



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ Μ/Υ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΝΑΥΤΙΛΙΑΣ ΚΑΙ ΒΙΟΜΗΧΑΝΙΑΣ
ΤΜΗΜΑΤΟΣ ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ & ΤΕΧΝΟΛΟΓΙΑΣ
ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΤΕΧΝΟ-ΟΙΚΟΝΟΜΙΚΑ ΣΥΣΤΗΜΑΤΑ»



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

Αναστάσιος Κουρής

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ

Συμεών Παπαβασιλείου
Καθηγητής ΕΜΠ

ΙΟΥΝΙΟΣ 2020

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

Περίληψη

Καθώς πληθαίνουν οι εφαρμογές πραγματικού χρόνου για διασυνδεδεμένες κινητές συσκευές, αυξάνονται και οι απαιτήσεις για ταχύτητα υπολογισμών και αποθήκευση πληροφορίας. Εξαιτίας των περιορισμών των δυνατοτήτων των συσκευών, κρίνεται αναγκαία η εξ αποστάσεως εκτέλεση εργασιών σε ομάδες υπολογιστών που δεν έχουν αυτούς τους περιορισμούς. Ιδανικά, οι υπολογιστές που αναλαμβάνουν την εκτέλεση τέτοιων εργασιών πρέπει να βρίσκονται στα άκρα του δικτύου των κινητών συσκευών, προκειμένου να διασφαλίζονται οι απαιτήσεις ποιότητας υπηρεσίας. Ωστόσο, οι πόροι που διαθέτουν οι υπολογιστές σε τέτοιες διατάξεις δεν μπορούν να είναι απεριόριστοι, οπότε κρίνεται αναγκαία η αποδοτική διαχείριση τους. Η κλιμάκωση της εκάστοτε εφαρμογής ανάλογα με τον όγκο των αιτήσεων που επεξεργάζεται περιορίζει τους πόρους που δεσμεύονται στους απολύτως αναγκαίους, απελευθερώνοντας πλεονάζοντες πόρους. Στην παρούσα εργασία πραγματοποιείται μελέτη κλιμακωσιμότητας εφαρμογών που εκτελούνται στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes. Το Kubernetes προσφέρει δυνατότητες ελέγχου της κλιμάκωσης μια εφαρμογής μέσω του Horizontal Pod Autoscaler. Η οντότητα αυτή επιτρέπει τον έλεγχο του αριθμού των εκτελούμενων πανομοιότυπων της εφαρμογής, γνωστά ως Containers, βάσει τιμών μετρικών που η ίδια παράγει. Οι μετρικές αυτές είναι είτε τα μεγέθη των χρησιμοποιούμενων υπολογιστικών πόρων είτε εξατομικευμένες μετρικές που αφορούν ποιοτικά χαρακτηριστικά της εφαρμογής. Στο σενάριο που παρουσιάζεται, πραγματοποιείται κλιμάκωση με χρήση και των δύο τύπων μετρικών, καθώς επίσης και παραλλαγή ως προς το μέγεθος των πόρων που διατίθενται ανά Container εφαρμογής. Τα πειραματικά αποτελέσματα καταδεικνύουν τη δυνατότητα χρήσης του μηχανισμού κλιμακωσιμότητας που το Kubernetes προσφέρει, για εφαρμογές στα άκρα του δικτύου.

Λέξεις κλειδιά: Υπολογιστικό Νέφος, Μεταφόρτωση Υπολογισμών, Υπολογισμός στα Άκρα του Δικτύου, Κλιμακωσιμότητα, Kubernetes.

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

Abstract

With the increased use of real-time Internet of Things (IoT) applications, there is a growing demand for fast computation and data storage. Because of the mobile devices' current technical restrictions, computations are offloaded to computer clusters which do not show such restrictions. It is preferable that such computers reside close to and on the same network with the mobile devices so that Quality of Service (QoS) requirements are met. Such computer clusters cannot possess unlimited resources, so it is important that they are widely distributed among applications. Scaling applications according to their request load allows reserving as many resources as necessary to meet their needs, letting remaining ones to be consumed by other applications. In this study the scalability of applications executing on Edge Servers with the use of Kubernetes software is investigated. Kubernetes enables control of application scaling through its Horizontal Pod Autoscaler resource. This resource can control the number of executing Container instances of an application by using metrics collected by it. Such metrics can describe either the size of resources used the application or any other aspect of it. The latter are also called custom metrics. In the scenario presented, scaling is performed via both types of metrics. Also, a variation based on amount of resources available to each Container is tested. Experimental results prove that using Kubernetes' scaling mechanism for such applications can be effective.

Keywords: Fog Computing, Computation Offloading, Edge Computing, Scalability, Kubernetes.

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον Καθηγητή της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Ε.Μ.Π. κύριο Συμεών Παπαβασιλείου για την ευκαιρία που μου έδωσε να εκπονήσω την παρούσα διπλωματική εργασία. Επίσης, θα ήθελα να ευχαριστήσω τον Διδάκτορα Μηχανικό Δημήτρη Δεχουνιώτη για την αμέριστη στήριξη και καθοδήγηση του, καθώς και τους υποψήφιους διδάκτορες Δημήτρη Σπαθαράκη και Μάριο Αυγέρη για την συνεχή υποστήριξη και τις πολύτιμες συμβουλές τους κατά την εκπόνηση και συγγραφή της εργασίας αυτής.

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

Περιεχόμενα

Περίληψη	i
Abstract	iii
Ευχαριστίες	v
Περιεχόμενα.....	vii
1 Εισαγωγή	1
2 Βιβλιογραφία	5
2.1 Ποιότητα υπηρεσίας.....	5
2.2 Κόστος.....	6
2.3 Ενέργεια	6
2.4 Εύρος ζώνης	6
2.5 Ασφάλεια.....	7
2.6 Θεμελίωση.....	7
2.7 Αξιοπιστία, διαθεσιμότητα, επιβιωσιμότητα	7
2.8 Κινητικότητα	8
2.9 Κλιμακωσιμότητα	8
2.10 Ετερογένεια	10
2.11 Διαχείριση	10
2.12 Δυνατότητα προγραμματισμού	11
3 Αρχιτεκτονική Συστήματος.....	12
3.1 Kubernetes.....	13
3.2 Kube-prometheus	15
3.3 Η εφαρμογή.....	17
3.4 Παραλλαγές σεναρίου ως προς τον έλεγχο κλιμάκωσης	17
3.5 Προσομοίωση φορτίου.....	18
3.6 Υπολογιστικό σύστημα	19
3.7 Μετρικές εφαρμογής και παρατήρηση εφαρμογής.....	21

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

3.8	Σύνθεση εφαρμογής	22
3.8.1	Deployment	23
3.8.2	HorizontalPodAutoscaler.....	24
3.8.3	Service	25
3.8.4	Service Monitor	25
3.9	Σχετικά με τους χρόνους μετρήσεων	25
3.10	Σχετικά με το επίπεδο λειτουργίας της εφαρμογής.....	26
3.11	Σχετικά με την επεξεργαστική ισχύ των υπολογιστών	26
3.12	Τιμές στόχοι για τον ελεγκτή κλιμάκωσης	27
4	Πειραματικά Αποτελέσματα.....	29
5	Συμπεράσματα και μελλοντική εργασία	36
5.1	Συμπεράσματα	36
5.2	Μελλοντική Εργασία	37
	Παραπομπές.....	39

