στο τέλος προστίθεται το συνολικό κόστος των υπηρεσιών αυτών. Κάθε τέτοιο αρχείο αποθηκεύεται σε συγκεκριμένο collection (price resolution collection) στη βάση δεδομένων.

Μετά την επεξεργασία ενός εγγράφου από το business resolution collection, διαγράφεται το έγγραφο από τη βάση δεδομένων. Στο τέλος του τρίτου σταδίου το business resolution collection είναι άδειο. Ο price στέλνει μήνυμα στην ουρά resolution.

6.6.4 Final resolution

Γίνεται σύνδεση με τη RabbitMQ και ανοίγεται κανάλι επικοινωνίας. Δηλώνεται η ουρά η ουρά η ουρά resolution.

Γίνεται σύνδεση με το MongoClient και με τη βάση δεδομένων. Φέρνει από τη βάση το έγγραφο UserRequirements.json.

Περιμένει για μηνύματα και όταν καταναλώσει μήνυμα από την ουρά business κάνει ένα query στη βάση δεδομένων και φέρνει από το business resolution collection τα αποτελέσματα από το προηγούμενο στάδιο. Η λειτουργία του είναι να απορρίψει τα σύνολα των υπηρεσιών cloud που δεν ικανοποιούν τις επιχειρηματικές απαιτήσεις του χρήστη. Επίσης για τα σύνολα υπηρεσιών που δεν απορρίπτονται να τα βαθμολογήσει για να μπορούν να ταξινομηθούν σε μία φθίνουσα σειρά αρχίζοντας από τις από τις βέλτιστες επιλογές με βάση τις ανάγκες του χρήστη σύμφωνα με τις απαιτήσεις που έχει ορίσει.

Παίρνει τα έγγραφα από το price resolution collection. Για κάθε ένα έγγραφο εξετάζεται αν ικανοποιούνται οι επιχειρηματικές απαιτήσεις του χρήστη ως εξής. Για τα χαρακτηριστικά που αφορούν τη διαθεσιμότητα, την ασφάλεια τη φήμη και την αξιοπιστία υπολογίζεται ο μέσος όρος όλων των υπηρεσιών. Αν ο μέσος όρος για ένα από αυτά τα χαρακτηριστικά είναι μικρότερος από την αντίστοιχη απαίτηση του χρήστη τότε το σύνολο των υπηρεσιών απορρίπτεται. Όσο αφορά την εμπιστευτικότητα, οι τιμές που μπορεί να πάρει είναι ΝΑΙ και ΟΧΙ. Οι τιμές της εμπιστευτικότητας μετατρέπονται σε ακέραιοι σύμφωνα με τον ακόλουθο τρόπο. Η τιμή ΝΑΙ γίνεται 1 και η τιμή ΟΧΙ γίνεται 0. Στη συνέχεια υπολογίζεται ο μέσος όρος. Αν η απαίτηση για την εμπιστευτικότητα έχει την τιμή ΝΑΙ και ο μέσος όρος των υπηρεσιών είναι μικρότερος από 0.5 τότε το σύνολο απορρίπτεται. Αν η απαίτηση για την εμπιστευτικότητα έχει την τιμή ΟΧΙ τότε το σύνολο δεν απορρίπτεται ανεξάρτητα από το μέσο όρο της εμπιστευτικότητας των υπηρεσιών. Οι μέσοι όροι των επιχειρηματικών

χαρακτηριστικών ονομάζονται συνολική διαθεσιμότητα, συνολική ασφάλεια, συνολική φήμη, συνολική αξιοπιστία, συνολική εμπιστευτικότητα.

Για τα σύνολα που δεν απορρίφθηκαν υπολογίζεται ένας βαθμό αξιολόγησης. Αυτό γίνεται σύμφωνα με την ακόλουθη συνάρτηση.

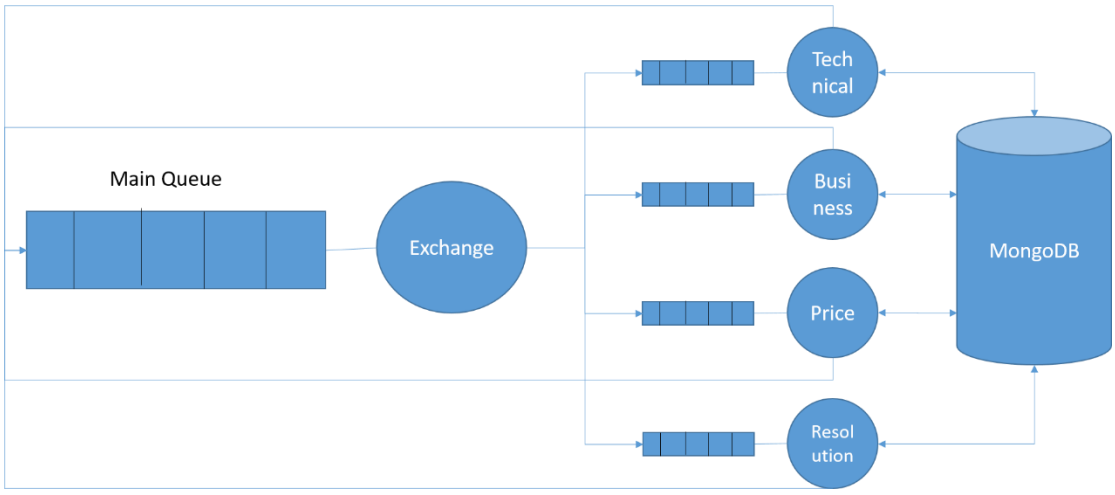
Βαθμός = 0.3*συνολική διαθεσιμότητα + 0.6*συνολική ασφάλεια + 0.5*συνολική φήμη + 0.7*συνολική αξιοπιστία + 0.5*συνολική εμπιστευτικότητα - 0.005*συνολικό κόστος

Για κάθε σύνολο υπηρεσιών cloud που δεν απορρίφθηκε από τον final resolution δημιουργείται ένα JSON αρχείο με τις απαιτήσεις του χρήστη, τις υπηρεσίες, τα τεχνικά και επιχειρηματικά χαρακτηριστικά τους, το συνολικό κόστος και τα συνολικά επιχειρηματικά χαρακτηριστικά και το βαθμό του συνόλου. Το αρχείο αποθηκεύεται σε συγκεκριμένο collection (final resolution collection) στη βάση δεδομένων.

Μετά την επεξεργασία ενός εγγράφου από το price resolution collection, διαγράφεται το έγγραφο από τη βάση δεδομένων. Στο τέλος του τρίτου σταδίου το price resolution collection είναι άδειο.

6.7 *Αρχιτεκτονική της παράλληλης επεξεργασίας*

Στο παρακάτω σχήμα παρουσιάζεται η αρχιτεκτονική της δεύτερης περίπτωσης του marketplace. Το σχήμα αυτό αποτελείται από μία κύρια ουρά μηνυμάτων, ένα exchange και από τέσσερεις μονάδες επεξεργασίας. Το technical, το business, το price και το resolution. Η κάθε μονάδα είναι δεμένη με μία ουρά από την οποία λαμβάνει μηνύματα. Τέλος υπάρχει και η βάση δεδομένων.



Εικόνα 9 Αρχιτεκτονική παράλληλης επεξεργασίας

Όπως έχει αναφερθεί, το σύστημα λειτουργεί με παράλληλη επεξεργασία. Αυτό σημαίνει και οι τέσσερεις μονάδες λειτουργούν παράλληλα. Οι μονάδες επικοινωνούν με μηνύματα μέσω των ουρών. Οι κάθε μονάδα στέλνει μήνυμα στην κύρια ουρά και μέσω του exchange και ανάλογα με το topic του μηνύματος δρομολογείται στην ουρά της κατάλληλης μονάδας. Δηλαδή η κάθε μονάδα καταναλώνει μηνύματα από την κύρια ουρά παράλληλα με τις άλλες μονάδες.

Αρχικά, οι μονάδες είναι ενεργές και περιμένουν να καταναλώσουν μήνυμα από την αντίστοιχη ουρά τους και να αρχίσουν τη λειτουργία τους. Σε πρώτη φάση, αρχίζει να λειτουργεί ο technical. Όταν παραχθεί ένα αποτέλεσμα στέλνει μήνυμα στην κύρια ουρά και μέσω του exchange το μήνυμα δρομολογείται στον business, ο οποίος αρχίζει τη λειτουργία. Όταν παραχθεί αποτέλεσμα από τον business αποστέλλεται μήνυμα στην κύρια ουρά και το καταναλώνει ο price ο οποίος αρχίζει τη λειτουργία του. Αντίστοιχα πραγματοποιείται η ίδια διαδικασία με τον price και το final resolution. Η κύρια ουρά περιέχει μηνύματα για τις μονάδες και αυτές, παράλληλα μεταξύ τους, λειτουργούν και καταναλώνουν μηνύματα από την κύρια ουρά.

Η πληροφορία για τις διαθέσιμες cloud υπηρεσίες, για τα technical και business χαρακτηριστικά τους και για το κόστος τους είναι αποθηκευμένη σε αρχεία μορφής JSON στη βάση δεδομένων.

(ListOfTechnicalResolutionBlueprints.json, ListOfBusinessResolutionProducts.json). Επίσης σε μορφή JOSN είναι αποθηκευμένες οι απαιτήσεις του χρήστη.

(UserRequirements.json).

Η κάθε μονάδα αποτελείται από ένα python script. Ακολουθεί περιγραφή του κώδικα.

6.8 Περιγραφή του κώδικα για την περίπτωση της παράλληλης επεξεργασίας

Ο κώδικας προφανώς είναι σχεδόν ίδιος αφού δεν αλλάζει ο τρόπος επεξεργασίας των δεδομένων και η διαδικασία εξαγωγής των αποτελεσμάτων. Παρακάτω αναφέρονται οι επιπρόσθετες λειτουργίες σε κάθε στάδιο.

6.8.1 Technical resolution

Δηλώνεται το exchange και ο τύπος του exchange (topic). Όταν δημιουργείται ένα αρχείο JSON σαν έξοδος του technical resolution, αποστέλλεται μήνυμα στην κύρια ουρά με topic ‘technical’.

6.8.2 Business resolution

Δηλώνεται το exchange και ο τύπος του exchange (topic). Δένεται η ουρά του business resolution με το exchange και δηλώνεται ότι καταναλώνει μόνο μηνύματα με topic ‘technical’. Όταν δημιουργείται ένα αρχείο JSON σαν έξοδος του business resolution, αποστέλλεται μήνυμα στην κύρια ουρά με topic ‘business.

6.8.3 Price aggregator

Δηλώνεται το exchange και ο τύπος του exchange (topic). Δένεται η ουρά του price aggregator με το exchange και δηλώνεται ότι καταναλώνει μόνο μηνύματα με topic ‘business’. Όταν δημιουργείται ένα αρχείο JSON σαν έξοδος του price aggregator, αποστέλλεται μήνυμα στην κύρια ουρά με topic ‘price.

6.8.4 Final resolution

Δηλώνεται το exchange και ο τύπος του exchange (topic). Δένεται η ουρά του final resolution με το exchange και δηλώνεται ότι καταναλώνει μόνο μηνύματα με topic ‘price’.

7

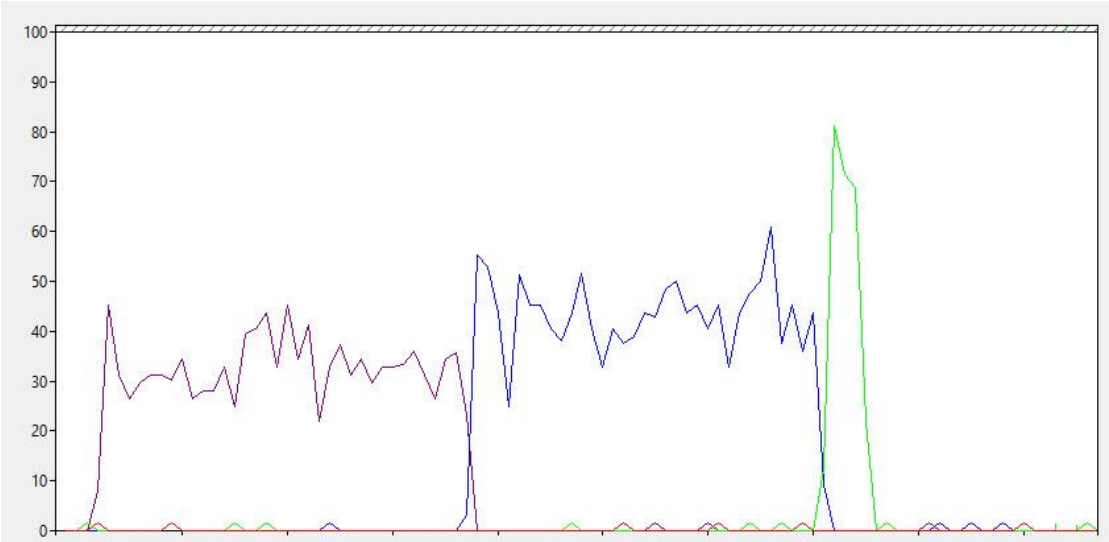
Μετρήσεις και Συμπεράσματα

7.1 Μετρήσεις

Για τη σύγκριση των δύο αρχιτεκτονικών πραγματοποιήθηκαν μετρήσεις με δύο διαφορετικούς τρόπους. Στην πρώτη περίπτωση έγινε εισαγωγή τεχνητής καθυστέρησης στα διάφορα στάδια ώστε να προσομοιωθεί επεξεργασία που παίρνει περισσότερο χρόνο. Στη δεύτερη περίπτωση έγινε παραμετροποίηση στις απαιτήσεις του χρήστη. Έγιναν πιο χαμηλές σε διάφορα σημεία τους ώστε να παράγονται περισσότερα αρχεία στα αντίστοιχα στάδια και να στέλνονται περισσότερα μηνύματα στις ουρές.

Αυτό που μετρήθηκε είναι η χρήση του επεξεργαστή και ο χρόνος εκτέλεσης κάθε σταδίου καθώς και της συνολικής διαδικασίας. Η χρησιμοποίηση της μνήμης παραμένει σταθερή και η μέτρησή της δεν οδηγεί στην εξαγωγή κάποιου συμπεράσματος.

Στη συνέχεια φαίνεται η χρήση του επεξεργαστή και ο χρόνος εκτέλεσης χωρίς εισαγωγή καθυστέρησης και χωρίς παραμετροποίηση των απαιτήσεων χρήστη, για τη σειριακή αρχιτεκτονική.



Εικόνα 10 Χρήση του επεξεργαστή – Σειριακή επεξεργασία

Στο σχήμα φαίνεται η χρήση του επεξεργαστή κατά τη διάρκεια της διαδικασίας από τα τέσσερα στάδια. Με κόκκινο χρώμα είναι το Technical, με μωβ το Business, με μπλε το Price και με πράσινο το Resolution.

Όπως φαίνεται, ο Technical τελειώνει γρήγορα τη λειτουργία του και δε χρησιμοποιεί σχεδόν καθόλου τον επεξεργαστή. Αντίθετα ο Business και ο Price λειτουργούν περίπου τον ίδιο χρόνο και χρησιμοποιούν περίπου το 50% του επεξεργαστή. Τέλος ο Resolution λειτουργεί αρκετά λιγότερο από το Business και Resolution αλλά χρησιμοποιεί περισσότερο τον επεξεργαστή. Στο σήμα φαίνεται η σειριακή επεξεργασία αφού το κάθε στάδιο αρχίζει όταν τελειώνει το προηγούμενο.

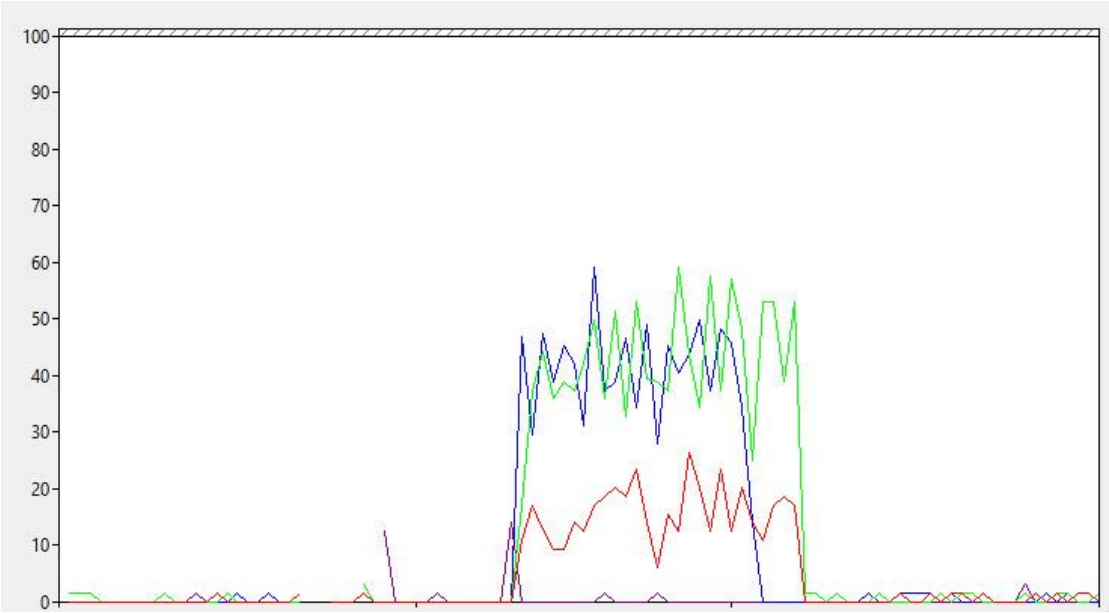
Πίνακας 1 Χρήση επεξεργαστή, χρόνος επεξεργαστή - Σειριακή επεξεργασία

	Average CPU Usage (%)	**CPU Time (Seconds)**
Technical	0.3	1.52
Business	35	34.31
Price	46	36.74
Resolution	78	4.23
Total	-	74.62

Στον πίνακα παρουσιάζεται η μέση χρήση του επεξεργαστή για κάθε στάδιο. Επιπλέον παρουσιάζεται ο χρόνος εκτέλεσης για κάθε στάδιο και ο συνολικός χρόνος εκτέλεσης. Παρατηρούμε ότι ο Technical χρειάζεται πολύ λίγο χρόνο σε σχέση με το Business και το

Price που χρειάζονται περισσότερο και περίπου τον ίδιο μεταξύ τους. Ο Resolution χρειάζεται αρκετά λιγότερο από το Business και Resolution και λίγο περισσότερο από το Technical.