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

1 Εισαγωγή

Τα τελευταία χρόνια υπάρχει έντονο ενδιαφέρον για εφαρμογές πραγματικού χρόνου που απαιτούν τη συλλογή και επεξεργασία δεδομένων από κινητές συσκευές με στόχο την εξαγωγή χρήσιμων πληροφοριών. Παρόλο που η τεχνολογία φορητών συσκευών έχει προχωρήσει αισθητά δεν έχουν ξεπεραστεί ακόμη σημαντικά προβλήματα όπως αυτό της διάρκειας λειτουργίας χωρίς επαναφόρτιση μπαταρίας και οι περιορισμένοι υπολογιστικοί πόροι. Ταυτόχρονα παρουσιάζεται ολοένα και μεγαλύτερο ενδιαφέρον για υπηρεσίες που απαιτούν χαμηλούς χρόνους απόκρισης στην επεξεργασία δεδομένων, όπως υπηρεσίες υποβοήθησης οδήγησης [1]. Η μέθοδος που χρησιμοποιείται προκειμένου να επιτευχθούν οι στόχοι τέτοιων υπηρεσιών χρησιμοποιώντας φορητές συσκευές περιορισμένων δυνατοτήτων είναι η μεταφόρτωση υπολογισμών (Computation Offloading). Μέσω αυτής της μεθόδου διαδικασίες απαιτητικές σε υπολογιστικούς πόρους ανατίθενται προς επεξεργασία σε ένα εξωτερικό υπολογιστικό σύστημα με αυξημένες υπολογιστικές δυνατότητες.

Τα οφέλη από την εφαρμογή αυτής της μεθόδου είναι πολλαπλά. Ενδεικτικά:

- ταχύτητα επεξεργασίας λόγω θεωρητικά απεριόριστων υπολογιστικών πόρων,
- δεν υφίσταται το ζήτημα περιορισμένης αποθηκευμένης ενέργειας,
- υπό προϋποθέσεις χαμηλός χρόνος απόκρισης δεδομένου ότι ο χρόνος υπολογισμού μειώνεται.

Καθώς ο όγκος των υπολογισμών και των δεδομένων αυξάνεται κρίνεται απαραίτητη μια τέτοια μετατόπιση. Παρόλο που ο Υπολογισμός Σύννεφου (Cloud Computing) είναι μια προφανής λύση δεν ενδείκνυται για κάθε εφαρμογή. Για υπηρεσίες που βασίζονται στην τοποθεσία και είναι ευαίσθητες ως προς τον χρόνο απόκρισης είναι αδύνατο να βασιστούν σε μεταφόρτωση υπολογισμών στο απομακρυσμένο Cloud. Ο Υπολογισμός Νέφους (Fog Computing) έχει προταθεί για να αντιμετωπιστούν τέτοιοι περιορισμοί. Ο υπολογισμός νέφους γεφυρώνει το χάσμα μεταξύ φορητών συσκευών και του Cloud, αφού επιτρέπει την εκτέλεση υπολογισμών, την αποθήκευση δεδομένων, την εξυπηρέτηση αναγκών δικτύωσης, τη διαχείριση δεδομένων και τη λήψη αποφάσεων. Όταν οι υπολογισμοί γίνονται στο χώρο που κινούνται οι φορητές συσκευές σε υπολογιστικά συστήματα που βρίσκονται στο ίδιο δίκτυο με αυτές τότε

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

έχουμε υπολογισμό στα άκρα του δικτύου (Edge Computing) [2]. Αυτό είναι και το πρώτο επίπεδο υπολογισμού νέφους κατά τη μεταφόρτωση υπολογισμών από τις κινητές συσκευές προς το Cloud, καθώς εντοπίζεται στα άκρα του δικτύου κοντά στις κινητές συσκευές.

Μια μορφή υπολογισμού στα άκρα του δικτύου είναι και ο Κινητός Υπολογισμός Νέφους (Mobile Cloud Computing) ο οποίος εισάγει στα άκρα του δικτύου κινητών συσκευών. τον υπολογισμό σύννεφου, με όλα τα χαρακτηριστικά και τα προτερήματα του. Σε αυτή τη μορφή μέσω κατάλληλου λογισμικού οι υπολογιστικοί πόροι μια συστοιχίας υπολογιστών που βρίσκονται στα άκρα του δικτύου διαμοιράζονται αφαιρετικά και διατίθενται για την εκτέλεση διαδικασιών που απαιτούνται από κινητές συσκευές [3]. Μέσω της μεταφόρτωσης υπολογισμών από τις κινητές συσκευές στον υπολογισμό νέφους αντιμετωπίζονται οι έμφυτοι περιορισμοί των κινητών συσκευών που προαναφέρθηκαν ενώ εφαρμόζεται με τέτοιο τρόπο ώστε να εκπληρώνονται περιορισμοί των υπηρεσιών όπως ανώτατο όριο χρόνου εκτέλεσης μιας διαδικασίας.

Όταν οι υπολογιστικοί πόροι που διατίθενται για ένα είδος υπηρεσίας είναι στατικοί είναι προφανές ότι καθώς αυξάνεται ο ρυθμός αιτήσεων εκτέλεσης διαδικασιών ο μέσος χρόνος απόκρισης αυξάνεται, ενδεχομένως και κάποιες αιτήσεις να απορρίπτονται. Αντίστροφα, οι πόροι πλεονάζουν σε περίπτωση χαμηλού ρυθμού αιτήσεων ενώ θα μπορούσαν να είχαν διαμοιραστεί σε άλλες υπηρεσίες. Η δυναμική διάθεση υπολογιστικών πόρων σε υπηρεσίες με κατάλληλο τρόπο μπορεί να επιτύχει να πληρούνται οι περιορισμοί ενώ ταυτόχρονα να μη σπαταλούνται πόροι [4].

Στόχος αυτής της εργασίας είναι η μελέτη της κλιμακωσιμότητας εφαρμογών (Application Scaling) στα άκρα του δικτύου ως μέθοδος για το δυναμικό διαμοιρασμό υπολογιστικών πόρων. Η εφαρμογή αυτή αφορά επεξεργασία αιτήσεων από χρήστες κινητών συσκευών που για λόγους εξοικονόμησης ενέργειας και περιορισμών καθυστέρησης επεξεργασίας επιλέγουν να πραγματοποιήσουν μεταφόρτωση υπολογισμών. Η εφαρμογή εκτελείται σε σύστημα Kubernetes το οποίο αναλαμβάνει την διαχείριση εφαρμογών που εκτελούνται περιορισμένες στο δικό τους περιβάλλον με δυνατότητα ελέγχου των υπολογιστικών πόρων του συστήματος που αυτές χρησιμοποιούν [5]. Η τεχνολογία που επιτρέπει την εκτέλεση εφαρμογών σε

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

περιοριστικό ως προς τους πόρους περιβάλλον ονομάζεται Container και το κυρίαρχο λογισμικό που την παρέχει είναι το Docker [6].

Το Kubernetes προσφέρει τη δυνατότητα για ταυτόχρονη εκτέλεση πολλαπλών Containers της εφαρμογής που εξυπηρετούν την ίδια υπηρεσία, το καθένα εκ των οποίων διαθέτει συγκεκριμένους στατικούς υπολογιστικούς πόρους. Μέσω της αυξομείωσης του αριθμού των Containers αυξομειώνονται αντίστοιχα και οι πόροι που διατίθενται συνολικά στην υπηρεσία. Η απόφαση για κλιμάκωση ή αποκλιμάκωση της εφαρμογής στηρίζεται στις τιμές συγκεκριμένων μετρικών που συνδέονται με την ποιότητα της υπηρεσίας, όπως ο χρόνος απόκρισης για μία αίτηση. Ο ελεγκτής που προσφέρει το Kubernetes, λαμβάνοντας υπόψιν αυτές τις μετρικές, αποφασίζει εάν απαιτείται κάποια ενέργεια κλιμάκωσης και εφόσον χρειαστεί την εκτελεί αυξάνοντας ή μειώνοντας κατάλληλα τον αριθμό των Containers της υπηρεσίας.

Στο πείραμα που παρουσιάζεται εκτελείται ένα σενάριο προσομοίωσης λήψης αιτήσεων μεταβαλλόμενου ρυθμού με στόχο να μελετηθεί η συμπεριφορά του συστήματος σε διαφορετικά επίπεδα ζήτησης. Τα συμπεράσματα εξάγονται μέσω παρατήρησης της μέσου χρόνου απόκρισης της εφαρμογής και των χρησιμοποιούμενων πόρων του υπολογιστικού συστήματος. Προκειμένου να εξαχθούν περαιτέρω συμπεράσματα, το σενάριο εκτελείται με παραλλαγές ως προς τη ρύθμιση κλιμάκωσης του συστήματος εξυπηρέτησης αιτήσεων.

Στο επόμενο κεφάλαιο παρουσιάζονται προβλήματα της περιοχής που αφορούν τη μεταφόρτωση υπολογισμών και κάποιες λύσεις που έχουν προταθεί στη βιβλιογραφία. Αυτό γίνεται μέσω μιας πρότυπης κατηγοριοποίησης των προβλημάτων που αφορά τους στόχους και τα χαρακτηριστικά που αυτά παρουσιάζουν.

Στη συνέχεια στο τρίτο κεφάλαιο περιγράφεται η αρχιτεκτονική του συστήματος αυτής της εργασίας. Παρουσιάζονται οι ορισμοί και τα σημαντικότερα χαρακτηριστικά των τεχνολογιών που χρησιμοποιούνται. Παρουσιάζονται επίσης η εφαρμογή που χρησιμοποιείται για τη μελέτη κλιμακωσιμότητας και ο τρόπος που προσομοιώνεται το φορτίο αιτήσεων προς αυτή. Ακόμη, γίνεται αναφορά στις μετρικές που συλλέγονται κατά την εκτέλεση της εφαρμογής και πως αυτές εκτίθενται στο σύστημα Kubernetes. Τέλος περιγράφεται η διάταξη ελέγχου κλιμάκωσης αυτής της εργασίας.

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

Στο τέταρτο κεφάλαιο παρουσιάζονται τα αποτελέσματα από τα πειράματα που εκτελέστηκαν στη διάταξη του προβλήματος.

Στο πέμπτο και τελευταίο κεφάλαιο παρουσιάζονται συμπεράσματα από την εργασία και γίνονται προτάσεις για μελλοντική εργασία.

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

2 Βιβλιογραφία

Η έρευνα γύρω από τον υπολογισμό νέφους για τη μεταφόρτωση υπολογισμών από κινητές συσκευές είναι σημαντική και πολύπλευρη. Σε αυτό το κεφάλαιο παρουσιάζονται κάποιες από τα προβλήματα που έχουν συζητηθεί και τις λύσεις που έχουν προταθεί για διάφορες υπηρεσίες. Όπως έδειξαν οι Yousefrou et al.[7], αυτές οι λύσεις έχουν κάποιους στόχους και χαρακτηριστικά που είναι κοινά και με βάση αυτά μπορεί να γίνει μια κατηγοριοποίηση των προβλημάτων της περιοχής. Σημειώνεται ότι κάθε λύση που παρουσιάζεται μπορεί να έχει πολλαπλούς στόχους και χαρακτηριστικά. Στη συνέχεια παρουσιάζονται αυτοί οι στόχοι και χαρακτηριστικά καθώς και δημοσιευμένες εργασίες που τα διαθέτουν.

2.1 Ποιότητα υπηρεσίας

Στόχος των λύσεων σε αυτή την κατηγορία είναι η βελτίωση της ποιότητας της υπηρεσίας (QoS) όπως γίνεται αντιληπτή κατά τη χρήση της. Αυτό μπορεί να επιτυγχάνεται είτε μέσω μείωσης ή της διαχείρισης του χρόνου εξυπηρέτησης είτε αυξάνοντας το ποσοστό επιτυχίας.

Οι Zhao et al. [8] μελέτησαν το πρόβλημα της υποβάθμισης της ποιότητας υπηρεσίας στα άκρα του δικτύου (Edge) λόγω περιορισμένων υπολογιστικών πόρων. Η λύση που προτείνουν είναι η συντονισμένη χρήση ενός ετερογενούς υπολογισμού νέφους το οποίο αποτελείται από το Edge Cloud και ένα απομακρυσμένο Cloud. Έτσι μέσω ενός αλγορίθμου δρομολόγησης στο ετερογενές αυτό Cloud, όταν ο όγκος αιτήσεων ανεβαίνει σημαντικά οι διαδικασίες που έχουν χαλαρούς περιορισμούς καθυστέρησης εξυπηρέτησης δρομολογούνται προς εκτέλεση στο απομακρυσμένο σύστημα. Αντίθετα αυτές που είναι πιο ευαίσθητες ως προς την καθυστέρηση εξυπηρετούνται από το σύστημα στα άκρα του δικτύου.

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

2.2 Κόστος

Πολλά προβλήματα έχουν απασχοληθεί με το πρόβλημα του κόστους. Αυτό μπορεί να είναι είτε το κόστος λειτουργίας ενός συστήματος είτε το κόστος κεφαλαίου που σχετίζεται με αυτό.

Οι Mehta et al. [9] μελετούν πως επιδρά στη μείωση του λειτουργικού κόστους η τοποθέτηση υπολογιστικών κέντρων κοντά άκρα του δικτύου. Αποδεικνύεται ότι η ύπαρξη υπολογιστικών πόρων στα άκρα του δικτύου μειώνει το κόστος για εφαρμογές που έχουν υψηλές απαιτήσεις σε χρήση εύρους ζώνης αφού εξυπηρετούνται τοπικά. Ακόμη απαντούν σε ερωτήματα όπως που πρέπει να τοποθετούνται τα υπολογιστικά κέντρα και με τι πόρους ώστε να ελαχιστοποιείται το κόστος ενώ επιτυγχάνονται και στόχοι απόδοσης των υπηρεσιών.

2.3 Ενέργεια

Οι συγγραφείς αναλύουν την κατανάλωση ενέργειας ή ισχύος από τις λύσεις που προτείνουν.