Ακολουθεί η χρήση του επεξεργαστή και ο χρόνος εκτέλεσης χωρίς εισαγωγή καθυστέρησης και χωρίς παραμετροποίηση των απαιτήσεων χρήστη, για την παράλληλη αρχιτεκτονική.



Εικόνα 11 Χρήση επεξεργαστή - Παράλληλη επεξεργασία

Στο σχήμα φαίνεται η χρήση του επεξεργαστή κατά τη διάρκεια της διαδικασίας από τα τέσσερα στάδια. Με μωβ χρώμα είναι το Technical, με μπλε το Business, με πράσινο το Price και με κόκκινο το Resolution.

Σε αυτό το σχήμα φαίνεται η παράλληλη επεξεργασία αφού όλα τα στάδια ξεκινούν σχεδόν ταυτόχρονα και τελειώνουν σχεδόν μαζί τη λειτουργία τους. Ο Resolution κάνει αισθητά μικρότερη χρήση του επεξεργαστή σε σχέση με το Business και το Price, ενώ ο Technical τον χρησιμοποιεί ελάχιστα.

Εξίσωση 1 Χρήση επεξεργαστή, χρόνος επεξεργαστή - Παράλληλη επεξεργασία

	Average CPU Usage (%)	**CPU Time (Seconds)**
Technical	16	1.52
Business	48	23.45
Price	56	28.07

Resolution	19	27.94
Total	-	28.15

Αντίστοιχα ο πίνακας με τη μέση χρήση επεξεργαστή και τους χρόνους εκτέλεσης. Όλα τα στάδια κάνουν περίπου τον ίδιο χρόνο, κάτι που είναι λογικά αφού γίνεται παράλληλη επεξεργασία. Μόνο ο Technical τελειώνει πολύ γρήγορα τη λειτουργία του.

Στη συνέχεια αναφέρονται οι μετρήσεις για την πρώτη περίπτωση, με την εισαγωγή καθυστέρησης.

7.2 Μετρήσεις με εισαγωγή καθυστέρησης

Λήφθηκαν μετρήσεις για τέσσερεις διαφορετικές περιπτώσεις.

Α) Στην πρώτη περίπτωση εισήχθη καθυστέρηση στο Technical με τον εξής τρόπο. Για κάθε έλεγχο του συνόλου υπηρεσιών για τον αν πληρούν τα τεχνικά χαρακτηριστικά εισήχθη καθυστέρηση 0.5 δευτερολέπτου.

Β) Στη δεύτερη περίπτωση εισήχθη καθυστέρηση στο Business με τον εξής τρόπο. Στη δημιουργία κάθε δυνατού συνδυασμού συνόλων υπηρεσιών με βάση τα επιχειρηματικά χαρακτηριστικά εισήχθη καθυστέρηση 0.1 δευτερολέπτου.

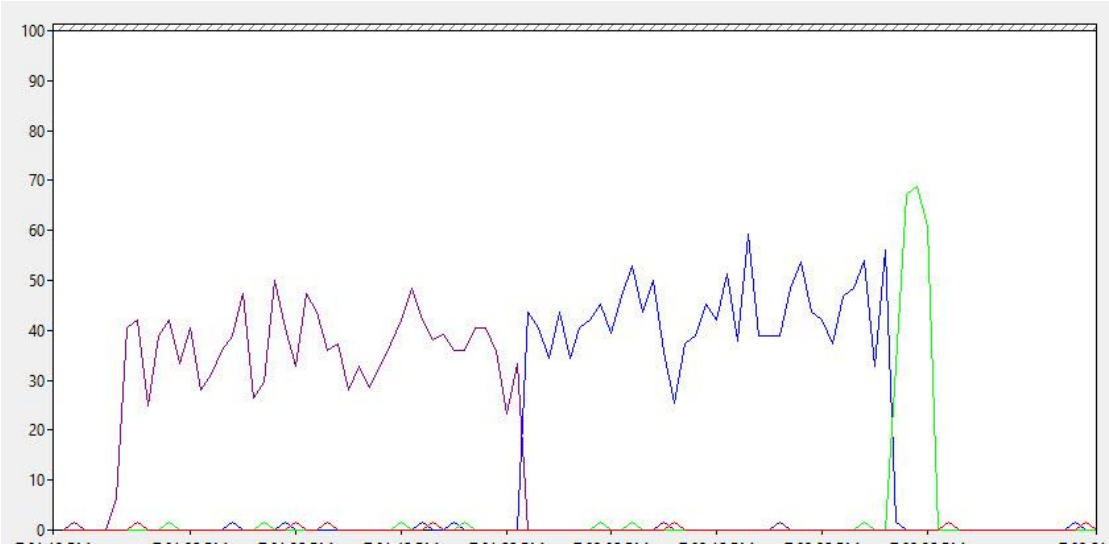
Γ) Στην Τρίτη περίπτωση εισήχθη καθυστέρηση στο Resolution με τον εξής τρόπο. Για κάθε σύνολο υπηρεσιών που δεν απορρίπτεται εισήχθη καθυστέρηση ενός δευτερολέπτου.

Δ) Στην τέταρτη περίπτωση εφαρμόστηκαν όλες οι καθυστερήσεις των προηγούμενων περιπτώσεων μαζί.

Ακολουθούν οι αντίστοιχες μετρήσεις.

7.2.1 Μετρήσεις με εισαγωγή καθυστέρησης – Περίπτωση Α

Σειριακή Αρχιτεκτονική



Εικόνα 12 Χρήση επεξεργαστή, σειριακή επεξεργασία - εισαγωγή καθυστέρησης περίπτωση Α

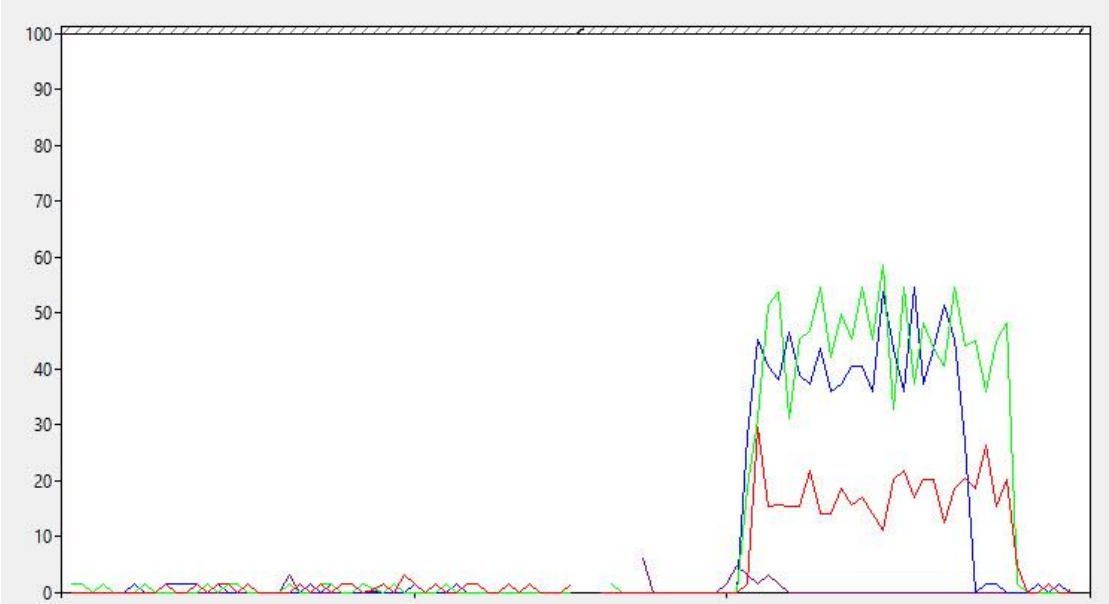
Παρόλη την καθυστέρηση στο Technical δε βλέπουμε αλλαγή στο σχήμα της χρήσης του επεξεργαστή.

Πίνακας 2 Χρήση επεξεργαστή, χρόνος επεξεργαστή, σειριακή επεξεργασία - εισαγωγή καθυστέρησης περίπτωση Α

	Average CPU Usage (%)	**CPU Time (Seconds)**
Technical	0.4	17.72
Business	41	23.45
Price	52	28.07
Resolution	64	4.17
Total	-	91.24

Παρατηρείται αύξηση στο χρόνο εκτέλεσης του Technical και στο συνολικό χρόνο εκτέλεσης.

Παράλληλη Αρχιτεκτονική



Εικόνα 13 Χρήση επεξεργαστή, παράλληλη επεξεργασία - εισαγωγή καθυστέρησης περίπτωση Α

Δεν παρατηρείται διαφορά στην χρήση επεξεργαστή.

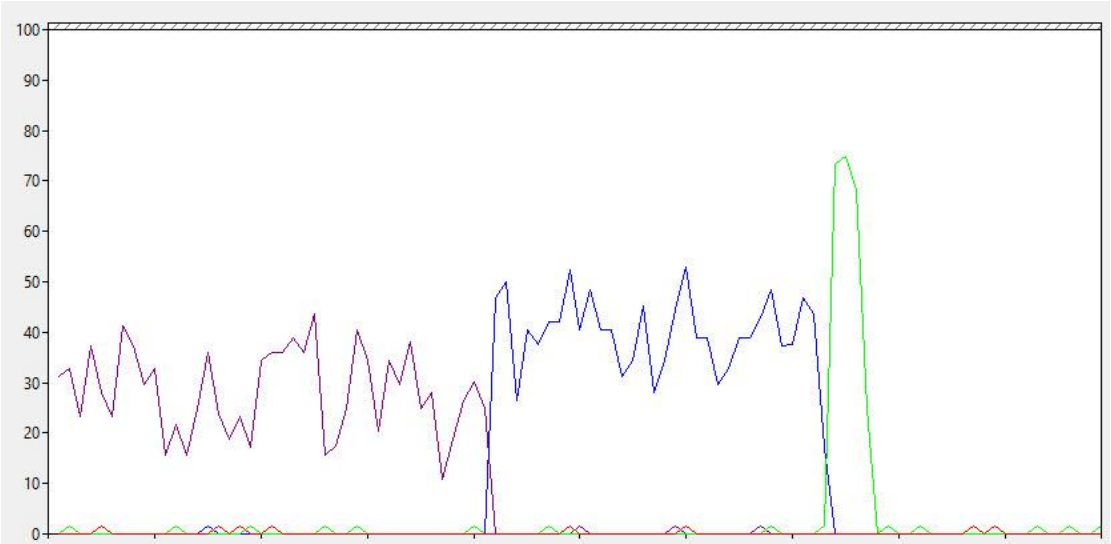
Πίνακας 3 Χρήση επεξεργαστή, χρόνος επεξεργαστή, παράλληλη επεξεργασία - εισαγωγή καθυστέρησης περίπτωση Α

	Average CPU Usage (%)	**CPU Time (Seconds)**
Technical	9	2.92
Business	47	23.66
Price	53	28.07
Resolution	22	27.84
Total	-	28.31

Παρατηρούμε αύξηση του χρόνου εκτέλεσης του Technical αλλά ο συνολικός χρόνος εκτέλεσης επηρεάστηκε ελάχιστα.

7.2.2 Μετρήσεις με εισαγωγή καθυστέρησης – Περίπτωση Β

Σειριακή Αρχιτεκτονική



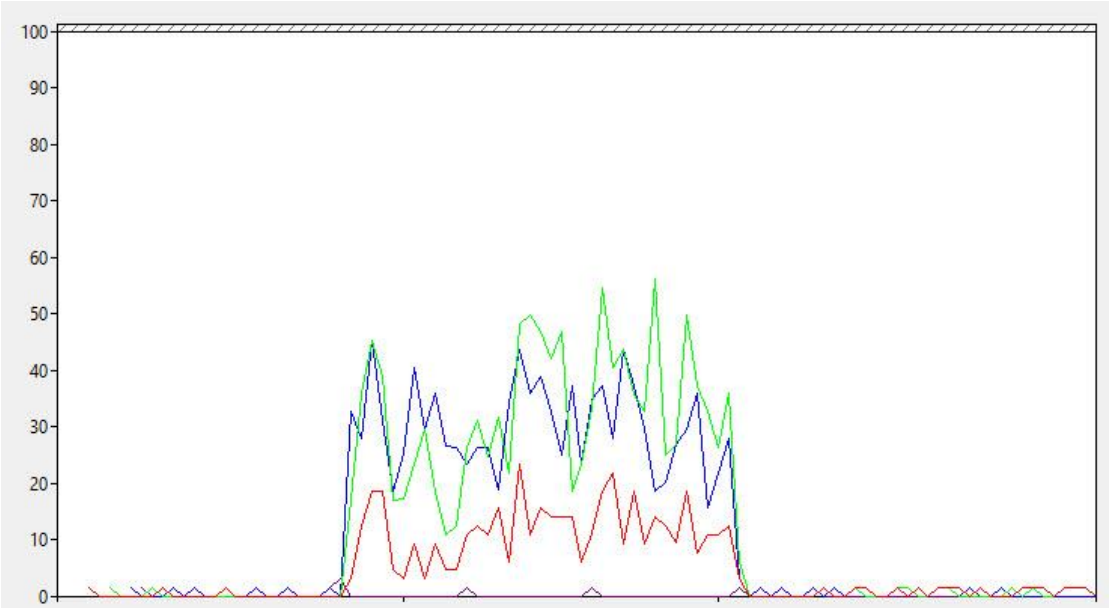
Εικόνα 14 Χρήση επεξεργαστή, σειριακή επεξεργασία - εισαγωγή καθυστέρησης περίπτωση Β

Πίνακας 4 Χρήση επεξεργαστή, χρόνος επεξεργαστή, σειριακή επεξεργασία - εισαγωγή καθυστέρησης περίπτωση Β

	Average CPU Usage (%)	**CPU Time (Seconds)**
Technical	0.3	1.58
Business	36	54.22
Price	41	36.74
Resolution	79	4.73
Total	-	93.51

Παρατηρείται αύξηση στο χρόνο εκτέλεσης του Business και του συνολικού χρόνου.

Παράλληλη Αρχιτεκτονική



Εικόνα 15 Χρήση επεξεργαστή, παράλληλη επεξεργασία - εισαγωγή καθυστέρησης περίπτωση Β

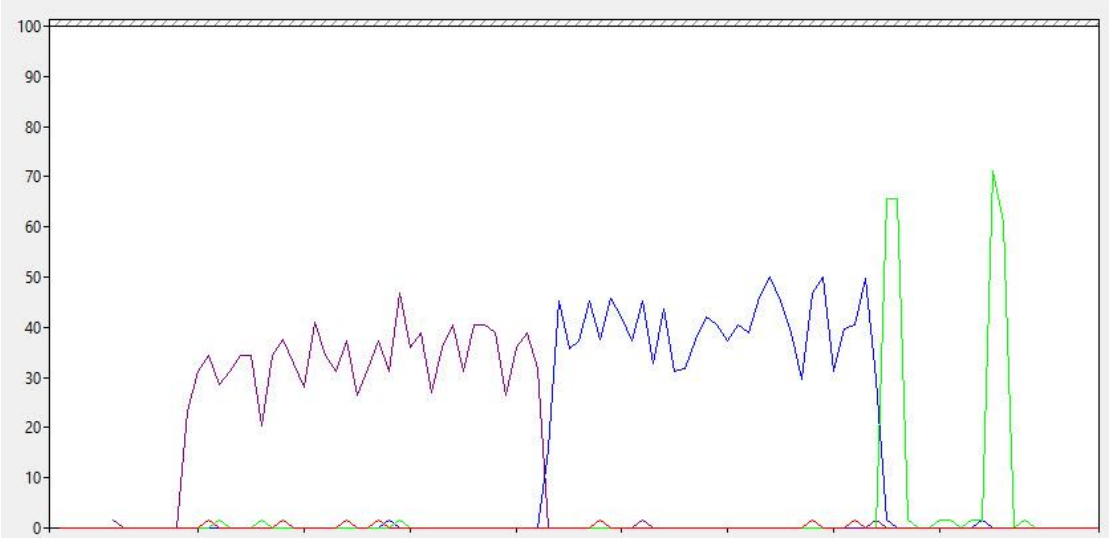
Πίνακας 5 Χρήση επεξεργαστή, χρόνος επεξεργαστή, παράλληλη επεξεργασία - εισαγωγή καθυστέρησης περίπτωση Β

	Average CPU Usage (%)	**CPU Time (Seconds)**
Technical	3	1.49
Business	41	38.37
Price	50	38.16
Resolution	20	37.88
Total	-	38.89

Παρατηρούμε αύξηση σε όλα τα στάδια εκτός του πρώτου που έχει μικρή λειτουργία. Ως εκ τούτου υπάρχει αύξηση στο συνολικό χρόνο εκτέλεσης.