Οι Jalali et al. [10] μελετούν το πρόβλημα της κατανάλωσης ενέργειας από τα υπολογιστικά κέντρα σε εφαρμογές υπολογισμού νέφους. Αποδεικνύουν ότι συγκεκριμένες εφαρμογές είναι πιο αποδοτικές ως προς την ενέργεια που καταναλώνουν όταν οι υπολογισμοί πραγματοποιούνται σε νάνο-υπολογιστικά κέντρα από ό,τι σε μεγάλα.

2.4 Εύρος ζώνης

Σε αυτές τις μελέτες συζητούνται και προτείνονται αλγόριθμοι που βοηθούν στην αποδοτική εκμετάλλευση του διαθέσιμου εύρους ζώνης ή στη μείωση του ρυθμού εξυπηρέτησης.

Για παράδειγμα οι Hao et al. [11] χρησιμοποιούν υπολογισμό στα άκρα του δικτύου για να περιοριστεί η εκτεταμένη χρήση εύρους ζώνης από εφαρμογές γραφείου που αποθηκεύουν τα δεδομένα σε απομακρυσμένο αποθηκευτικό χώρο. Ύστερα από μετρήσεις που έγιναν απέδειξαν ότι μετά από μικρές αλλαγές σε κάποιο έγγραφο αυτό αποστέλλεται ολόκληρο σπαταλώντας έτσι εύρος ζώνης. Σχεδιάζουν μια προσέγγιση

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

σταδιακής αποθήκευσης εγγράφου με χρήση υπολογισμού στα άκρα του δικτύου, που αποδεικνύεται ότι χρησιμοποιεί αποδοτικότερα το διαθέσιμο εύρος ζώνης.

2.5 Ασφάλεια

Οι μελέτες σε αυτή την περιοχή ασχολούνται με θέματα ασφαλείας και ιδιωτικότητας σε συστήματα υπολογισμού νέφους. Πιο συγκεκριμένα ασχολούνται με τρωτά σημεία, μηχανισμούς προστασίας, θέματα ιδιωτικότητας και πρωτόκολλα ασφαλείας τέτοιων συστημάτων.

Οι Singh et al. [12] προτείνουν έναν αλγόριθμο δρομολόγησης της μεταφόρτωσης υπολογισμών μεταξύ δύο υπολογιστικών κέντρων, το ένα στα άκρα του δικτύου και το άλλο απομακρυσμένο, που ονομάζεται RT-SANE. Ο αλγόριθμος αυτός λαμβάνει υπόψιν του τόσο τις προθεσμίες εκτέλεσης των υπολογισμών προκειμένου να επιτυγχάνεται ο στόχος της ποιότητας υπηρεσίας όσο και περιορισμούς ασφαλείας. Πιο συγκεκριμένα υπολογισμοί για εφαρμογές που έχουν περιορισμούς ιδιωτικότητας (ευαίσθητα προσωπικά δεδομένα), δρομολογούνται προς εκτέλεση στο τοπικό κέντρο υπολογισμών ενώ άλλοι μπορούν να δρομολογηθούν είτε στο ένα είτε στο άλλο.

2.6 Θεμελίωση

Σε εργασίες αυτής της κατηγορίας παρουσιάζονται οι θεμελιώδεις συνιστώσες και έννοιες της περιοχής.

Ως παράδειγμα αυτής της κατηγορίας οι Ahmed et al. [13] παρουσιάζουν μια εκτενή μελέτη για τον υπολογισμό στα άκρα του δικτύου, παραθέτοντας τους παράγοντες που ευνόησαν την ανάπτυξή του, τα πλεονεκτήματά του, απαιτήσεις για επιτυχή εφαρμογή, κατάλληλες εφαρμογές για χρήση και προκλήσεις. Μερικές από τις εφαρμογές που προτείνουν ως κατάλληλες περιπτώσεις για υπολογισμό στα άκρα του δικτύου είναι παιχνιδιών, επεξεργασίας εικόνας πραγματικού χρόνου, έξυπνων ηλεκτρικών δικτύων (Smart Grid) και έξυπνων μεταφορών (Smart Transportation).

2.7 Αξιοπιστία, διαθεσιμότητα, επιβιωσιμότητα

Προτάσεις σε αυτή την κατηγορία λύσεων βελτιώνουν την αξιοπιστία, τη διαθεσιμότητα και της ικανότητα επιβίωσης των συστημάτων υπολογισμού νέφους ή

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

παρουσιάζουν λύσεις που χρησιμοποιούν συστήματα υπολογισμού νέφους για τη βελτίωση αυτών των τριών ιδιοτήτων σε περίπτωση δικτυακών σφαλμάτων ή σφαλμάτων κόμβων.

Οι Zhu et al. [14] προτείνουν ένα μοντέλο εξισορρόπησης του κόστους από χρήση εύρους ζώνης με τη διαθεσιμότητα υπηρεσίας σε περιβάλλον υπολογισμού στα άκρα του δικτύου για κινητές συσκευές (Mobile Edge Computing). Συγκεκριμένα μελετούν την τοποθέτηση εικονικών υπολογιστών (Virtual Machines) είτε σε έναν πραγματικό υπολογιστή (Physical Host), είτε σε περισσότερους. Στην πρώτη περίπτωση το κόστος από χρήση εύρους ζώνης δικτύου είναι μηδενικό κατά την επικοινωνία των Virtual Machines αλλά μειώνεται η διαθεσιμότητα σε περίπτωση σφάλματος του πραγματικού υπολογιστή. Με το μοντέλο που προτείνουν είναι δυνατό για διάφορες υπηρεσίες να βρίσκονται τα κατάλληλα «σημεία ισορροπίας» που διατηρούν σημαντικό επίπεδο διαθεσιμότητας χωρίς να αυξάνεται ιδιαίτερα πολύ η χρήση του δικτύου και κατ' επέκταση και το κόστος.

2.8 Κινητικότητα

Οι λύσεις σε αυτή την κατηγορία πραγματεύονται τη δυνατότητα κινητικότητας των συσκευών του συστήματος υπολογισμού νέφους.

Οι Bittencourt et al. [15] μελετούν την επίδραση της κινητικότητας των κινητών συσκευών στον προγραμματισμό εκτέλεσης υπολογισμού νέφους και προτείνουν τρεις πολιτικές προγραμματισμού με στόχο τη βελτίωση της εκτέλεσης εφαρμογών με διάφορα χαρακτηριστικά.

2.9 Κλιμακωσιμότητα

Μελέτες σε αυτή την κατηγορία ασχολούνται με το πρόβλημα της δυνατότητας αποδοτικής εξυπηρέτησης κινητών συσκευών όταν ο αριθμός τους και ο όγκος των αιτήσεών τους αυξάνεται.

Στην εργασία [16] προτάθηκε ένα κυβερνο-φυσικό σύστημα για την έγκαιρη ανίχνευση πυρκαγιών σε δασικές εκτάσεις. Μη επανδρωμένα ιπτάμενα οχήματα συλλέγουν φωτογραφίες, οι οποίες αποστέλλονται για επεξεργασία σε μια υπολογιστική υποδομή στα άκρα του δικτύου. Για την διαχείριση των πόρων της υποδομής, η εφαρμογή της

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

επεξεργασίας εικόνας έχει μοντελοποιηθεί με γραμμικά συστήματα και για την εξυπηρέτηση του μεταβαλλόμενου φορτίου σχεδιάστηκαν ελεγκτές ανατροφοδότησης κατάστασης.

Οι Gupta et al. [17] προτείνουν ένα ενδιάμεσο λογισμικό σε επίπεδο υπηρεσίας, το SDFog, που έχει σαν στόχο να κατανέμει υπηρεσίες σε κόμβους δικτύου Νέφους για κλιμακωσιμότητα. Το λογισμικό αυτό με τη βοήθεια της τεχνολογίας δικτύου μέσω λογισμικού (Software Defined Network) πραγματοποιεί ενορχήστρωση που επιτυγχάνει τον περιορισμό της ποιότητας υπηρεσίας με το να κατανέμει φορτίο μεταξύ των υπηρεσιών αυτών. Η αρχιτεκτονική της λύσης που προτείνουν αποτελείται από δύο οντότητες το ενδιάμεσο λογισμικό σε επίπεδο υπηρεσίας και ένα και ένα κατανεμημένο σύστημα ενορχήστρωσης υπηρεσιών. Το πρώτο επιτρέπει όλους τους κόμβους του δικτύου αφαιρετικά να θεωρούνται πλατφόρμες φιλοξενίας υπηρεσιών ενώ το δεύτερο πραγματοποιεί την ενορχήστρωση μεταξύ των κόμβων. Το γεγονός ότι το σύστημα είναι κατανεμημένο στους κόμβους επιτρέπει την κλιμακωσιμότητά του συστήματος συνολικά.

Οι Wang et al. [18] προτείνουν ένα λογισμικό διαχείρισης πόρων στα άκρα του δικτύου που έχει δυνατότητες παροχής πόρων και αυτόματης κλιμακωσιμότητας. Το λογισμικό αυτό ενοποιεί έναν κόμβο στα άκρα του δικτύου με έναν κόμβο Cloud ώστε μέσω της κλιμακωσιμότητας να επιτυγχάνεται δυναμική κατανομή πόρων. Επίσης είναι κατάλληλο για περιβάλλοντα με περιορισμένους διαθέσιμους πόρους. Οι συγγραφείς ελέγχουν την εγκυρότητα της λύσης μέσω μιας υπηρεσίας πραγματικού χρόνου με βάση την τοποθεσία και με περιορισμούς χρόνου απόκρισης (παιχνίδι Pokemon Go).

Οι El Kafhali and Salah [19] προκειμένου να αποδώσουν ελαστικότητα και κλιμακωσιμότητα στα δίκτυα κινητών συσκευών προτείνουν ένα αναλυτικό μοντέλο βασισμένο στη θεωρία ουρών αναμονής που υπολογίζει τους απαιτούμενους πόρους για ένα δεδομένο όγκο φορτίου που προέρχεται από κινητές συσκευές προκειμένου να επιτευχθούν οι στόχοι ποιότητας υπηρεσίας. Οι κόμβοι του δικτύου που εξυπηρετούν αυτό το φορτίο κλιμακώνονται δυναμικά ανάλογα με το φορτίο που δέχονται σύμφωνα με τους υπολογισμούς του μοντέλου. Ιδανικά, όταν το φορτίο είναι υψηλό διατίθενται περισσότεροι κόμβοι στα άκρα του δικτύου ενώ όταν αυτό είναι χαμηλό κάποιοι κόμβοι αποδεσμεύονται. Αποδεικνύουν ότι μέσω δυναμικής κατανομής πόρων με τη

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

βοήθεια του μοντέλου που προτείνουν επιτυγχάνεται αποδοτικότερη διαχείριση των διαθέσιμων πόρων στα άκρα του δικτύου ενώ διασφαλίζεται η ποιότητα υπηρεσίας.

Οι Avgeris et al. [20] προτείνουν έναν μηχανισμό δύο επιπέδων για την αποδοχή αιτήσεων και την δυναμική κατανομή πόρων για μεταφόρτωση υπολογισμών των κινητών συσκευών. Ο μηχανισμός αυτός είναι ικανός να μεταβάλλει δυναμικά τους πόρους που παρέχονται για υπολογισμούς ανάλογα με τον όγκο των αιτήσεων, κλιμακώνοντας κάθετα (Vertical Scaling) και οριζόντια (Horizontal Scaling). Στόχος αυτού του μηχανισμού είναι να επιτυγχάνονται οι περιορισμοί ποιότητας υπηρεσίας ενώ ταυτόχρονα γίνεται αποδοτική χρήση των διαθέσιμων υπολογιστικών πόρων.

2.10 Ετερογένεια

Λύσεις και αλγόριθμοι που προτείνονται σε αυτή την κατηγορία παρουσιάζουν διατάξεις που χρησιμοποιούν ετερογενή υπολογιστικά συστήματα ως προς την υπολογιστική ισχύ, τον αποθηκευτικό χώρο και τις δικτυακές ικανότητες.

Οι Cirani et al. [21] συζητούν το πρόβλημα της δικτύωσης διαφόρων οντοτήτων που απαρτίζουν συστήματα μεταφόρτωσης υπολογισμών στο νέφος. Αυτές οι οντότητες περιλαμβάνουν τόσο τις κινητές συσκευές όσο και τους κόμβους μεταφόρτωσης υπολογισμών και τους κόμβους δικτύωσης αυτών. Οι ετερογένεια προκύπτει από το ότι χρησιμοποιούνται διάφοροι εξοπλισμοί (Hardware) και λογισμικό (Software) διαφορετικών υπολογιστικών προδιαγραφών. Προτείνουν την ενσωμάτωση ενδιάμεσων υπολογιστικών κόμβων που ονομάζουν «IoT Hub» ανάμεσα σε ετερογενή δίκτυα υπολογισμών για τη διασύνδεσή τους.

2.11 Διαχείριση

Έρευνες σε αυτή την κατηγορία ασχολούνται με ζητήματα διαχείρισης, παρακολούθησης και προτείνουν ομοσπονδιακά σχήματα (federation schemes) για τις κινητές συσκευές σε συστήματα νέφους.

Οι Wen et al. [22] παρουσιάζουν μια σειρά προβλημάτων που υπάρχουν στην διαχείριση και λειτουργία συστημάτων υπολογισμών νέφους που προκύπτουν από την ετερογένεια ως προς τις απαιτήσεις και τη συμπεριφορά πολλαπλών εξυπηρετούμενων

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

εφαρμογών στο ίδιο σύστημα. Προτείνουν επίσης έναν κατανεμημένο γενετικό αλγόριθμο που επιχειρεί να δώσει μια λύση σε προσομοίωση ενός τέτοιου συστήματος.

2.12 Δυνατότητα προγραμματισμού

Σε αυτή την κατηγορία ανήκουν λύσεις που προτείνουν κάποια προγραμματιστική γλώσσα, προγραμματιστικό πλαίσιο (framework) ή μοντελοποίηση δεδομένων για συστήματα υπολογισμού νέφους.

Οι Lingen et al. [23] δημιούργησαν μία μοντελοποίηση YANG για κόμβους συστημάτων υπολογισμού νέφους που επιτρέπει ένα κοινό πλαίσιο διαχείρισής τους. Οι συγγραφείς επίσης αποσυνθέτουν την πολυπλοκότητα στη μοντελοποίηση των υπηρεσιών για επαναχρησιμοποίηση και σύνθεση υψηλότερου επιπέδου υπηρεσιών για καλύτερη διαχείριση των παρεχόμενων υπηρεσιών.