7.2.3 Μετρήσεις με εισαγωγή καθυστέρησης – Περίπτωση Γ

Σειριακή Αρχιτεκτονική



Εικόνα 16 Χρήση επεξεργαστή, σειριακή επεξεργασία - εισαγωγή καθυστέρησης περίπτωση Γ

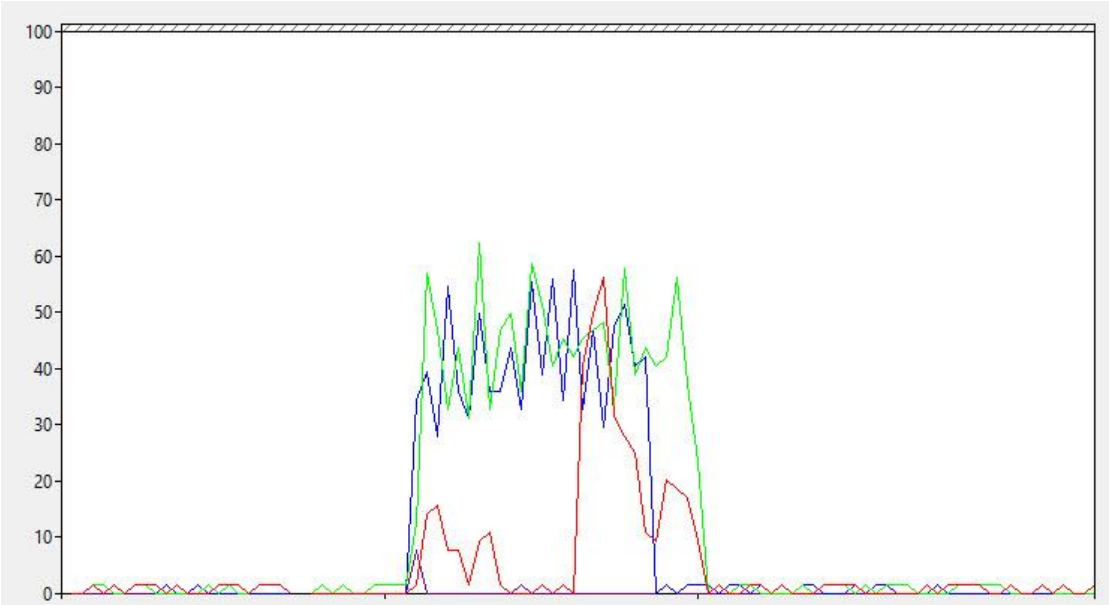
Παρατηρούνται δύο μεγάλα spikes στη χρήση επεξεργαστή από το Resolution. Αυτό συμβαίνει γιατί η καθυστέρηση έχει εισαχθεί μόνο όταν δημιουργείται σύνολο υπηρεσιών που καλύπτει τις απαιτήσεις και όχι για όλες τις άλλες φορές που απορρίπτεται.

Πίνακας 6 Χρήση επεξεργαστή, χρόνος επεξεργαστή, σειριακή επεξεργασία - εισαγωγή καθυστέρησης περίπτωση Γ

	Average CPU Usage (%)	**CPU Time (Seconds)**
Technical	0.4	1.51
Business	39	34.37
Price	49	36.64
Resolution	68	11.15
Total	-	81.18

Παρατηρείται αύξηση στο χρόνο εκτέλεσης στο Resolution και στο συνολικό χρόνο εκτέλεσης.

Παράλληλη Αρχιτεκτονική



Εικόνα 17 Χρήση επεξεργαστή, παράλληλη επεξεργασία - εισαγωγή καθυστέρησης περίπτωση Γ

Παρατηρείται ότι η χρήση του επεξεργαστή από το Resolution έχει διακυμάνσεις ενώ στις προηγούμενες περιπτώσεις ήταν σταθερή. Αυτό δικαιολογείται από τον τρόπο με τον οποίο έχει γίνει η εισαγωγή της καθυστέρησης στο συγκεκριμένο στάδιο.

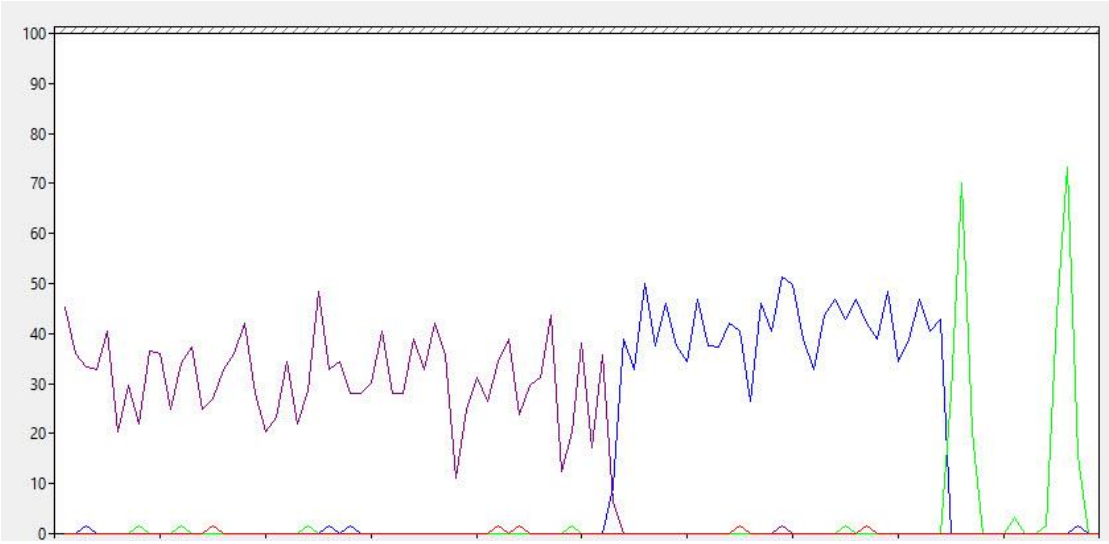
Πίνακας 7 Χρήση επεξεργαστή, χρόνος επεξεργαστή, παράλληλη επεξεργασία - εισαγωγή καθυστέρησης περίπτωση Γ

	Average CPU Usage (%)	**CPU Time (Seconds)**
Technical	8	1.49
Business	52	33.54
Price	61	38.17
Resolution	48	11.24
Total	-	38.47

Παρατηρείται αύξηση του χρόνου εκτέλεσης σε όλα τα στάδια εκτός του πρώτου. Στο Business η αύξηση είναι μικρότερη από ότι στο Price και στο Resolution.

7.2.4 Μετρήσεις με εισαγωγή καθυστέρησης – Περίπτωση Δ

Σειριακή Αρχιτεκτονική



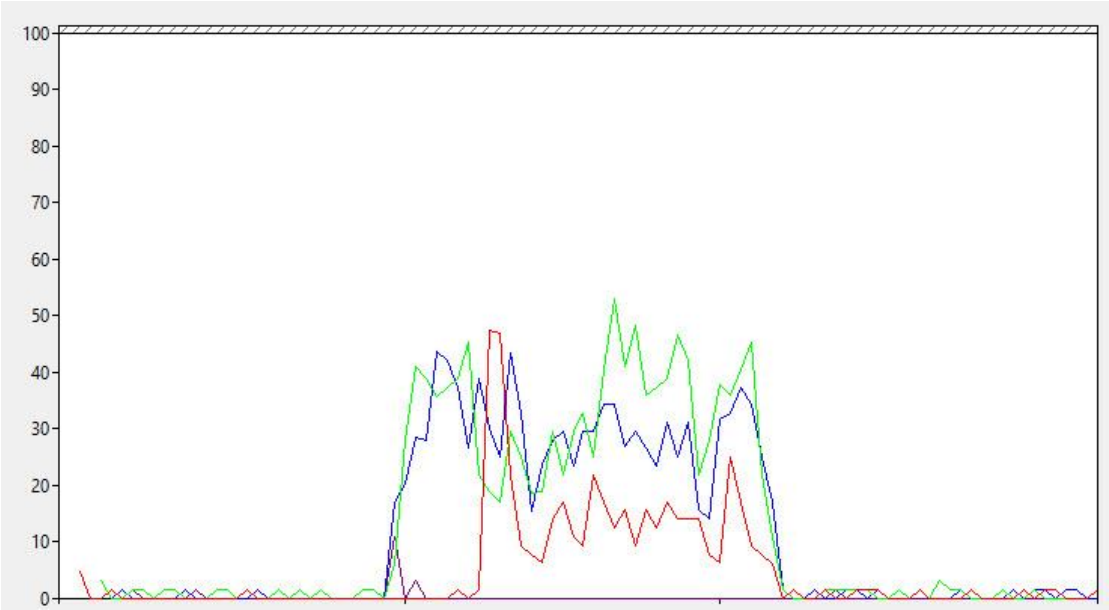
Εικόνα 18 Χρήση επεξεργαστή, σειριακή επεξεργασία - εισαγωγή καθυστέρησης περίπτωση Δ

Πίνακας 8 Χρήση επεξεργαστή, χρόνος επεξεργαστή, σειριακή επεξεργασία - εισαγωγή καθυστέρησης περίπτωση Δ

	Average CPU Usage (%)	**CPU Time (Seconds)**
Technical	0.3	19.42
Business	46	56.36
Price	47	38.74
Resolution	72	6.33
Total	-	120.85

Παρατηρείται αύξηση σε όλα τα στάδια εκτός του Price που δεν έχει γίνει εισαγωγή καθυστέρησης. Επίσης έχει αυξηθεί ο συνολικός χρόνος.

Παράλληλη Αρχιτεκτονική



Εικόνα 19 Χρήση επεξεργαστή, παράλληλη επεξεργασία - εισαγωγή καθυστέρησης περίπτωση Δ

Παρατηρούνται αυξομειώσεις στη χρήση του επεξεργαστή από όλα τα στάδια.

Πίνακας 9 Χρήση επεξεργαστή, χρόνος επεξεργαστή, παράλληλη επεξεργασία - εισαγωγή καθυστέρησης περίπτωση Δ

	Average CPU Usage (%)	**CPU Time (Seconds)**
Technical	16	5.16
Business	38	38.45
Price	44	38.07
Resolution	29	28.85
Total	-	38.47

Παρατηρείται αύξηση του χρόνου εκτέλεσης σε όλα τα στάδια καθώς και αύξηση του συνολικού χρόνου.

7.3 Μετρήσεις με παραμετροποίηση των απαιτήσεων, αύξηση των αρχείων/μηνυμάτων

Χωρίς παραμετροποίηση των απαιτήσεων τα αρχεία που παράγονται σε κάθε στάδιο φαίνονται στον επόμενο πίνακα.

Πίνακας 10 Πλήθος αρχείων χωρίς παραμετροποίηση

Technical	120
Business	11880
Price	10551
Resolution	8

Λήφθηκαν μετρήσεις για τέσσερεις διαφορετικές περιπτώσεις.

Α) Στην πρώτη περίπτωση παραμετροποιήθηκαν οι τεχνικές απαιτήσεις ώστε να παραχθούν περισσότερα αρχεία από το Technical. Τα αρχεία που παράγονται σε κάθε στάδιο φαίνονται στον επόμενο πίνακα.

Πίνακας 11 Πλήθος αρχείων - παραμετροποίηση περίπτωση Α

Technical	360
Business	21384
Price	18448
Resolution	0

Β) Στη δεύτερη περίπτωση αυξήθηκε το μέγιστο κόστος ώστε να παραχθούν περισσότερα αρχεία από το Price. Τα αρχεία που παράγονται σε κάθε στάδιο φαίνονται στον επόμενο πίνακα.

Πίνακας 12 Πλήθος αρχείων - παραμετροποίηση περίπτωση Β

Technical	120
Business	11880
Price	11880
Resolution	24

Γ) Στην τρίτη περίπτωση παραμετροποιήθηκαν οι επιχειρηματικές απαιτήσεις ώστε να παραχθούν περισσότερα αρχεία από το Resolution. Τα αρχεία που παράγονται σε κάθε στάδιο φαίνονται στον επόμενο πίνακα.

Πίνακας 13 Πλήθος αρχείων - παραμετροποίηση περίπτωση Γ

Technical	120
Business	11880
Price	10551
Resolution	2820

Δ) Στην τέταρτη περίπτωση εφαρμόστηκαν όλες οι παραμετροποιήσεις των προηγούμενων περιπτώσεων μαζί. Τα αρχεία που παράγονται σε κάθε στάδιο φαίνονται στον επόμενο πίνακα.

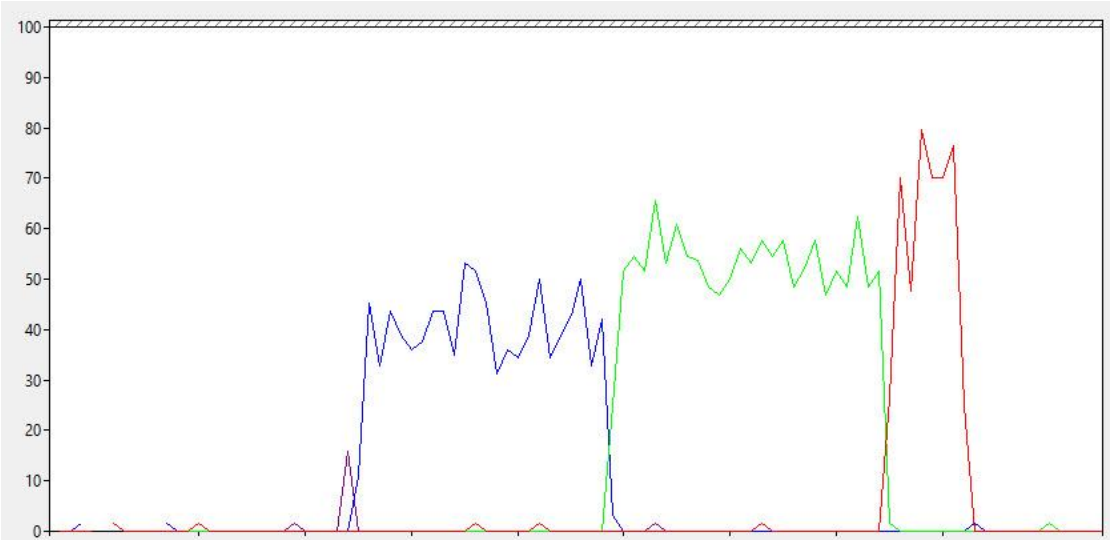
Πίνακας 14 Πλήθος αρχείων - παραμετροποίηση περίπτωση Δ

Technical	360
Business	21384
Price	21342
Resolution	4868

Ακολουθούν οι αντίστοιχες μετρήσεις.

7.3.1 Μετρήσεις με παραμετροποίηση απαιτήσεων – Περίπτωση Α

Σειριακή Αρχιτεκτονική



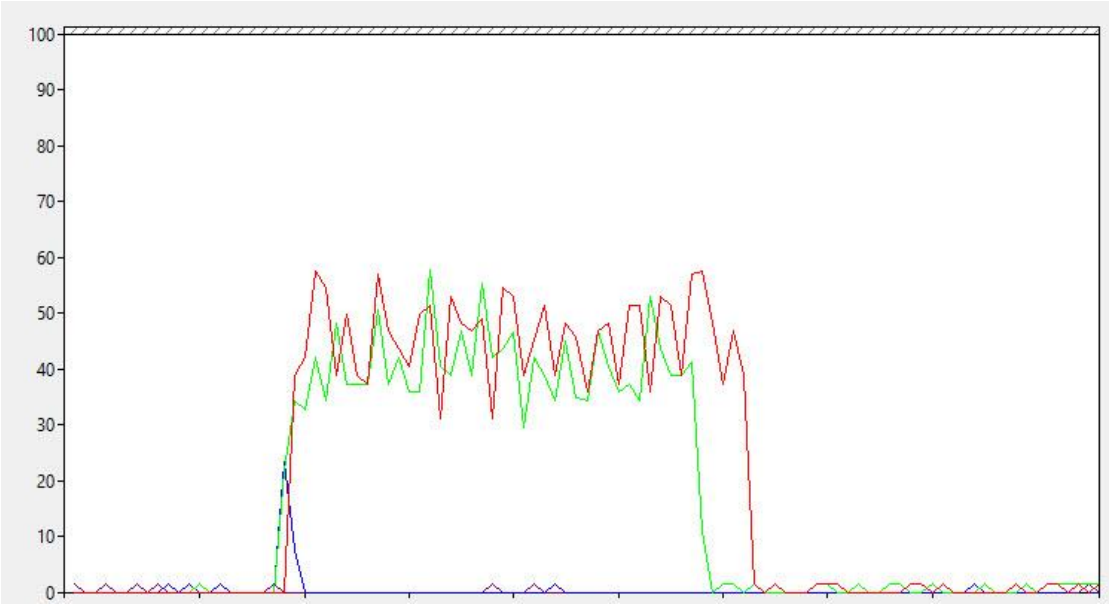
Εικόνα 20 Χρήση επεξεργαστή, σειριακή επεξεργασία - παραμετροποίηση απαιτήσεων περίπτωση Α

Πίνακας 15 Χρήση επεξεργαστή, χρόνος επεξεργαστή, σειριακή επεξεργασία - παραμετροποίηση απαιτήσεων περίπτωση Α

	Average CPU Usage (%)	**CPU Time (Seconds)**
Technical	15	1.54
Business	49	34.41
Price	58	36.84
Resolution	78	4.23
Total	-	74.84

Παρατηρείται μικρή αύξηση του συνολικού χρόνου εκτέλεσης.