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

3 Αρχιτεκτονική Συστήματος

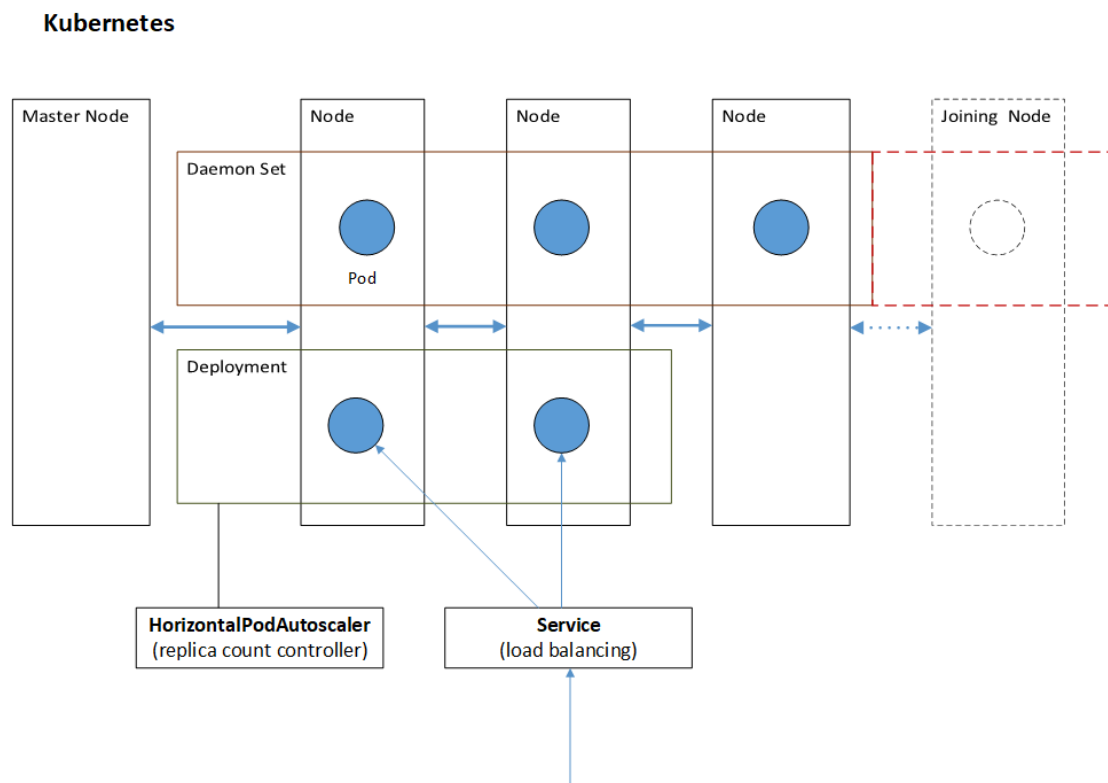
Αυτή η εργασία μελετά το πρόβλημα της κλιμακωσιμότητας εφαρμογής που εκτελείται σε υπολογιστικό σύστημα στα άκρα του δικτύου υλοποιώντας μεταφόρτωση υπολογισμών. Στο σενάριο του προβλήματος θεωρείται ότι υπάρχει ένας αριθμός κινητών συσκευών οι οποίες αποστέλλουν αιτήσεις προς επεξεργασία στο υπολογιστικό σύστημα που βρίσκεται στα άκρα του δικτύου. Στόχος είναι η εξοικονόμηση ενέργειας στις κινητές συσκευές καθώς και η πρόσβαση σε σημαντικής κλίμακας υπολογιστικούς πόρους. Βασικός περιορισμός λειτουργίας του συστήματος είναι ο χρόνος απόκρισης της εφαρμογής να μην υπερβαίνει ένα όριο ώστε να εξασφαλίζεται η ποιότητα υπηρεσίας που απολαμβάνουν οι χρήστες. Γι' αυτό το λόγο το σύστημα πρέπει να κατανέμει κατάλληλα υπολογιστικούς πόρους στην εφαρμογή εξυπηρέτησης αιτήσεων. Ταυτόχρονα πλεονάζοντες πόροι πρέπει να απελευθερώνονται για να χρησιμοποιηθούν από άλλες εφαρμογές.

Το υπολογιστικό σύστημα στα άκρα του δικτύου χρησιμοποιεί το λογισμικό Kubernetes [5] για τη εκτέλεση της εφαρμογής και μέσω αυτού εφαρμόζονται πολιτικές κλιμάκωσης. Η απόφαση για κλιμάκωση ή αποκλιμάκωση λαμβάνεται μέσω σύγκρισης παρατηρούμενων τιμών και τιμών στόχων για μετρικές της υπηρεσίας. Η παρακολούθηση, η καταγραφή των τιμών αυτών και η έκθεσή τους στο Kubernetes γίνεται μέσω μιας σύνθεσης λογισμικού το kube-prometheus [24] το οποίο στον πυρήνα του χρησιμοποιεί το λογισμικό Prometheus [25] για την καταγραφή των μετρικών.

Στη συνέχεια περιγράφονται οι τεχνολογίες που χρησιμοποιούνται στη διάταξη και παρουσιάζεται η αρχιτεκτονική του συστήματος.

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

3.1 Kubernetes



Εικόνα 1: Γραφική παράσταση των βασικότερων οντοτήτων ενός Kubernetes Cluster.

Το Kubernetes (ή εν συντομία K8s) [5] είναι μία μεταφερόμενη (portable), επεκτάσιμη πλατφόρμα ανοιχτού κώδικα για τη διαχείριση περιορισμένων (containerized) διαδικασιών και υπηρεσιών που υποστηρίζει τόσο εκτέλεση κατ' εντολή όσο και αυτοματοποιημένη. Τα Containers έχουν το δικό τους σύστημα αρχείων, Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ) και μνήμη μεταξύ άλλων ενώ μπορούν να μοιράζονται μαζί με άλλες εφαρμογές το λειτουργικό σύστημα του υπολογιστή που φιλοξενούνται. Μοιάζουν πολύ με Virtual Machines, είναι «ελαφρά» κατά τη χρήση τους και μπορούν να μεταφερθούν και να εκτελεστούν σε οποιοδήποτε σύστημα μπορεί να φιλοξενήσει Containers. Η κυρίαρχη τεχνολογία για Containers είναι το Docker [6].

Το K8s εκτελείται κατανεμημένα σε συστοιχία υπολογιστών (πραγματικών ή εικονικών) που αποτελούν το K8s Cluster. Είναι ένα σύστημα που παρέχει ένα πλαίσιο για την ανθεκτική (resilient) εκτέλεση εφαρμογών υπό τη μορφή Container. Αναλαμβάνει μεταξύ άλλων την κλιμάκωση των εφαρμογών, την αντιμετώπιση

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

σφαλμάτων κατά την εκτέλεση τους και μοντέλα ανάπτυξης τους (deployment patterns).

Το K8s παρέχει τη δυνατότητα χρήσης των Kubernetes Objects. Αυτά είναι λογικές οντότητες στο σύστημά του που αποτελούν την κατάσταση του Cluster. Συγκεκριμένα μπορούν να περιγράψουν:

- Τι Containerized εφαρμογές εκτελούνται και που.
- Οι πόροι που είναι στη διάθεση αυτών των εφαρμογών.
- Οι πολιτικές (Policies) που περιγράφουν τη συμπεριφορά που θα ακολουθηθεί σε κάποιο γεγονός που αφορά τις εφαρμογές (επανεκκίνηση, σφάλματα, αναβάθμιση).

Οι κόμβοι (Nodes) που αποτελούν ένα K8s Cluster αποτελούνται από απλούς κόμβους που αναλαμβάνουν την φιλοξενία Pods και κόμβους με διαχειριστικό ρόλο που ονομάζονται Master Nodes. Τα Pods μπορούν να περιέχουν ένα ή περισσότερα Containers και εικονικό αποθηκευτικό χώρο για αυτά και αποτελούν τη θεμελιώδη μονάδα για την εκτέλεση μιας διαδικασίας. Κάθε Node που συμμετέχει στο Cluster παρέχει τους υπολογιστικούς του πόρους σε αυτό. Αυτό σημαίνει πως το K8s μπορεί να προγραμματίσει την εκτέλεση Pods σε αυτό. Μέσω της φιλοξενίας ενός Pod ένα μέρος των πόρων που αυτό διαθέτει δεσμεύεται και αφαιρείται από τους διαθέσιμους του Node.

Στο API του K8s ορίζονται μια σειρά από οντότητες που δίνουν τη δυνατότητα για έλεγχο των εκτελούμενων Pods μέσω κατάλληλου ελεγκτή. Μια τέτοια οντότητα είναι το Deployment. Μέσω της χρήσης αυτής της οντότητας μπορεί να οριστεί η εκτέλεση ενός Pod με βαθμό επανάληψης (replica count). Ο βαθμός επανάληψης δηλώνει πόσα αντίγραφα του ίδιου Pod πρέπει να εκτελούνται ανά πάσα στιγμή στο Cluster. Αντίστοιχα υπάρχει το DaemonSet το οποίο επιτρέπει την εκτέλεση ενός αντιγράφου Pod σε καθένα από μια σειρά Nodes του Cluster. Αν κάποιο Node προστίθεται ή αφαιρείται σε αυτή την ομάδα ο ελεγκτής εκκινεί ή καταργεί ένα αντίγραφο του Pod αντίστοιχα για αυτό.

Μια άλλη σημαντική οντότητα που μπορεί να χρησιμοποιηθεί για την έκθεση/δημοσίευση μιας υπηρεσίας που εκτελείται ως μια ομάδα από Pods είναι η

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

οντότητα Service. Μέσω ενός Service παρέχεται η δυνατότητα εξεύρεσης μιας υπηρεσίας και ο καταμερισμός φορτίου στην ομάδα των Pods αυτής. Ακόμη προσφέρει δυνατότητες δρομολόγησης αιτήσεων με επιπλέον κριτήρια όπως π.χ. δρομολόγηση στο Pod το οποίο φιλοξενείται στο Node από το οποίο προήλθε η αίτηση.

Το K8s δίνει τη δυνατότητα μέσω της Horizontal Pod Autoscaler (HPA) οντότητας να πραγματοποιείται αυτόματη κλιμάκωση μιας εφαρμογής. Συγκεκριμένα αυξάνεται ή μειώνεται αυτόματα ο βαθμός επανάληψης των αντιγράφων ενός Pod που εκτελούνται για ένα Deployment. Η μεταβολή αυτή εξαρτάται από την παρατηρούμενη χρήση υπολογιστικών πόρων ή από άλλες τιμές μετρικών και από τη συμπεριφορά που έχει οριστεί. Για παράδειγμα μπορεί να οριστεί ο ελάχιστος και μέγιστος αριθμός Pods και μια τιμή στόχος. Ο ελεγκτής του Cluster περιοδικά ελέγχει την παρατηρούμενη τιμή και προσαρμόζει το αριθμό των Pods ανάλογα σύμφωνα με τον εξής τύπο:

$$P_D = \text{CEIL}[P_C * (M_C/M_T)]$$

Όπου P_D είναι ο επιθυμητός αριθμός Pods, P_C είναι ο αριθμός Pods αυτή τη στιγμή, M_C είναι η τωρινή τιμή της μετρικής που έχει οριστεί να παρατηρείται και M_T είναι η τιμή στόχος αυτής της μετρικής.

Οι μετρικές που συμβουλεύεται κατά το βρόχο ελέγχου ο HPA παρέχονται από τις διάφορες διεπαφές μετρικών (metrics APIs) που υποστηρίζονται στο Kubernetes. Δύο από αυτά τα APIs που χρησιμοποιούνται σε αυτή την εργασία είναι:

- Το metrics.k8s.io API που παρέχει μετρικές που αφορούν τους πόρους του συστήματος, όπως παραδείγματος χάρη, τη χρήση κεντρικής μονάδας επεξεργασίας από ένα Container.
- Το custom.metrics.k8s.io API που παρέχει κάθε είδους εξατομικευμένες μετρικές που δεν αφορούν την χρήση υπολογιστικών πόρων του συστήματος.

Το kube-prometheus [24] που παρουσιάζεται στη συνέχεια και εγκαθίσταται επιπλέον στο K8s Cluster του προβλήματος παρέχει υλοποιήσεις αυτών των δύο APIs.

3.2 Kube-prometheus

Το kube-prometheus [24] είναι μια συλλογή από επιμέρους λογισμικά μαζί με διαδικασίες που βοηθούν στην εύκολη εγκατάσταση και λειτουργία ενός συστήματος

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

παρακολούθησης ενός Kubernetes Cluster βασισμένο στο Prometheus. Στη συνέχεια παρουσιάζονται τα σημαντικότερα μέρη αυτού.

Το Prometheus [25] είναι ένα λογισμικό που βοηθά στην παρακολούθηση συστημάτων μέσω της αποθήκευσης απλών τύπων μετρικών (όπως απλών μετρητών) ανά τακτά χρονικά διαστήματα. Έτσι μπορεί να ρυθμιστεί με συγκεκριμένα συστήματα στόχους από τα οποία με προκαθορισμένη συχνότητα συλλέγει εσωτερικές μετρικές που αυτά διατηρούν. Για παράδειγμα ένας εξυπηρετητής HTTP μπορεί να διατηρεί έναν μετρητή αριθμού αιτήσεων που δέχεται τον οποίον αυξάνει κατά μία μονάδα με κάθε νέα αίτηση. Αυτές τις μετρικές το Prometheus τις αποθηκεύει στην βάση δεδομένων του μαζί με τη χρονική στιγμή λήψης. Αυτά τα δεδομένα χρονοσειράς στη συνέχεια μπορούν να εξαχθούν και να χρησιμοποιηθούν για την κατανόηση λειτουργίας των εφαρμογών που τα παρέχουν. Για παράδειγμα για τον εξυπηρετητή που προαναφέρθηκε μπορεί να γίνει γραφική αναπαράσταση του ρυθμού αφίξεων αιτήσεων.