Παράλληλη Αρχιτεκτονική



Εικόνα 21 Χρήση επεξεργαστή, παράλληλη επεξεργασία - παραμετροποίηση απαιτήσεων περίπτωση Α

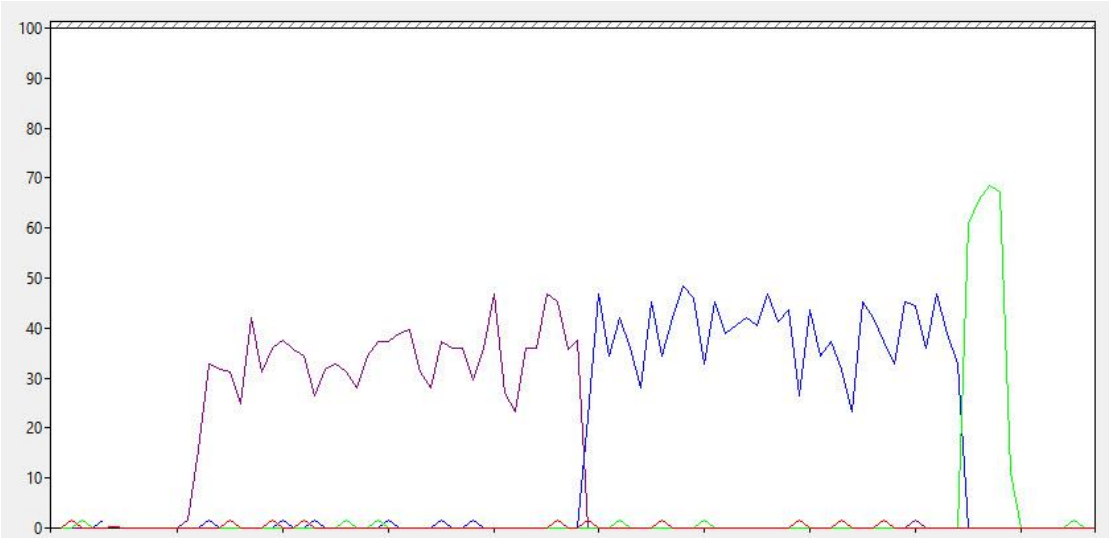
Πίνακας 16 Χρήση επεξεργαστή, χρόνος επεξεργαστή, παράλληλη επεξεργασία - παραμετροποίηση απαιτήσεων περίπτωση Α

	Average CPU Usage (%)	**CPU Time (Seconds)**
Technical	16	2.29
Business	48	44.45
Price	56	43.77
Resolution	19	45.86
Total	-	46.58

Παρατηρείται αύξηση στο συνολικό χρόνο εκτέλεσης καθώς και σε όλα τα στάδια.

7.3.2 Μετρήσεις με παραμετροποίηση απαιτήσεων – Περίπτωση Β

Σειριακή Αρχιτεκτονική



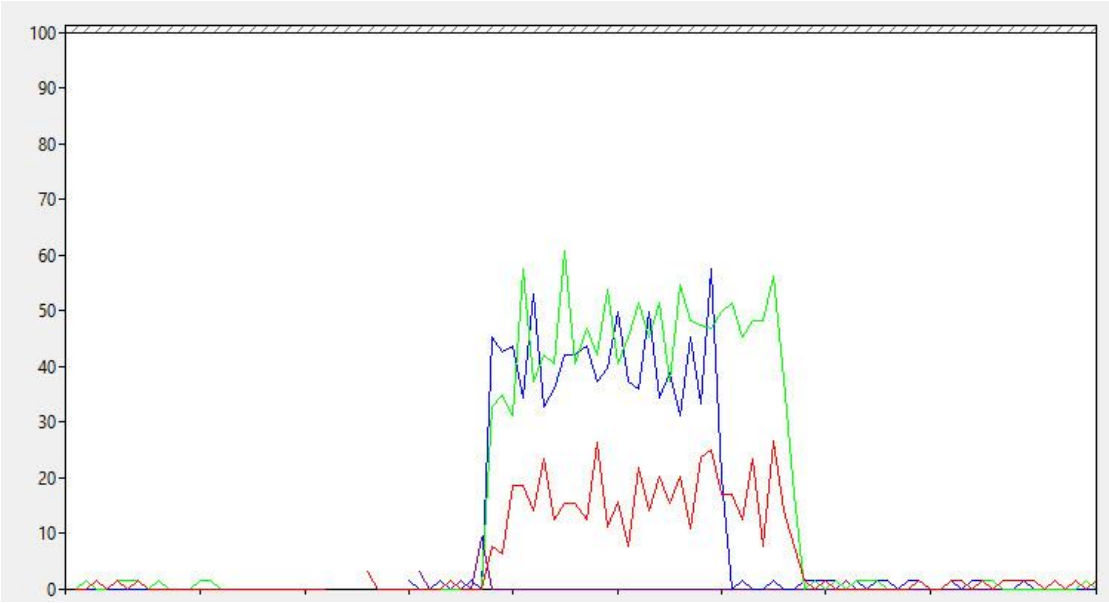
Εικόνα 22 Χρήση επεξεργαστή, σειριακή επεξεργασία - παραμετροποίηση απαιτήσεων περίπτωση Β

Πίνακας 17 Χρήση επεξεργαστή, χρόνος επεξεργαστή, σειριακή επεξεργασία - παραμετροποίηση απαιτήσεων περίπτωση Β

	Average CPU Usage (%)	**CPU Time (Seconds)**
Technical	0.3	1.42
Business	41	35.31
Price	46	37.84
Resolution	70	4.83
Total	-	78.13

Παρατηρείται μικρή αύξηση στο χρόνο εκτέλεσης στα στάδια Business, Price, Resolution και στο συνολικό χρόνο.

Παράλληλη Αρχιτεκτονική



Εικόνα 23 Χρήση επεξεργαστή, παράλληλη επεξεργασία - παραμετροποίηση απαιτήσεων περίπτωση Β

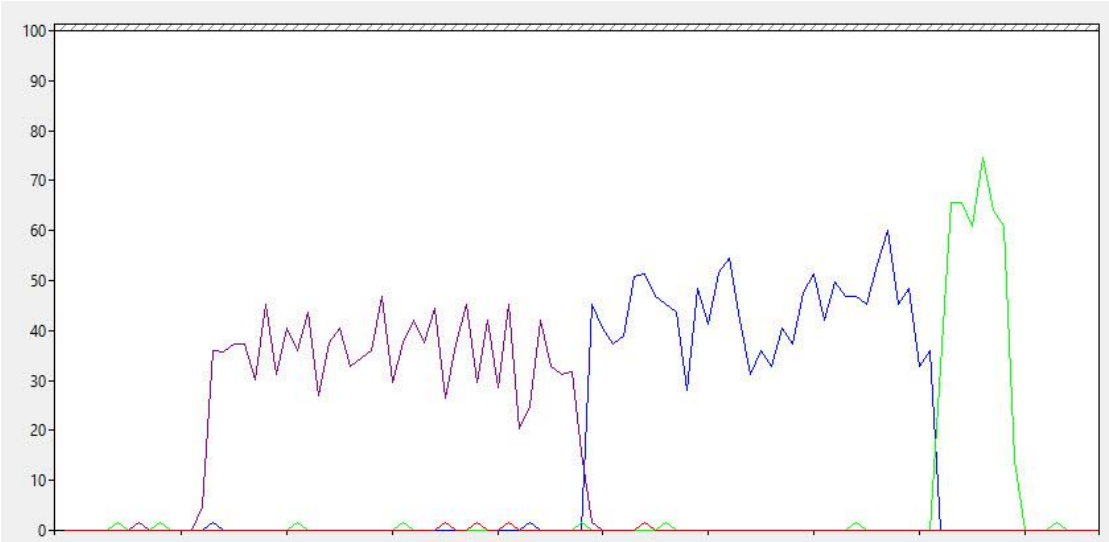
Πίνακας 18 Χρήση επεξεργαστή, χρόνος επεξεργαστή, παράλληλη επεξεργασία - παραμετροποίηση απαιτήσεων περίπτωση Β

	Average CPU Usage (%)	**CPU Time (Seconds)**
Technical	9	1.51
Business	48	25.55
Price	56	29.47
Resolution	24	29.92
Total	-	30.23

Παρατηρείται μικρή αύξηση στα στάδια Business, Price και Resolution.

7.3.3 Μετρήσεις με παραμετροποίηση απαιτήσεων – Περίπτωση Γ

Σειριακή Αρχιτεκτονική



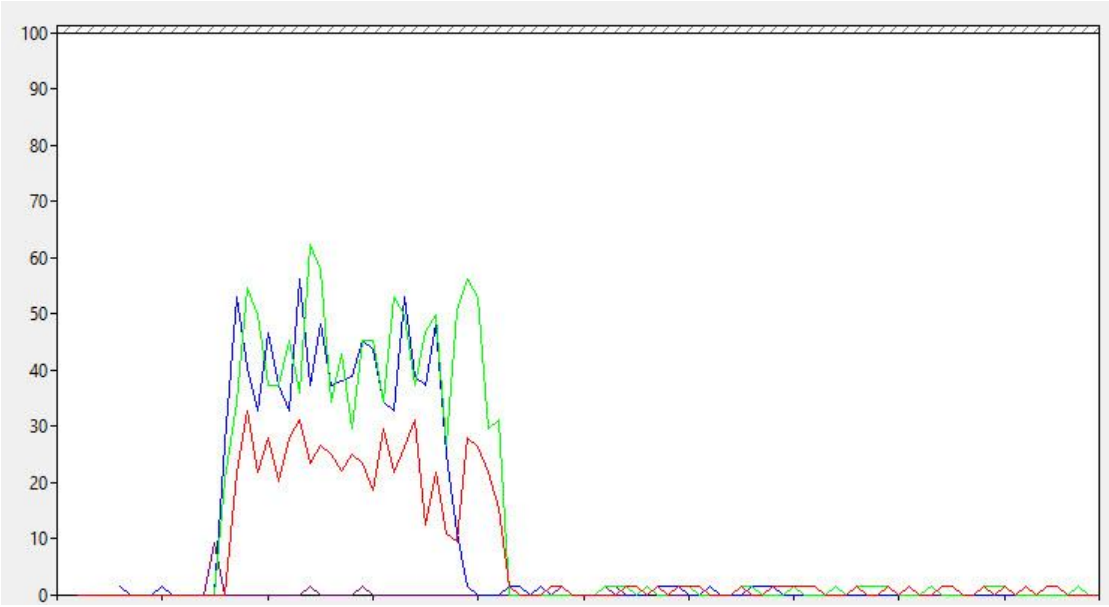
Εικόνα 24 Χρήση επεξεργαστή, σειριακή επεξεργασία - παραμετροποίηση απαιτήσεων περίπτωση Γ

Πίνακας 19 Χρήση επεξεργαστή, χρόνος επεξεργαστή, σειριακή επεξεργασία - παραμετροποίηση απαιτήσεων περίπτωση Γ

	Average CPU Usage (%)	**CPU Time (Seconds)**
Technical	0.4	1.52
Business	42	40.31
Price	51	42.74
Resolution	79	3.23
Total	-	89.04

Παρατηρείται μικρή αύξηση στο χρόνο εκτέλεσης στα στάδια Business, Price, Resolution και στο συνολικό χρόνο.

Παράλληλη Αρχιτεκτονική



Εικόνα 25 Χρήση επεξεργαστή, παράλληλη επεξεργασία - παραμετροποίηση απαιτήσεων περίπτωση Γ

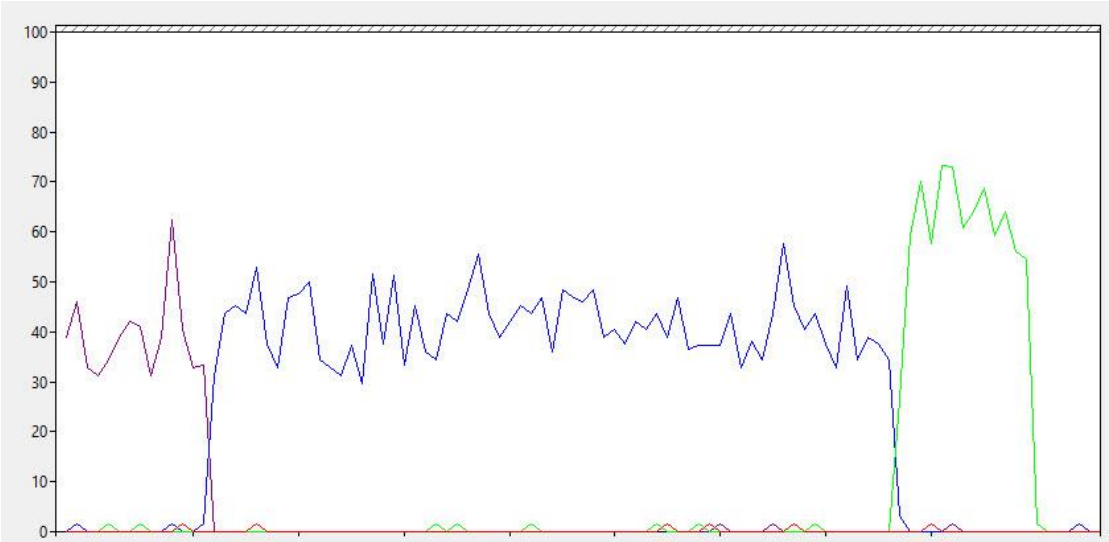
Πίνακας 20 Χρήση επεξεργαστή, χρόνος επεξεργαστή, παράλληλη επεξεργασία - παραμετροποίηση απαιτήσεων περίπτωση Γ

	Average CPU Usage (%)	**CPU Time (Seconds)**
Technical	8	1.47
Business	48	25.45
Price	56	28.57
Resolution	30	27.64
Total	-	28.82

Παρατηρείται ελάχιστη αύξηση στο συνολικό χρόνο εκτέλεσης.

7.3.4 Μετρήσεις με παραμετροποίηση απαιτήσεων – Περίπτωση Δ

Σειριακή Αρχιτεκτονική



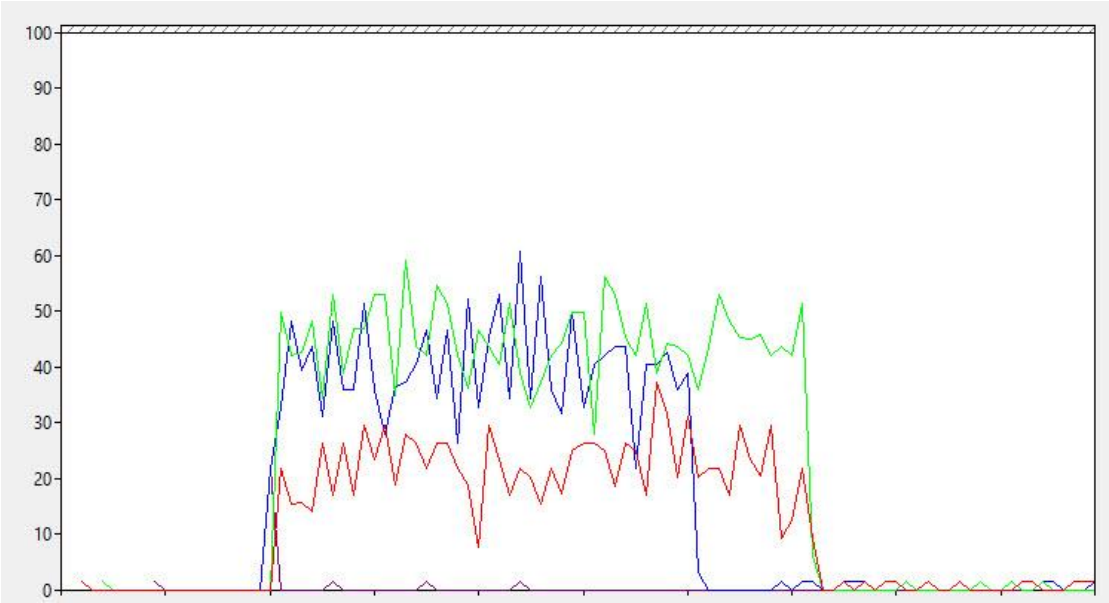
Εικόνα 26 Χρήση επεξεργαστή, σειριακή επεξεργασία - παραμετροποίηση απαιτήσεων περίπτωση Δ

Πίνακας 21 Χρήση επεξεργαστή, χρόνος επεξεργαστή, σειριακή επεξεργασία - παραμετροποίηση απαιτήσεων περίπτωση Δ

	Average CPU Usage (%)	**CPU Time (Seconds)**
Technical	0.4	8.92
Business	49	76.31
Price	51	79.94
Resolution	74	19.78
Total	-	184.95

Παρατηρείται σημαντική αύξηση στο χρόνο εκτέλεσης σε όλα τα στάδια καθώς και στο συνολικό χρόνο εκτέλεσης.

Παράλληλη Αρχιτεκτονική



Εικόνα 27 Χρήση επεξεργαστή, παράλληλη επεξεργασία - παραμετροποίηση απαιτήσεων περίπτωση Δ

Πίνακας 22 Χρήση επεξεργαστή, χρόνος επεξεργαστή, παράλληλη επεξεργασία - παραμετροποίηση απαιτήσεων περίπτωση Δ

	Average CPU Usage (%)	**CPU Time (Seconds)**
Technical	16	2.12
Business	48	41.15
Price	56	53.89
Resolution	19	53.73
Total	-	54.04

Παρατηρείται αύξηση στο χρόνο εκτέλεσης σε όλα τα στάδια και στο συνολικό χρόνο.

7.4 Συμπεράσματα

Με βάση τις μετρήσεις διαπιστώνεται ότι η παράλληλη αρχιτεκτονική είναι γενικά πολύ ταχύτερη από τη σειριακή. Σε όλες τις περιπτώσεις ο χρόνος εκτέλεσης είναι αρκετά μικρότερος.

Η χρήση του επεξεργαστή είναι περίπου ίδια εκτός από το Resolution που στην περίπτωση της παράλληλης επεξεργασίας η χρήση του επεξεργαστή είναι αισθητά μικρότερη.

Στην περίπτωση της εισαγωγής καθυστέρησης, στη σειριακή αρχιτεκτονική προστίθεται η επιπλέον καθυστέρηση στο αντίστοιχο στάδιο και στο συνολικό χρόνο εκτέλεσης. Αντίθετα

στην παράλληλη αρχιτεκτονική επειδή τα στάδια λειτουργούν ταυτόχρονα παρατηρείται αύξηση σε όλα τα στάδια. Αυτή η αύξηση όμως είναι πολύ μικρότερη. Αυτό συμβαίνει γιατί παρόλο που καθυστερεί ένα στάδιο περισσότερα, παράλληλα τα υπόλοιπα λειτουργούν και απλά περιμένουν λίγο περισσότερο για να πάρουν μήνυμα. Αυτό έχει σαν αποτέλεσμα μικρή αύξηση του συνολικού χρόνου εκτέλεσης. Αυτό φαίνεται περισσότερο στην τελευταία περίπτωση που εισάγονται καθυστερήσεις σε όλα τα στάδια. Ο συνολικός χρόνος εκτέλεσης στη σειριακή επεξεργασία αυξάνεται δραματικά ενώ στην παράλληλη επεξεργασία δεν παρατηρείται μεγάλη αύξηση.

Στην περίπτωση της παραμετροποίησης των απαιτήσεων στη σειριακή αρχιτεκτονική παρατηρείται μικρότερη αύξηση του συνολικού χρόνου εκτέλεσης από την παράλληλη αρχιτεκτονική. Αυτό συμβαίνει γιατί τα αρχεία που διαβάζονται από τη βάση και αυτά που γράφονται σε αυτή καθώς και τα μηνύματα που στέλνονται είναι πολύ περισσότερα και αυτό επηρεάζει όλα τα στάδια στην παράλληλη επεξεργασία. Αντίθετα στη σειριακή επεξεργασία επηρεάζεται μόνο το αντίστοιχο στάδιο. Αυτό έχει σαν αποτέλεσμα όμως μόνο την αύξηση στην κάθε περίπτωση ξεχωριστά αφού συνολικά και πάλι η παράλληλη επεξεργασία είναι πολύ ταχύτερη από τη σειριακή. Ειδικά στην τελευταία παραμετροποίηση όπου το πλήθος των αρχείων και των μηνυμάτων είναι αυξημένο σε όλα τα στάδια, η σειριακή αρχιτεκτονική παρουσιάζει αύξηση 147% ενώ η παράλληλη αύξηση 91%.

Συμπεραίνεται ότι η παράλληλη αρχιτεκτονική με την ενιαία ουρά μηνυμάτων είναι καλύτερη σε χρόνο εκτέλεσης και σε χρήση του επεξεργαστή, σε όλες τις περιπτώσεις από τη σειριακή αρχιτεκτονική.

8

Βιβλιογραφία

[1] Min Chen, Shiwen Mao, Yunhao Liu. Big Data: A Survey. Springer Science and Business Media New York 2014

[2] Wu, D., Liu. X., Hebert, S., Gentzsch, W., Terpenny, J. (2015). Performance Evaluation of Cloud-Based High Performance Computing for Finite Element Analysis. Proceedings of the ASME 2015 International Design Engineering Technical Conference & Computers and Information in Engineering Conference (IDETC/CIE2015), Boston, Massachusetts, U.S.

[3] Wu, D.; Rosen, D.W.; Wang, L.; Schaefer, D. (2015). "Cloud-Based Design and Manufacturing: A New Paradigm in Digital Manufacturing and Design Innovation". Computer-Aided Design

[4] Peter Mell, Timothy Grance “The NIST Definition of Cloud Computing”, Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899-8930, September

[5] M. Haghighat, S. Zonouz, & M. Abdel-Mottaleb (2015). CloudID: Trustworthy Cloud-based and Cross-Enterprise Biometric Identification. Expert Systems with Applications, 42(21), 7905–7916

[6] Gruman, Galen (2008-04-07). "What cloud computing really means". InfoWorld. Retrieved 2009-06-02

[7] Chhibber, A (2013). "SECURITY ANALYSIS OF CLOUD COMPUTING". International Journal of Advanced Research in Engineering and Applied Sciences 2 (3): 2278–6252. Retrieved 27 February 2015

[8] O'Hara, J. "Toward a commodity enterprise middleware". ACM Queue 2007

[9] "OASIS AMQP version 1.0, section 1.1". OASIS AMQP Technical Committee. Retrieved 18 June 2012

Παράρτημα – Παράθεση κώδικα, αρχείων

Παρατίθενται τα αρχεία με τον κώδικα του marketplace.

Σειριακή επεξεργασία

TechnicalResolution.py

```
import pika
import pymongo
from pymongo import MongoClient
import json

#Connect to pipeline db
client = MongoClient()
db = client.pipeline

#Get collections needed
req = db.req
technical = db.technical
treso = db.technicalreso

technicalist=db.technical.find_one()
usereq=db.req.find_one()

#Connect to rabbitmq
connection =
pika.BlockingConnection(pika.ConnectionParameters(host='localhost'))
channel = connection.channel()

#Declare Business queue
result = channel.queue_declare(queue='Business')
queue_name = result.method.queue
channel.exchange_declare(exchange='MainX', type='topic')

#Technical result
counter = 0 #counter for mongo id
def callback(ch, method, properties, body):
      for i in range(0, len(technicalist["database"])):
            #time.sleep(0.5)
            if technicalist["database"][i]["Type"] ==
usereq["UserRequirements"]["Service"]["Database"]["Type"] and
technicalist["database"][i]["OP"] ==
usereq["UserRequirements"]["Service"]["Database"]["OP"]:
                  u=usereq
```

```
                  u["Technical"]={"database" :
technicalist["database"][i]}
                  for j in range(0,
len(technicalist["OperationSystem"])):
                        if technicalist["OperationSystem"][j]["Type"]
== usereq["UserRequirements"]["Service"]["OperationSystem"]["Type"]
and technicalist["OperationSystem"][j]["OP"] ==
usereq["UserRequirements"]["Service"]["OperationSystem"]["OP"]:

      u["Technical"]['OperationSystem']=technicalist["OperationSystem
"][j]
                              for k in range(0,
len(technicalist["CMS"])):
                                    if technicalist["CMS"][k]["OP"]
== usereq["UserRequirements"]["Service"]["CMS"]["OP"]:

      u["Technical"]['CMS']=technicalist["CMS"][k]
                                          for l in range(0,
len(technicalist["Network"])):
                                                if
technicalist["Network"][l]["Type"] ==
usereq["UserRequirements"]["Service"]["Network"]["Type"]:

      u["Technical"]['Network']=technicalist["Network"][l]
                                                      global counter
                                                      counter += 1

      u["_id"]=counter

      result=treso.insert_one(u)

      channel.basic_publish(exchange='MainX', routing_key='skata',
body='Business')
                                                      print " [x]
Sent to technical reso"


#Waiting for message from user
while "true":
      result = channel.queue_declare(queue='Technical')
      if not result.method.message_count == 0:
            method, properties, body =
channel.basic_get(queue='Technical', no_ack=True)
            callback(channel, method, properties, body)
            result = channel.queue_declare(queue='Business')
            channel.basic_publish(exchange='',
routing_key='Business', body='')
            print " [x] Sent tech to business"
```

BusinessResolution.py

```
import pika
import pymongo
from pymongo import MongoClient
import json

#Connect to rabbitmq
connection =
pika.BlockingConnection(pika.ConnectionParameters(host='localhost'))
```

```
channel = connection.channel()

#Connect to pipeline db
client = MongoClient()
db = client.pipeline

#Get collections needed
business = db.business
breso = db.businessreso
treso = db.technicalreso

businesslist=db.business.find_one()

channel.exchange_declare(exchange='MainX', type='topic')
counter = 0 #counter for mongo id
def callback(ch, method, properties, body):
      #Business result
    for tech in treso.find():
            #time.sleep(0.1)
            treso.delete_many({'_id':tech['_id']})
            for i in range(0, len(businesslist['PDatabase'])):
                  if businesslist['PDatabase'][i]['id'] ==
tech['Technical']['database']['id']:
                        t=tech

      t['Business']={'PDatabase':businesslist['PDatabase'][i]}
                        for j in range(0,
len(businesslist['POperationSystem'])):
                              if
businesslist['POperationSystem'][j]['id'] ==
tech['Technical']['OperationSystem']['id']:

      t['Business']['POperationSystem']=businesslist['POperationSyste
m'][j]
                                    for k in range(0,
len(businesslist['PCMS'])):
                                          if
businesslist['PCMS'][k]['id'] == tech['Technical']['CMS']['id']:

      t['Business']['PCMS']=businesslist['PCMS'][k]
                                                for l in range(0,
len(businesslist['PNetwork'])):
                                                      if
businesslist['PNetwork'][l]['id'] ==
tech['Technical']['Network']['id']:

      t['Business']['PNetwork']=businesslist['PNetwork'][l]
                                                            global
counter
                                                            counter
+= 1

      t['_id']=counter

      result=breso.insert_one(t)

      channel.basic_publish(exchange='MainX', routing_key='skata',
body='Business')
                                                            print
"[x] Business sent to businessreso"
```

```
#Waiting for message from tech unit
while "true":
      result = channel.queue_declare(queue='Business')
      if not result.method.message_count == 0:
            method, properties, body =
channel.basic_get(queue='Business', no_ack=True)
            callback(channel, method, properties, body)
            result = channel.queue_declare(queue='Pirce')
            channel.basic_publish(exchange='', routing_key='Price',
body='')
            print " [x] Sent tech to price"
```

PriceAggregator.py

```
import pika
import pymongo
from pymongo import MongoClient
import json
import time

#Connect to rabbitmq
connection =
pika.BlockingConnection(pika.ConnectionParameters(host='localhost'))
channel = connection.channel()

#Connect to pipeline db
client = MongoClient()
db = client.pipeline

#Get collections needed
req = db.req
breso = db.businessreso
preso = db.pricereso

#Get user requirements
usereq=db.req.find_one()
channel.exchange_declare(exchange='MainX', type='topic')
counter = 0 #counter for mongo id
def callback(ch, method, properties, body):
      #Price result
      for bus in breso.find():
            breso.delete_many({'_id':bus['_id']})
            tprice = int(bus['Business']['PDatabase']['Price']) +
int(bus['Business']['POperationSystem']['Price']) +
int(bus['Business']['PCMS']['Price']) +
int(bus['Business']['PNetwork']['Price'])
            if tprice <=
int(usereq['UserRequirements']['Optimization']['MAXPrice']):
                  b=bus
                  b['Total']={'TotalPrice':tprice}
                  global counter
                  counter += 1
                  b['_id']=counter
                  result=preso.insert_one(b)
                  channel.basic_publish(exchange='MainX',
routing_key='skata', body='Business')
                  print "[x] Price sent to resolution"
```

```
#Waiting for message from business unit
while "true":
      result = channel.queue_declare(queue='Price')
      if not result.method.message_count == 0:
            method, properties, body =
channel.basic_get(queue='Price', no_ack=True)
            callback(channel, method, properties, body)
            result = channel.queue_declare(queue='Resolution')
            channel.basic_publish(exchange='',
routing_key='Resolution', body='Diavase json zwo')
            print " [x] Sent tech to resolution"
```

FinalResolution.py

```
import pika
import pymongo
from pymongo import MongoClient
import json

#Connect to rabbitmq
connection =
pika.BlockingConnection(pika.ConnectionParameters(host='localhost'))
channel = connection.channel()

#Connect to pipeline db
client = MongoClient()
db = client.pipeline

#Get collections needed
req = db.req
preso = db.pricereso
reso = db.resolution

#Get user requirements
usereq=db.req.find_one()

counter = 0 #counter for mongo id
def callback(ch, method, properties, body):
      #Resolution!
      for pr in preso.find():
            #time.sleep(0.1)
            preso.delete_many({'_id':pr['_id']})
            tavailability =
(int(pr['Business']['PDatabase']['Availability']) +
int(pr['Business']['POperationSystem']['Availability']) +
int(pr['Business']['PCMS']['Availability']) +
int(pr['Business']['PNetwork']['Availability']))/4
            if tavailability >=
float(usereq['UserRequirements']['Optimization']['MINAvailability']):
                  tsecurity =
(int(pr['Business']['PDatabase']['Security']) +
int(pr['Business']['POperationSystem']['Security']) +
int(pr['Business']['PCMS']['Security']) +
int(pr['Business']['PNetwork']['Security']))/4
                  if tsecurity >=
float(usereq['UserRequirements']['Optimization']['MINSecurity']):
                        treputation =
(int(pr['Business']['PDatabase']['Reputation']) +
int(pr['Business']['POperationSystem']['Reputation']) +
```

```
int(pr['Business']['PCMS']['Reputation']) +
int(pr['Business']['PNetwork']['Reputation']))/4
                        if treputation >=
float(usereq['UserRequirements']['Optimization']['MINReputation']):
                                treliability =
(int(pr['Business']['PDatabase']['Reliability']) +
int(pr['Business']['POperationSystem']['Reliability']) +
int(pr['Business']['PCMS']['Reliability']) +
int(pr['Business']['PNetwork']['Reliability']))/4
                                if treliability >=
float(usereq['UserRequirements']['Optimization']['MINReliability']):
                                        tconfidentiality=0
                                        if
pr['Business']['PDatabase']['Confidentiality'] == 'YES':
                                                tconfidentiality+=1
                                        if
pr['Business']['POperationSystem']['Confidentiality'] == 'YES':
                                                tconfidentiality+=1
                                        if
pr['Business']['PCMS']['Confidentiality'] == 'YES':
                                                tconfidentiality+=1
                                        if
pr['Business']['PNetwork']['Confidentiality'] == 'YES':
                                                tconfidentiality+=1

        tconfidentiality=tconfidentiality/4
                                        if tconfidentiality >= 0.5:
                                                stconfidentiality='YES'
                                        else:
                                                stconfidentiality='NO'
                                        if stconfidentiality ==
usereq['UserRequirements']['Optimization']['Confidentiality']:
                                                ttotal = 0.3*tavailability
+ 0.6*tsecurity + 0.5*treputation + 0.7*treliability +
0.5*tconfidentiality - 0.005*pr['Total']['TotalPrice']
                                                p=pr

        p['Total']['TotalAvailability']=int(tavailability)

        p['Total']['TotalSecurity']=int(tsecurity)

        p['Total']['TotalReputation']=int(treputation)

        p['Total']['TotalReliability']=int(treliability)

        p['Total']['TotalConfidentiality']=stconfidentiality

        p['Total']['TotalTotal']=ttotal

                                                global counter
                                                counter += 1
                                                p['_id']=counter
                                                result=reso.insert_one(p)
                                                print "[x] Resolution sent
results"


#Waiting for message from price unit
while "true":
        result = channel.queue_declare(queue='Resolution')
        if not result.method.message_count == 0:
```

```
            method, properties, body =
channel.basic_get(queue='Resolution', no_ack=True)
            callback(channel, method, properties, body)
            print " [x] Finished!"
```

Σειριακή επεξεργασία

QTechnicalResolution.py

```
import pika
import psutil
import pymongo
from pymongo import MongoClient
import json


#Connect to rabbitmq
connection =
pika.BlockingConnection(pika.ConnectionParameters(host='localhost'))
channel = connection.channel()


#Connect to pipeline db
client = MongoClient()
db = client.pipeline

#Get collections needed
req = db.req
technical = db.technical
treso = db.technicalreso

technicalist=db.technical.find_one()
usereq=db.req.find_one()

#Technical result
def callback(ch, method, properties, body):
      counter = 0 #counter for mongo id
      for i in range(0, len(technicalist["database"])):
            #time.sleep(0.5)
            if technicalist["database"][i]["Type"] ==
usereq["UserRequirements"]["Service"]["Database"]["Type"] and
technicalist["database"][i]["OP"] ==
usereq["UserRequirements"]["Service"]["Database"]["OP"]:
                  u=usereq
                  u["Technical"]={"database" :
technicalist["database"][i]}
                  for j in range(0,
len(technicalist["OperationSystem"])):
                        if technicalist["OperationSystem"][j]["Type"]
== usereq["UserRequirements"]["Service"]["OperationSystem"]["Type"]
and technicalist["OperationSystem"][j]["OP"] ==
usereq["UserRequirements"]["Service"]["OperationSystem"]["OP"]:

      u["Technical"]['OperationSystem']=technicalist["OperationSystem
"][j]
                              for k in range(0,
len(technicalist["CMS"])):
                                    if technicalist["CMS"][k]["OP"]
== usereq["UserRequirements"]["Service"]["CMS"]["OP"]:
```

```
      u["Technical"]['CMS']=technicalist["CMS"][k]
                                          for l in range(0,
len(technicalist["Network"])):
                                                if
technicalist["Network"][l]["Type"] ==
usereq["UserRequirements"]["Service"]["Network"]["Type"]:

      u["Technical"]['Network']=technicalist["Network"][l]
                                                      counter += 1

      u["_id"]=counter

      result=treso.insert_one(u)
                                                    print counter

      channel.basic_publish(exchange='MainX', routing_key='tech',
body='Technical')



while "true":
      result = channel.queue_declare(queue='TechnicalQ')
      if not result.method.message_count == 0:
            method, properties, body =
channel.basic_get(queue='TechnicalQ', no_ack=True)
            callback(channel, method, properties, body)
```

QBusinessResolution.py

```
import pika
import pymongo
from pymongo import MongoClient
import json

#Connect to rabbitmq
connection =
pika.BlockingConnection(pika.ConnectionParameters(host='localhost'))
channel = connection.channel()


channel.exchange_declare(exchange='MainX', type='topic')
result = channel.queue_declare(queue="BusinessQ")#, exclusive=True,
arguments=args)
queue_name = result.method.queue



channel.queue_bind(exchange='MainX', queue=queue_name,
routing_key='tech')

client = MongoClient()
db = client.pipeline

#Get collections needed
business = db.business
breso = db.businessreso
treso = db.technicalreso
```

```
businesslist=business.find_one()

print ' [*] Business waiting for messages. To exit press CTRL+C'
counter = 0 #counter for mongo id
def callback(ch, method, properties, body):
      if treso.count()>0:
            for tech in treso.find():
                  #time.sleep(0.1)
                  treso.delete_many({'_id':tech['_id']})
                  for i in range(0, len(businesslist['PDatabase'])):
                        if businesslist['PDatabase'][i]['id'] ==
tech['Technical']['database']['id']:
                              t=tech

      t['Business']={'PDatabase':businesslist['PDatabase'][i]}
                              for j in range(0,
len(businesslist['POperationSystem'])):
                                    if
businesslist['POperationSystem'][j]['id'] ==
tech['Technical']['OperationSystem']['id']:

      t['Business']['POperationSystem']=businesslist['POperationSyste
m'][j]
                                          for k in range(0,
len(businesslist['PCMS'])):
                                                if
businesslist['PCMS'][k]['id'] == tech['Technical']['CMS']['id']:

      t['Business']['PCMS']=businesslist['PCMS'][k]
                                                      for l in
range(0, len(businesslist['PNetwork'])):
                                                            if
businesslist['PNetwork'][l]['id'] ==
tech['Technical']['Network']['id']:

      t['Business']['PNetwork']=businesslist['PNetwork'][l]

      global counter

      global flag

      counter += 1

      t['_id']=counter

      result=breso.insert_one(t)

      print counter

      channel.basic_publish(exchange='MainX', routing_key='price',
body='Business')



while "true":
    result = channel.queue_declare(queue="BusinessQ")#,
exclusive=True, arguments=args)
    if not result.method.message_count == 0:
        method, properties, body =
channel.basic_get(queue='BusinessQ', no_ack=True)
```

```
        callback(channel, method, properties, body)
```

QPriceAggregator.py

```
import pika
import pymongo
from pymongo import MongoClient
import json

#Connect to rabbitmq
connection =
pika.BlockingConnection(pika.ConnectionParameters(host='localhost'))
channel = connection.channel()

channel.exchange_declare(exchange='MainX', type='topic')
result = channel.queue_declare(queue="PriceQ")#, exclusive=True,
arguments=args)
queue_name = result.method.queue


channel.queue_bind(exchange='MainX', queue=queue_name,
routing_key='price')

#Connect to pipeline db
client = MongoClient()
db = client.pipeline

#Get collections needed
req = db.req
breso = db.businessreso
preso = db.pricereso

#Get user requirements
usereq=db.req.find_one()

print ' [*] Price waiting for messages. To exit press CTRL+C'

counter = 0 #counter for mongo id
def callback(ch, method, properties, body):
      if breso.count()>0:
            for bus in breso.find():
                  breso.delete_many({'_id':bus['_id']})
                  tprice = int(bus['Business']['PDatabase']['Price'])
+ int(bus['Business']['POperationSystem']['Price']) +
int(bus['Business']['PCMS']['Price']) +
int(bus['Business']['PNetwork']['Price'])
                  if tprice <=
int(usereq['UserRequirements']['Optimization']['MAXPrice']):
                        b=bus
                        b['Total']={'TotalPrice':tprice}
                        global counter
                        global flag
                        counter += 1
                        b['_id']=counter
                        result=preso.insert_one(b)
                        print counter
                        channel.basic_publish(exchange='MainX',
routing_key='resolution', body='Price')

while "true":
```

```
    result = channel.queue_declare(queue="PriceQ")#, exclusive=True,
arguments=args)
    if not result.method.message_count == 0:
        method, properties, body = channel.basic_get(queue='PriceQ',
no_ack=True)
        callback(channel, method, properties, body)
```

QFinalResolution.py

```
import pika
import pymongo
from pymongo import MongoClient
import json


#Connect to rabbitmq
connection =
pika.BlockingConnection(pika.ConnectionParameters(host='localhost'))
channel = connection.channel()

channel.exchange_declare(exchange='MainX', type='topic')
result = channel.queue_declare(queue="ResolutionQ")#, exclusive=True,
arguments=args)
queue_name = result.method.queue


channel.queue_bind(exchange='MainX', queue=queue_name,
routing_key='resolution')

#Connect to pipeline db
client = MongoClient()
db = client.pipeline

#Get collections needed
req = db.req
preso = db.pricereso
reso = db.resolution

#Get user requirements
usereq=db.req.find_one()

print ' [*] Resolution waiting for messages. To exit press CTRL+C'

counter = 0 #counter for mongo id
def callback(ch, method, properties, body):
      if preso.count()>0:
            for pr in preso.find():
                  preso.delete_many({'_id':pr['_id']})
                  tavailability =
(int(pr['Business']['PDatabase']['Availability']) +
int(pr['Business']['POperationSystem']['Availability']) +
int(pr['Business']['PCMS']['Availability']) +
int(pr['Business']['PNetwork']['Availability']))/4
                  if tavailability >=
float(usereq['UserRequirements']['Optimization']['MINAvailability']):
                        tsecurity =
(int(pr['Business']['PDatabase']['Security']) +
int(pr['Business']['POperationSystem']['Security']) +
```

```
int(pr['Business']['PCMS']['Security']) +
int(pr['Business']['PNetwork']['Security']))/4
                        if tsecurity >=
float(usereq['UserRequirements']['Optimization']['MINSecurity']):
                                treputation =
(int(pr['Business']['PDatabase']['Reputation']) +
int(pr['Business']['POperationSystem']['Reputation']) +
int(pr['Business']['PCMS']['Reputation']) +
int(pr['Business']['PNetwork']['Reputation']))/4
                                if treputation >=
float(usereq['UserRequirements']['Optimization']['MINReputation']):
                                    treliability =
(int(pr['Business']['PDatabase']['Reliability']) +
int(pr['Business']['POperationSystem']['Reliability']) +
int(pr['Business']['PCMS']['Reliability']) +
int(pr['Business']['PNetwork']['Reliability']))/4
                                    if treliability >=
float(usereq['UserRequirements']['Optimization']['MINReliability']):
                                        tconfidentiality=0
                                        if
pr['Business']['PDatabase']['Confidentiality'] == 'YES':
                                            tconfidentiality+=1
                                        if
pr['Business']['POperationSystem']['Confidentiality'] == 'YES':
                                            tconfidentiality+=1
                                        if
pr['Business']['PCMS']['Confidentiality'] == 'YES':
                                            tconfidentiality+=1
                                        if
pr['Business']['PNetwork']['Confidentiality'] == 'YES':
                                            tconfidentiality+=1

      tconfidentiality=tconfidentiality/4
                                        if tconfidentiality >= 0.5:

      stconfidentiality='YES'
                                        else:

      stconfidentiality='NO'
                                        if stconfidentiality ==
usereq['UserRequirements']['Optimization']['Confidentiality']:
                                            ttotal =
0.3*tavailability + 0.6*tsecurity + 0.5*treputation +
0.7*treliability + 0.5*tconfidentiality -
0.005*pr['Total']['TotalPrice']
                                            p=pr

      p['Total']['TotalAvailability']=int(tavailability)

      p['Total']['TotalSecurity']=int(tsecurity)

      p['Total']['TotalReputation']=int(treputation)

      p['Total']['TotalReliability']=int(treliability)

      p['Total']['TotalConfidentiality']=stconfidentiality

      p['Total']['TotalTotal']=ttotal

                                            global counter
                                            counter += 1
```

```
                                                p['_id']=counter

      result=reso.insert_one(p)
                                                #time.sleep(1)
                                                print "[x] Resolution
sent results to mongo"
                                          else:
                                                print "Rejected"
                                    else:
                                          print "Rejected"
                              else:
                                    print "Rejected"
                        else:
                              print "Rejected"
                  else:
                        print "Rejected"


#Waiting for message from price unit
while "true":
      result = channel.queue_declare(queue="ResolutionQ")#,
exclusive=True, arguments=args)
      if not result.method.message_count == 0:
            method, properties, body =
channel.basic_get(queue='ResolutionQ', no_ack=True)
            callback(channel, method, properties, body)
```

Το αρχείο εκκίνησης της διαδικασίας

Initiate.py

```
import pika
import pymongo
from pymongo import MongoClient
import json


connection =
pika.BlockingConnection(pika.ConnectionParameters(host='localhost'))
channel = connection.channel()

result = channel.queue_declare(queue='Technical')
channel.basic_publish(exchange='', routing_key='Technical', body='')
print " [x] Sent tech to technical"
```

Το αρχείο εισαγωγής αρχείων JSON στη MongoDB

Initiate.py

```
import pika
import pymongo
from pymongo import MongoClient
import json

#Connect to pipeline db
client = MongoClient()
```

```
db = client.pipeline

#Get collections needed
i=db.req

with open('usereq.json') as data_file:
      data = json.load(data_file)
result=i.insert_one(data)
```

Τα JSON αρχεία για τις απαιτήσεις του χρήστη, τα τεχνικά χαρακτηριστικά και τα επιχειρηματικά χαρακτηριστικά των υπηρεσιών.

UserRequirements.json

```
{
      "UserRequirements":{
            "Optimization":{
                  "MAXPrice":"3000",
                  "MINAvailability":"90",
                  "MINSecurity":"2",
                  "MINReputation":"5",
                  "MINReliability":"4",
                  "Confidentiality":"YES"
            },
            "Service":{
                  "Database":{
                        "OP":"x86",
                        "Type":"SQL"
                  },
                  "OperationSystem":{
                        "Type":"Linux",
                        "OP":"x86"
                  },
                  "CMS":{
                        "OP":"x86"
                  },
                  "Network":{
                        "Type":"copper"
                  }
            }
      }
}
```

TechnicalList.json

```
{
      "database":[
            {
                  "id":"1002003201",
                  "name":"MySQL 5.3 Cluster",
                  "OP":"x86",
                  "Type":"SQL"
            },
            {
                  "id":"1002003202",
                  "name":"Apache Derby",
```

```
        "OP":"x86",
        "Type":"SQL"
    },
    {
        "id":"1002003203",
        "name":"MariaDB",
        "OP":"x86",
        "Type":"SQL"
    },
    {
        "id":"1002003204",
        "name":"MariaDB",
        "OP":"x64",
        "Type":"SQL"
    },
    {
        "id":"1002003205",
        "name":"PostgreSQL",
        "OP":"x86",
        "Type":"SQL"
    },
    {
        "id":"1002003206",
        "name":"IBM DB2",
        "OP":"x86",
        "Type":"SQL"
    },
    {
        "id":"1002003207",
        "name":"Amazon SimpleDB",
        "OP":"x86",
        "Type":"NoSQL"
    },
    {
        "id":"1002003208",
        "name":"Microsoft SQL Server",
        "OP":"x86",
        "Type":"SQL"
    },
    {
        "id":"1002003209",
        "name":"Microsoft SQL Server",
        "OP":"x64",
        "Type":"SQL"
    },
    {
        "id":"1002003210",
        "name":"IBM Informix",
        "OP":"x64",
        "Type":"NoSQL"
    },
    {
        "id":"1002003211",
        "name":"Amazon SimpleDB",
        "OP":"x64",
        "Type":"NoSQL"
    },
    {
        "id":"1002003212",
        "name":"Cassandra",
        "OP":"x86",
```

```
        "Type":"NoSQL"
    },
    {
        "id":"1002003213",
        "name":"MongoDB 3.06",
        "OP":"x86",
        "Type":"NoSQL"
    },
    {
        "id":"1002003214",
        "name":"Cassandra",
        "OP":"x64",
        "Type":"NoSQL"
    },
    {
        "id":"1002003215",
        "name":"MonetDB",
        "OP":"x86",
        "Type":"NoSQL"
    },
    {
        "id":"1002003216",
        "name":"MongoDB",
        "OP":"x86",
        "Type":"NoSQL"
    },
    {
        "id":"1002003217",
        "name":"MongoDB",
        "OP":"x64",
        "Type":"NoSQL"
    },
    {
        "id":"1002003218",
        "name":"CouchDB",
        "OP":"x86",
        "Type":"NoSQL"
    },
    {
        "id":"1002003219",
        "name":"DynamoDB",
        "OP":"x86",
        "Type":"NoSQL"
    },
    {
        "id":"1002003220",
        "name":"Azure Table Storage",
        "OP":"x86",
        "Type":"NoSQL"
    },
    {
        "id":"1002003221",
        "name":"Aerospike",
        "OP":"x86",
        "Type":"NoSQL"
    },
    {
        "id":"1002003222",
        "name":"Voldemort",
        "OP":"x64",
        "Type":"NoSQL"
```

```
        },
        {
                "id":"1002003223",
                "name":"Neo4J",
                "OP":"x64",
                "Type":"NoSQL"
        },
        {
                "id":"1002003224",
                "name":"FatDB",
                "OP":"x64",
                "Type":"NoSQL"
        },
        {
                "id":"1002003225",
                "name":"CortexDB",
                "OP":"x64",
                "Type":"NoSQL"
        },
        {
                "id":"1002003226",
                "name":"Trinity",
                "OP":"x64",
                "Type":"NoSQL"
        },
        {
                "id":"1002003227",
                "name":"Trinity",
                "OP":"x86",
                "Type":"NoSQL"
        },
        {
                "id":"1002003228",
                "name":"FatDB",
                "OP":"x86",
                "Type":"NoSQL"
        },
        {
                "id":"1002003229",
                "name":"CortexDB",
                "OP":"x86",
                "Type":"NoSQL"
        },
        {
                "id":"1002003230",
                "name":"Neo4J",
                "OP":"x86",
                "Type":"NoSQL"
        }
],

"OperationSystem":[
        {
                "id":"2002003201",
                "Distribution":"Debian",
                "Type":"Linux",
                "OP":"x86"
        },
        {
                "id":"2002003202",
                "Distribution":"RedHat",
```

```
                "Type":"Linux",
                "OP":"x64"
        },
        {
                "id":"2002003203",
                "Distribution":"8.1",
                "Type":"Windows",
                "OP":"x64"
        },
        {
                "id":"2002003204",
                "Distribution":"10",
                "Type":"Windows",
                "OP":"x86"
        },
        {
                "id":"2002003205",
                "Distribution":"10",
                "Type":"Windows",
                "OP":"x64"
        },
        {
                "id":"2002003206",
                "Distribution":"10.11: El Capitan",
                "Type":"MacOS X",
                "OP":"x64"
        },
        {
                "id":"2002003207",
                "Distribution":"10.10: Yosemite",
                "Type":"MacOS X",
                "OP":"x64"
        },
        {
                "id":"2002003208",
                "Distribution":"10.9: Mavericks",
                "Type":"MacOS X",
                "OP":"x64"
        },
        {
                "id":"2002003209",
                "Distribution":"10.6: Snow Leopard",
                "Type":"MacOS X",
                "OP":"x64"
        },
        {
                "id":"2002003210",
                "Distribution":"OpenSuSe 42.1",
                "Type":"Linux",
                "OP":"x86"
        }
],

"CMS":[
        {
                "id":"3002003201",
                "Name":"Drupal 7",
                "OP":"x86"
        },
        {
                "id":"3002003202",
```

```
                "Name":"Drupal 8 beta 1",
                "OP":"x86"
        },
        {
                "id":"3002003203",
                "Name":"Drupal 9 beta 1",
                "OP":"x86"
        },
        {
                "id":"3002003204",
                "Name":"Drupal 9 beta 1",
                "OP":"x64"
        },
        {
                "id":"3002003205",
                "Name":"Drupal 8 beta 1",
                "OP":"x64"
        },
        {
                "id":"3002003206",
                "Name":"Joomla 3",
                "OP":"x86"
        },
        {
                "id":"3002003207",
                "Name":"Joomla 3",
                "OP":"x64"
        },
        {
                "id":"3002003208",
                "Name":"WordPress 3.2.1",
                "OP":"x86"
        },
        {
                "id":"3002003209",
                "Name":"WordPress 3.2.1",
                "OP":"x64"
        },
        {
                "id":"3002003210",
                "Name":"WordPress 3.2",
                "OP":"x64"
        }
],

"Network":[
        {
                "id":"4002003201",
                "Name":"ADSL Connection",
                "Type":"copper"
        },
        {
                "id":"4002003202",
                "Name":"VDSL Connection",
                "Type":"copper"
        },
        {
                "id":"4002003203",
                "Name":"VDSL Connection",
                "Type":"fiber"
        },
```

```
            {
                  "id":"4002003204",
                  "Name":"T3 Connection",
                  "Type":"fiber"
            },
            {
                  "id":"4002003205",
                  "Name":"ADSL2+  Connection",
                  "Type":"fiber"
            }
      ]
}
```


BusinessList.json


```
{
      "PDatabase":[
            {
                  "id":"1002003201",
                  "ServiceProvider":"Oracle Provider",
                  "DName":"MySQL 5.3 Cluster",
                  "Availability":"95",
                  "Security":"5",
                  "Reputation":"4",
                  "Reliability":"4",
                  "Confidentiality":"YES",
                  "Price":"1000"
            },
            {
                  "id":"1002003201",
                  "ServiceProvider":"Amazon",
                  "DName":"MySQL 5.3 Cluster",
                  "Availability":"97",
                  "Security":"5",
                  "Reputation":"5",
                  "Reliability":"4",
                  "Confidentiality":"YES",
                  "Price":"1100"
            },
            {
                  "id":"1002003201",
                  "ServiceProvider":"SalesForge",
                  "DName":"MySQL 5.3 Cluster",
                  "Availability":"93",
                  "Security":"5",
                  "Reputation":"5",
                  "Reliability":"3",
                  "Confidentiality":"NO",
                  "Price":"900"
            },
            {
                  "id":"1002003201",
                  "ServiceProvider":"SalesForge",
                  "DName":"MySQL 5.3 Cluster",
                  "Availability":"93",
                  "Security":"5",
                  "Reputation":"5",
                  "Reliability":"3",
                  "Confidentiality":"NO",
                  "Price":"900"
```

```
},
{
        "id":"1002003202",
        "ServiceProvider":"SalesForge",
        "DName":"Apache Derby",
        "Availability":"93",
        "Security":"5",
        "Reputation":"5",
        "Reliability":"3",
        "Confidentiality":"NO",
        "Price":"900"
},
{
        "id":"1002003202",
        "ServiceProvider":"Amazon",
        "DName":"Apache Derby",
        "Availability":"97",
        "Security":"4",
        "Reputation":"5",
        "Reliability":"2",
        "Confidentiality":"YES",
        "Price":"800"
},
{
        "id":"1002003202",
        "ServiceProvider":"Cloudera",
        "DName":"Apache Derby",
        "Availability":"90",
        "Security":"5",
        "Reputation":"3",
        "Reliability":"3",
        "Confidentiality":"YES",
        "Price":"700"
},
{
        "id":"1002003202",
        "ServiceProvider":"VMWare",
        "DName":"Apache Derby",
        "Availability":"91",
        "Security":"4",
        "Reputation":"3",
        "Reliability":"3",
        "Confidentiality":"YES",
        "Price":"760"
},
{
        "id":"1002003202",
        "ServiceProvider":"RackSpace",
        "DName":"Apache Derby",
        "Availability":"98",
        "Security":"4",
        "Reputation":"2",
        "Reliability":"3",
        "Confidentiality":"YES",
        "Price":"860"
},
{
        "id":"1002003203",
        "ServiceProvider":"RackSpace",
        "DName":"MariaDB",
        "Availability":"95",
```

```
        "Security":"4",
        "Reputation":"2",
        "Reliability":"3",
        "Confidentiality":"YES",
        "Price":"860"
    },
    {
        "id":"1002003203",
        "ServiceProvider":"Amazon",
        "DName":"MariaDB",
        "Availability":"80",
        "Security":"4",
        "Reputation":"3",
        "Reliability":"3",
        "Confidentiality":"YES",
        "Price":"960"
    },
    {
        "id":"1002003204",
        "ServiceProvider":"RackSpace",
        "DName":"MariaDB",
        "Availability":"95",
        "Security":"4",
        "Reputation":"2",
        "Reliability":"3",
        "Confidentiality":"YES",
        "Price":"860"
    },
    {
        "id":"1002003204",
        "ServiceProvider":"Amazon",
        "DName":"MariaDB",
        "Availability":"80",
        "Security":"4",
        "Reputation":"3",
        "Reliability":"3",
        "Confidentiality":"YES",
        "Price":"960"
    },
    {
        "id":"1002003205",
        "ServiceProvider":"Amazon",
        "DName":"PostgreSQL",
        "Availability":"90",
        "Security":"5",
        "Reputation":"3",
        "Reliability":"3",
        "Confidentiality":"NO",
        "Price":"1260"
    },
    {
        "id":"1002003205",
        "ServiceProvider":"VMWare",
        "DName":"PostgreSQL",
        "Availability":"91",
        "Security":"3",
        "Reputation":"2",
        "Reliability":"3",
        "Confidentiality":"NO",
        "Price":"680"
    },
```

```
{
        "id":"1002003205",
        "ServiceProvider":"RedHat",
        "DName":"PostgreSQL",
        "Availability":"98",
        "Security":"5",
        "Reputation":"2",
        "Reliability":"3",
        "Confidentiality":"NO",
        "Price":"900"
},
{
        "id":"1002003206",
        "ServiceProvider":"RedHat",
        "DName":"IBM DB2",
        "Availability":"94",
        "Security":"5",
        "Reputation":"2",
        "Reliability":"5",
        "Confidentiality":"YES",
        "Price":"1900"
},
{
        "id":"1002003206",
        "ServiceProvider":"RackSpace",
        "DName":"IBM DB2",
        "Availability":"97",
        "Security":"4",
        "Reputation":"5",
        "Reliability":"5",
        "Confidentiality":"YES",
        "Price":"2900"
},
{
        "id":"1002003206",
        "ServiceProvider":"IBM",
        "DName":"IBM DB2",
        "Availability":"98",
        "Security":"5",
        "Reputation":"5",
        "Reliability":"5",
        "Confidentiality":"YES",
        "Price":"3300"
},
{
        "id":"1002003207",
        "ServiceProvider":"Amazon",
        "DName":"Amazon SimpleDB",
        "Availability":"66",
        "Security":"3",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"300"
},
{
        "id":"1002003208",
        "ServiceProvider":"Microsoft",
        "DName":"Microsoft SQL Server",
        "Availability":"77",
        "Security":"4",
```

```
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"1200"
    },
    {
        "id":"1002003209",
        "ServiceProvider":"Microsoft",
        "DName":"Microsoft SQL Server",
        "Availability":"77",
        "Security":"4",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"1200"
    },
    {
        "id":"1002003210",
        "ServiceProvider":"Citrix",
        "DName":"IBM Informix",
        "Availability":"87",
        "Security":"5",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"2200"
    },
    {
        "id":"1002003210",
        "ServiceProvider":"IBM",
        "DName":"IBM Informix",
        "Availability":"89",
        "Security":"5",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"2000"
    },
    {
        "id":"1002003211",
        "ServiceProvider":"Amazon",
        "DName":"Amazon SimpleDB",
        "Availability":"79",
        "Security":"4",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"500"
    },
    {
        "id":"1002003212",
        "ServiceProvider":"Amazon",
        "DName":"Cassandra",
        "Availability":"83",
        "Security":"4",
        "Reputation":"3",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"800"
    },
    {
```

```
        "id":"1002003212",
        "ServiceProvider":"IBM",
        "DName":"Cassandra",
        "Availability":"87",
        "Security":"3",
        "Reputation":"3",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"870"
    },
    {
        "id":"1002003212",
        "ServiceProvider":"FaceBook",
        "DName":"Cassandra",
        "Availability":"84",
        "Security":"4",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"570"
    },
    {
        "id":"1002003213",
        "ServiceProvider":"IBM",
        "DName":"MongoDB 3.06",
        "Availability":"89",
        "Security":"3",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"999"
    },
    {
        "id":"1002003214",
        "ServiceProvider":"Amazon",
        "DName":"Cassandra",
        "Availability":"77",
        "Security":"5",
        "Reputation":"3",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"700"
    },
    {
        "id":"1002003214",
        "ServiceProvider":"SalesForge",
        "DName":"Cassandra",
        "Availability":"83",
        "Security":"4",
        "Reputation":"3",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"800"
    },
    {
        "id":"1002003214",
        "ServiceProvider":"IBM",
        "DName":"Cassandra",
        "Availability":"87",
        "Security":"3",
        "Reputation":"3",
```

```
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"870"
    },
    {
        "id":"1002003214",
        "ServiceProvider":"FaceBook",
        "DName":"Cassandra",
        "Availability":"84",
        "Security":"4",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"570"
    },
    {
        "id":"1002003215",
        "ServiceProvider":"FaceBook",
        "DName":"MonetDB",
        "Availability":"55",
        "Security":"5",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"910"
    },
    {
        "id":"1002003215",
        "ServiceProvider":"VMWare",
        "DName":"MonetDB",
        "Availability":"55",
        "Security":"5",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"930"
    },
    {
        "id":"1002003216",
        "ServiceProvider":"Citrix",
        "DName":"MongoDB",
        "Availability":"88",
        "Security":"1",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"200"
    },
    {
        "id":"1002003216",
        "ServiceProvider":"RedHat",
        "DName":"MongoDB",
        "Availability":"89",
        "Security":"2",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"240"
    },
    {
        "id":"1002003216",
```

```
        "ServiceProvider":"RackSpace",
        "DName":"MongoDB",
        "Availability":"88",
        "Security":"1",
        "Reputation":"2",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"190"
    },
    {
        "id":"1002003216",
        "ServiceProvider":"10Gen",
        "DName":"MongoDB",
        "Availability":"90",
        "Security":"2",
        "Reputation":"4",
        "Reliability":"5",
        "Confidentiality":"YES",
        "Price":"220"
    },
    {
        "id":"1002003216",
        "ServiceProvider":"BlueHost",
        "DName":"MongoDB",
        "Availability":"98",
        "Security":"2",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"100"
    },
    {
        "id":"1002003217",
        "ServiceProvider":"Citrix",
        "DName":"MongoDB",
        "Availability":"88",
        "Security":"1",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"200"
    },
    {
        "id":"1002003217",
        "ServiceProvider":"RackSpace",
        "DName":"MongoDB",
        "Availability":"88",
        "Security":"1",
        "Reputation":"2",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"190"
    },
    {
        "id":"1002003217",
        "ServiceProvider":"10Gen",
        "DName":"MongoDB",
        "Availability":"90",
        "Security":"2",
        "Reputation":"4",
        "Reliability":"5",
```

```
        "Confidentiality":"YES",
        "Price":"220"
    },
    {
        "id":"1002003217",
        "ServiceProvider":"BlueHost",
        "DName":"MongoDB",
        "Availability":"98",
        "Security":"2",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"100"
    },
    {
        "id":"1002003218",
        "ServiceProvider":"BlueHost",
        "DName":"CouchDB",
        "Availability":"68",
        "Security":"3",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"300"
    },
    {
        "id":"1002003219",
        "ServiceProvider":"Amazon",
        "DName":"DynamoDB",
        "Availability":"53",
        "Security":"1",
        "Reputation":"2",
        "Reliability":"5",
        "Confidentiality":"YES",
        "Price":"300"
    },
    {
        "id":"1002003219",
        "ServiceProvider":"RedHat",
        "DName":"DynamoDB",
        "Availability":"63",
        "Security":"2",
        "Reputation":"3",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"500"
    },
    {
        "id":"1002003220",
        "ServiceProvider":"Microsoft",
        "DName":"Azure Table Storage",
        "Availability":"23",
        "Security":"2",
        "Reputation":"2",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"1600"
    },
    {
        "id":"1002003221",
        "ServiceProvider":"Amazon",
```

```
        "DName":"Aerospike",
        "Availability":"63",
        "Security":"4",
        "Reputation":"2",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"600"
    },
    {
        "id":"1002003221",
        "ServiceProvider":"VMWare",
        "DName":"Aerospike",
        "Availability":"90",
        "Security":"4",
        "Reputation":"3",
        "Reliability":"3",
        "Confidentiality":"YES",
        "Price":"700"
    },
    {
        "id":"1002003221",
        "ServiceProvider":"OpenShift",
        "DName":"Aerospike",
        "Availability":"92",
        "Security":"4",
        "Reputation":"3",
        "Reliability":"3",
        "Confidentiality":"YES",
        "Price":"710"
    },
    {
        "id":"1002003222",
        "ServiceProvider":"Amazon",
        "DName":"Voldemort",
        "Availability":"88",
        "Security":"4",
        "Reputation":"2",
        "Reliability":"3",
        "Confidentiality":"NO",
        "Price":"210"
    },
    {
        "id":"1002003222",
        "ServiceProvider":"Google",
        "DName":"Voldemort",
        "Availability":"90",
        "Security":"4",
        "Reputation":"2",
        "Reliability":"3",
        "Confidentiality":"NO",
        "Price":"240"
    },
    {
        "id":"1002003223",
        "ServiceProvider":"Google",
        "DName":"Neo4J",
        "Availability":"78",
        "Security":"1",
        "Reputation":"5",
        "Reliability":"2",
        "Confidentiality":"NO",
```

```
        "Price":"310"
    },
    {
        "id":"1002003223",
        "ServiceProvider":"Amazon",
        "DName":"Neo4J",
        "Availability":"72",
        "Security":"1",
        "Reputation":"5",
        "Reliability":"2",
        "Confidentiality":"YES",
        "Price":"320"
    },
    {
        "id":"1002003223",
        "ServiceProvider":"RackSpace",
        "DName":"Neo4J",
        "Availability":"62",
        "Security":"4",
        "Reputation":"4",
        "Reliability":"2",
        "Confidentiality":"YES",
        "Price":"220"
    },
    {
        "id":"1002003223",
        "ServiceProvider":"RedHat",
        "DName":"Neo4J",
        "Availability":"52",
        "Security":"4",
        "Reputation":"4",
        "Reliability":"5",
        "Confidentiality":"YES",
        "Price":"280"
    },
    {
        "id":"1002003230",
        "ServiceProvider":"Google",
        "DName":"Neo4J",
        "Availability":"78",
        "Security":"1",
        "Reputation":"5",
        "Reliability":"2",
        "Confidentiality":"NO",
        "Price":"310"
    },
    {
        "id":"1002003230",
        "ServiceProvider":"Amazon",
        "DName":"Neo4J",
        "Availability":"72",
        "Security":"1",
        "Reputation":"5",
        "Reliability":"2",
        "Confidentiality":"YES",
        "Price":"320"
    },
    {
        "id":"1002003230",
        "ServiceProvider":"RackSpace",
        "DName":"Neo4J",
```

```
        "Availability":"62",
        "Security":"4",
        "Reputation":"4",
        "Reliability":"2",
        "Confidentiality":"YES",
        "Price":"220"
    },
    {
        "id":"1002003230",
        "ServiceProvider":"RedHat",
        "DName":"Neo4J",
        "Availability":"52",
        "Security":"4",
        "Reputation":"4",
        "Reliability":"5",
        "Confidentiality":"YES",
        "Price":"280"
    },
    {
        "id":"1002003224",
        "ServiceProvider":"Amazon",
        "DName":"FatDB",
        "Availability":"42",
        "Security":"4",
        "Reputation":"4",
        "Reliability":"2",
        "Confidentiality":"YES",
        "Price":"380"
    },
    {
        "id":"1002003224",
        "ServiceProvider":"OpenShift",
        "DName":"FatDB",
        "Availability":"42",
        "Security":"3",
        "Reputation":"4",
        "Reliability":"2",
        "Confidentiality":"YES",
        "Price":"310"
    },
    {
        "id":"1002003224",
        "ServiceProvider":"RedHat",
        "DName":"FatDB",
        "Availability":"62",
        "Security":"1",
        "Reputation":"4",
        "Reliability":"2",
        "Confidentiality":"NO",
        "Price":"220"
    },
    {
        "id":"1002003228",
        "ServiceProvider":"Amazon",
        "DName":"FatDB",
        "Availability":"42",
        "Security":"4",
        "Reputation":"4",
        "Reliability":"2",
        "Confidentiality":"YES",
        "Price":"380"
```

```
},
{
        "id":"1002003228",
        "ServiceProvider":"OpenShift",
        "DName":"FatDB",
        "Availability":"42",
        "Security":"3",
        "Reputation":"4",
        "Reliability":"2",
        "Confidentiality":"YES",
        "Price":"310"
},
{
        "id":"1002003228",
        "ServiceProvider":"RedHat",
        "DName":"FatDB",
        "Availability":"62",
        "Security":"1",
        "Reputation":"4",
        "Reliability":"2",
        "Confidentiality":"NO",
        "Price":"220"
},
{
        "id":"1002003225",
        "ServiceProvider":"Google",
        "DName":"CortexDB",
        "Availability":"77",
        "Security":"4",
        "Reputation":"4",
        "Reliability":"3",
        "Confidentiality":"YES",
        "Price":"320"
},
{
        "id":"1002003225",
        "ServiceProvider":"BlueHost",
        "DName":"CortexDB",
        "Availability":"87",
        "Security":"2",
        "Reputation":"4",
        "Reliability":"3",
        "Confidentiality":"YES",
        "Price":"120"
},
{
        "id":"1002003225",
        "ServiceProvider":"RackSpace",
        "DName":"CortexDB",
        "Availability":"81",
        "Security":"2",
        "Reputation":"1",
        "Reliability":"3",
        "Confidentiality":"YES",
        "Price":"360"
},
{
        "id":"1002003229",
        "ServiceProvider":"Google",
        "DName":"CortexDB",
        "Availability":"77",
```

```
                "Security":"4",
                "Reputation":"4",
                "Reliability":"3",
                "Confidentiality":"YES",
                "Price":"320"
        },
        {
                "id":"1002003229",
                "ServiceProvider":"BlueHost",
                "DName":"CortexDB",
                "Availability":"87",
                "Security":"2",
                "Reputation":"4",
                "Reliability":"3",
                "Confidentiality":"YES",
                "Price":"120"
        },
        {
                "id":"1002003229",
                "ServiceProvider":"RackSpace",
                "DName":"CortexDB",
                "Availability":"81",
                "Security":"2",
                "Reputation":"1",
                "Reliability":"3",
                "Confidentiality":"YES",
                "Price":"360"
        },
        {
                "id":"1002003227",
                "ServiceProvider":"Amazon",
                "DName":"Trinity",
                "Availability":"89",
                "Security":"2",
                "Reputation":"4",
                "Reliability":"3",
                "Confidentiality":"YES",
                "Price":"160"
        },
        {
                "id":"1002003226",
                "ServiceProvider":"Amazon",
                "DName":"Trinity",
                "Availability":"89",
                "Security":"2",
                "Reputation":"4",
                "Reliability":"3",
                "Confidentiality":"YES",
                "Price":"160"
        }
],
"POperationSystem":[
        {
                "id":"2002003201",
                "ServiceProvider":"Amazon",
                "Availability":"99",
                "Security":"5",
                "Reputation":"5",
                "Reliability":"4",
                "Confidentiality":"YES",
                "Price":"100"
```

```
},
{
        "id":"2002003202",
        "ServiceProvider":"Amazon",
        "Availability":"99",
        "Security":"5",
        "Reputation":"5",
        "Reliability":"5",
        "Confidentiality":"YES",
        "Price":"300"
},
{
        "id":"2002003202",
        "ServiceProvider":"RedHat",
        "Availability":"91",
        "Security":"5",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"500"
},
{
        "id":"2002003202",
        "ServiceProvider":"SalesForge",
        "Availability":"88",
        "Security":"5",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"440"
},
{
        "id":"2002003203",
        "ServiceProvider":"Amazon",
        "Availability":"69",
        "Security":"5",
        "Reputation":"5",
        "Reliability":"5",
        "Confidentiality":"YES",
        "Price":"500"
},
{
        "id":"2002003203",
        "ServiceProvider":"Microsoft",
        "Availability":"91",
        "Security":"5",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"500"
},
{
        "id":"2002003203",
        "ServiceProvider":"VMWare",
        "Availability":"78",
        "Security":"5",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"540"
},
```

```
{
        "id":"2002003204",
        "ServiceProvider":"Amazon",
        "Availability":"73",
        "Security":"3",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"500"
},
{
        "id":"2002003204",
        "ServiceProvider":"Microsoft",
        "Availability":"91",
        "Security":"3",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"400"
},
{
        "id":"2002003204",
        "ServiceProvider":"VMWare",
        "Availability":"78",
        "Security":"5",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"440"
},
{
        "id":"2002003205",
        "ServiceProvider":"Amazon",
        "Availability":"73",
        "Security":"3",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"500"
},
{
        "id":"2002003205",
        "ServiceProvider":"Microsoft",
        "Availability":"91",
        "Security":"3",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"400"
},
{
        "id":"2002003205",
        "ServiceProvider":"VMWare",
        "Availability":"78",
        "Security":"5",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"440"
},
{
```

```
        "id":"2002003206",
        "ServiceProvider":"VMWare",
        "Availability":"78",
        "Security":"5",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"740"
    },
    {
        "id":"2002003206",
        "ServiceProvider":"VMWare",
        "Availability":"78",
        "Security":"5",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"700"
    },
    {
        "id":"2002003206",
        "ServiceProvider":"Amazon",
        "Availability":"78",
        "Security":"5",
        "Reputation":"3",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"410"
    },
    {
        "id":"2002003207",
        "ServiceProvider":"VMWare",
        "Availability":"78",
        "Security":"5",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"640"
    },
    {
        "id":"2002003207",
        "ServiceProvider":"VMWare",
        "Availability":"78",
        "Security":"5",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"600"
    },
    {
        "id":"2002003207",
        "ServiceProvider":"Amazon",
        "Availability":"78",
        "Security":"5",
        "Reputation":"3",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"400"
    },
    {
        "id":"2002003208",
```

```
        "ServiceProvider":"VMWare",
        "Availability":"78",
        "Security":"3",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"540"
    },
    {
        "id":"2002003208",
        "ServiceProvider":"VMWare",
        "Availability":"78",
        "Security":"3",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"400"
    },
    {
        "id":"2002003208",
        "ServiceProvider":"Amazon",
        "Availability":"78",
        "Security":"2",
        "Reputation":"3",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"400"
    },
    {
        "id":"2002003209",
        "ServiceProvider":"VMWare",
        "Availability":"78",
        "Security":"2",
        "Reputation":"4",
        "Reliability":"2",
        "Confidentiality":"YES",
        "Price":"330"
    },
    {
        "id":"2002003209",
        "ServiceProvider":"VMWare",
        "Availability":"78",
        "Security":"5",
        "Reputation":"4",
        "Reliability":"2",
        "Confidentiality":"NO",
        "Price":"300"
    },
    {
        "id":"2002003209",
        "ServiceProvider":"Amazon",
        "Availability":"78",
        "Security":"5",
        "Reputation":"3",
        "Reliability":"2",
        "Confidentiality":"YES",
        "Price":"200"
    },
    {
        "id":"2002003210",
        "ServiceProvider":"Amazon",
```

```
                "Availability":"78",
                "Security":"5",
                "Reputation":"4",
                "Reliability":"4",
                "Confidentiality":"YES",
                "Price":"230"
        },
        {
                "id":"2002003210",
                "ServiceProvider":"Amazon",
                "Availability":"78",
                "Security":"5",
                "Reputation":"3",
                "Reliability":"5",
                "Confidentiality":"YES",
                "Price":"270"
        },
        {
                "id":"2002003210",
                "ServiceProvider":"Amazon",
                "Availability":"78",
                "Security":"5",
                "Reputation":"3",
                "Reliability":"2",
                "Confidentiality":"NO",
                "Price":"210"
        },
        {
                "id":"2002003210",
                "ServiceProvider":"VMWare",
                "Availability":"78",
                "Security":"5",
                "Reputation":"5",
                "Reliability":"2",
                "Confidentiality":"NO",
                "Price":"410"
        },
        {
                "id":"2002003210",
                "ServiceProvider":"VMWare",
                "Availability":"78",
                "Security":"5",
                "Reputation":"3",
                "Reliability":"5",
                "Confidentiality":"YES",
                "Price":"440"
        }
    ],
    "PCMS":[
        {
                "id":"3002003201",
                "ServiceProvider":"Drupal Inc",
                "Availability":"90",
                "Security":"5",
                "Reputation":"5",
                "Reliability":"4",
                "Confidentiality":"YES",
                "Price":"100"
        },
        {
                "id":"3002003201",
```

```
        "ServiceProvider":"Drupal Inc",
        "Availability":"92",
        "Security":"5",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"150"
    },
    {
        "id":"3002003202",
        "ServiceProvider":"Drupal Inc",
        "Availability":"90",
        "Security":"5",
        "Reputation":"3",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"90"
    },
    {
        "id":"3002003202",
        "ServiceProvider":"Drupal Inc",
        "Availability":"92",
        "Security":"5",
        "Reputation":"3",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"90"
    },
    {
        "id":"3002003203",
        "ServiceProvider":"Drupal Inc",
        "Availability":"90",
        "Security":"5",
        "Reputation":"3",
        "Reliability":"5",
        "Confidentiality":"YES",
        "Price":"200"
    },
    {
        "id":"3002003203",
        "ServiceProvider":"HostGator",
        "Availability":"89",
        "Security":"5",
        "Reputation":"3",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"190"
    },
    {
        "id":"3002003204",
        "ServiceProvider":"Drupal Inc",
        "Availability":"89",
        "Security":"5",
        "Reputation":"3",
        "Reliability":"5",
        "Confidentiality":"NO",
        "Price":"200"
    },
    {
        "id":"3002003204",
        "ServiceProvider":"HostGator",
```

```
        "Availability":"88",
        "Security":"5",
        "Reputation":"3",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"190"
    },
    {
        "id":"3002003204",
        "ServiceProvider":"BlueHost",
        "Availability":"67",
        "Security":"5",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"250"
    },
    {
        "id":"3002003204",
        "ServiceProvider":"FatCow",
        "Availability":"76",
        "Security":"5",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"220"
    },
    {
        "id":"3002003205",
        "ServiceProvider":"BlueHost",
        "Availability":"82",
        "Security":"2",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"210"
    },
    {
        "id":"3002003205",
        "ServiceProvider":"FatCow",
        "Availability":"88",
        "Security":"3",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"210"
    },
    {
        "id":"3002003206",
        "ServiceProvider":"HostGator",
        "Availability":"56",
        "Security":"2",
        "Reputation":"3",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"90"
    },
    {
        "id":"3002003206",
        "ServiceProvider":"BlueHost",
        "Availability":"56",
```

```
        "Security":"1",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"50"
    },
    {
        "id":"3002003206",
        "ServiceProvider":"FatCow",
        "Availability":"44",
        "Security":"3",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"20"
    },
    {
        "id":"3002003207",
        "ServiceProvider":"HostGator",
        "Availability":"56",
        "Security":"2",
        "Reputation":"3",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"90"
    },
    {
        "id":"3002003207",
        "ServiceProvider":"BlueHost",
        "Availability":"56",
        "Security":"1",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"50"
    },
    {
        "id":"3002003207",
        "ServiceProvider":"FatCow",
        "Availability":"44",
        "Security":"3",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"20"
    },
    {
        "id":"3002003208",
        "ServiceProvider":"BlueHost",
        "Availability":"56",
        "Security":"5",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"NO",
        "Price":"510"
    },
    {
        "id":"3002003208",
        "ServiceProvider":"iPage",
        "Availability":"44",
        "Security":"3",
```

```
                "Reputation":"5",
                "Reliability":"4",
                "Confidentiality":"YES",
                "Price":"420"
        },
        {
                "id":"3002003209",
                "ServiceProvider":"HostGator",
                "Availability":"56",
                "Security":"5",
                "Reputation":"3",
                "Reliability":"4",
                "Confidentiality":"YES",
                "Price":"490"
        },
        {
                "id":"3002003209",
                "ServiceProvider":"BlueHost",
                "Availability":"56",
                "Security":"5",
                "Reputation":"4",
                "Reliability":"4",
                "Confidentiality":"NO",
                "Price":"510"
        },
        {
                "id":"3002003209",
                "ServiceProvider":"iPage",
                "Availability":"44",
                "Security":"3",
                "Reputation":"5",
                "Reliability":"4",
                "Confidentiality":"YES",
                "Price":"420"
        },
        {
                "id":"3002003210",
                "ServiceProvider":"iPage",
                "Availability":"44",
                "Security":"2",
                "Reputation":"4",
                "Reliability":"3",
                "Confidentiality":"NO",
                "Price":"300"
        }

    ],
    "PNetwork":[
        {
                "id":"4002003201",
                "ServiceProvider":"CosmOTE",
                "Availability":"80",
                "Security":"5",
                "Reputation":"5",
                "Reliability":"4",
                "Confidentiality":"YES",
                "Price":"110"
        },
        {
                "id":"4002003201",
                "ServiceProvider":"T-Mobile",
```

```
        "Availability":"70",
        "Security":"5",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"120"
    },
    {
        "id":"4002003201",
        "ServiceProvider":"Orange",
        "Availability":"70",
        "Security":"3",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"150"
    },
    {
        "id":"4002003201",
        "ServiceProvider":"Vodafone",
        "Availability":"90",
        "Security":"2",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"200"
    },
    {
        "id":"4002003201",
        "ServiceProvider":"Telefonica",
        "Availability":"73",
        "Security":"5",
        "Reputation":"3",
        "Reliability":"2",
        "Confidentiality":"NO",
        "Price":"10"
    },
    {
        "id":"4002003202",
        "ServiceProvider":"CosmOTE",
        "Availability":"80",
        "Security":"5",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"180"
    },
    {
        "id":"4002003202",
        "ServiceProvider":"T-Mobile",
        "Availability":"70",
        "Security":"5",
        "Reputation":"4",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"220"
    },
    {
        "id":"4002003202",
        "ServiceProvider":"Orange",
        "Availability":"70",
```

```
        "Security":"3",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"190"
    },
    {
        "id":"4002003202",
        "ServiceProvider":"Vodafone",
        "Availability":"90",
        "Security":"2",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"240"
    },
    {
        "id":"4002003202",
        "ServiceProvider":"Telefonica",
        "Availability":"73",
        "Security":"5",
        "Reputation":"3",
        "Reliability":"2",
        "Confidentiality":"NO",
        "Price":"80"
    },
    {
        "id":"4002003203",
        "ServiceProvider":"Orange",
        "Availability":"80",
        "Security":"3",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"590"
    },
    {
        "id":"4002003203",
        "ServiceProvider":"Vodafone",
        "Availability":"99",
        "Security":"4",
        "Reputation":"5",
        "Reliability":"4",
        "Confidentiality":"YES",
        "Price":"840"
    },
    {
        "id":"4002003203",
        "ServiceProvider":"Telefonica",
        "Availability":"73",
        "Security":"5",
        "Reputation":"4",
        "Reliability":"2",
        "Confidentiality":"NO",
        "Price":"280"
    },
    {
        "id":"4002003204",
        "ServiceProvider":"Telefonica",
        "Availability":"77",
        "Security":"3",
```

```
                "Reputation":"2",
                "Reliability":"2",
                "Confidentiality":"YES",
                "Price":"780"
            },
            {
                "id":"4002003204",
                "ServiceProvider":"Orange",
                "Availability":"86",
                "Security":"4",
                "Reputation":"2",
                "Reliability":"5",
                "Confidentiality":"NO",
                "Price":"600"
            },
            {
                "id":"4002003205",
                "ServiceProvider":"T-Mobile",
                "Availability":"70",
                "Security":"5",
                "Reputation":"4",
                "Reliability":"4",
                "Confidentiality":"YES",
                "Price":"150"
            },
            {
                "id":"4002003205",
                "ServiceProvider":"Orange",
                "Availability":"70",
                "Security":"3",
                "Reputation":"5",
                "Reliability":"4",
                "Confidentiality":"YES",
                "Price":"160"
            },
            {
                "id":"4002003205",
                "ServiceProvider":"Vodafone",
                "Availability":"90",
                "Security":"2",
                "Reputation":"5",
                "Reliability":"4",
                "Confidentiality":"YES",
                "Price":"210"
            },
            {
                "id":"4002003205",
                "ServiceProvider":"Telefonica",
                "Availability":"73",
                "Security":"5",
                "Reputation":"3",
                "Reliability":"2",
                "Confidentiality":"NO",
                "Price":"30"
            }
        ]
}
```