Το Prometheus-operator [26] είναι λογισμικό που παρέχει έναν εύκολο τρόπο εγκατάστασης και διαχείρισης μιας συστοιχίας εξυπηρετητών Prometheus σε ένα K8s Cluster υπό τη μορφή Container. Ακόμη παρέχει τη δυνατότητα αυτοματοποιημένης παρακολούθησης K8s Services μέσω χρήσης ειδικών ετικετών. Εν συντομία βοηθά στην αυτοματοποιημένη δημιουργία ρυθμίσεων στον Prometheus για την προσθήκη κόμβων στόχων που αυτό πρέπει να παρακολουθεί όπως εξηγήθηκε εν συντομία νωρίτερα. Αυτό πρακτικά σημαίνει πως μέσω μιας απλής ρύθμισης στην εγκατάσταση μιας υπηρεσίας μας στο Kubernetes Cluster στο οποίο έχει εγκατασταθεί προηγουμένως το Prometheus-operator μπορούμε να ενεργοποιήσουμε αυτόματα την παρακολούθηση των μετρικών που αυτή παρέχει.

Το k8s-prometheus-adapter [27] είναι μια υλοποίηση των K8s APIs μετρικών που παρουσιάστηκαν στην προηγούμενη ενότητα χρησιμοποιώντας το Prometheus ως πραγματική βάση δεδομένων από την οποία αντλούνται οι τιμές τους. Προσφέρει ένα απλό πλαίσιο κανόνων μέσω των οποίων μπορούν να ρυθμιστούν κανόνες συλλογής και παρουσίασης μετρικών από το Prometheus προς το Kubernetes. Έτσι για παράδειγμα αν συγκεντρώνονται τιμές μετρικών από Containers που αποτελούν ένα K8s Service, κατάλληλος κανόνας μπορεί να συλλέγει αυτές τις μετρικές και να τις

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

παρουσιάζει στο API μετρικών με συγκεκριμένο όνομα που αφορά την αντίστοιχη οντότητα (Service), χωρίς ο χρήστης να γνωρίζει για το monitoring σύστημα που το υλοποιεί. Αυτές τις μετρικές μπορεί να τις χρησιμοποιήσει αυτόματα και ο Horizontal Pod Autoscaler για τον έλεγχο του μεγέθους ενός Deployment μιας υπηρεσίας.

Εκτός από αυτά τα κύρια συστατικά τέλος το kube-prometheus παρέχει και τα εξής λιγότερο σημαντικά για αυτή την εργασία:

- Prometheus Alert Manager για την παρακολούθηση μετρικών και δημιουργία ειδοποιήσεων με την εμφάνιση γεγονότων.
- Prometheus Node exporter για την παρακολούθηση και εξαγωγή μετρικών των λειτουργικών συστημάτων στα οποία αυτό φιλοξενείται τα οποία στη συνέχεια συλλέγει το Prometheus.
- Grafana για την παρουσίαση μετρικών με διαδραστικά γραφήματα για αποδοτική παρακολούθηση.
- Kube-state-metrics για την παρακολούθηση του ίδιου του Kubernetes Cluster.

3.3 Η εφαρμογή

Η εφαρμογή που χρησιμοποιείται για την εκτέλεση των σεναρίων κλιμακωσιμότητας είναι ένας ταξινομητής εικόνων με τεχνολογία τεχνητής νοημοσύνης [28]. Η εφαρμογή αυτή δέχεται HTTP αιτήσεις που περιέχουν εικόνες και καλείται να αποκριθεί για το εάν και κατά πόσο αυτές απεικονίζουν πιθανή εστία φωτιάς.

Η εφαρμογή αυτή τροποποιήθηκε μερικώς με σκοπό τη συλλογή μετρικών λειτουργίας από το σύστημα παρακολούθησης. Συγκεκριμένα εκτίθενται μετρικές μέσω των οποίων μπορεί να υπολογιστεί ο ρυθμός εξυπηρέτησης αιτήσεων καθώς και μέσος χρόνος εξυπηρέτησης αυτών. Μέσω της παρατήρησης των μετρικών αυτών είναι δυνατό να οριστούν οι προδιαγραφές λειτουργίας της εφαρμογής που εξασφαλίζουν τον περιορισμό ποιότητας της υπηρεσίας δεδομένου των διαθέσιμων υπολογιστικών πόρων.

3.4 Παραλλαγές σεναρίου ως προς τον έλεγχο κλιμάκωσης

Όπως αναφέρθηκε προηγουμένως το σενάριο εκτελείται με παραλλαγές ως προς τη ρύθμιση κλιμάκωσης του συστήματος εξυπηρέτησης αιτήσεων.

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

Η πρώτη παραλλαγή αφορά το μέγεθος των στατικών διαθέσιμων υπολογιστικών πόρων ανά Container εφαρμογής. Αυτό ισοδυναμεί με το ελάχιστο μέγεθος μεταβολής των πόρων που διατίθενται στην υπηρεσία συνολικά αφού η ελάχιστη κλιμάκωση που μπορεί να πραγματοποιηθεί είναι αύξηση ή μείωση των εκτελούμενων Containers της εφαρμογής κατά ένα. Εντοπίζονται δύο εκδοχές Container εφαρμογής, μία που λαμβάνει το σύνολο των διαθέσιμων υπολογιστικών πόρων του συστήματος στο οποίο φιλοξενείται και μία όπου οι πόροι είναι περιορισμένοι στο ήμισυ. Στην δεύτερη εκδοχή ο μέγιστος αριθμός Containers είναι διπλάσιος σε σχέση με την πρώτη εκδοχή. Στόχος είναι η μελέτη των επιπτώσεων που επιφέρει το μέγεθος μεταβολής των διαθέσιμων πόρων στη διαχείριση κλιμάκωσης της εφαρμογής.

Η δεύτερη παραλλαγή αφορά τον τύπο της μετρικής μέσω της οποίας ο ελεγκτής κλιμάκωσης λαμβάνει απόφαση. Υπάρχουν οι μετρικές που εκφράζουν το μέγεθος χρησιμοποιούμενων υπολογιστικών πόρων και οι μετρικές που έχουν να κάνουν με χαρακτηριστικά της εφαρμογής όπως ο ρυθμός εξυπηρέτησης αιτήσεων. Η εφαρμογή του προβλήματος κλιμακώνεται με την παρακολούθηση και των δύο αυτών ειδών μετρικών. Στη μία περίπτωση παρακολουθούνται οι χρησιμοποιούμενοι υπολογιστικοί πόροι ενώ στη δεύτερη παρακολουθείται το μέγεθος του ρυθμού εξυπηρέτησης αιτήσεων. Στόχος είναι η μελέτη των επιπτώσεων που επιφέρει η χρήση του ενός ή του άλλου είδους μετρικής στη διαχείριση κλιμάκωσης της εφαρμογής.

3.5 Προσομοίωση φορτίου

Για την προσομοίωση φορτίου έχει δημιουργηθεί μια εφαρμογή η οποία στέλνει τυχαία εικόνες που σχετίζονται με το θέμα της εφαρμογής. Η παραγωγή αιτήσεων ακολουθεί κατανομή Poisson για την προσομοίωση τυχαίων γεγονότων αποστολής εικόνων δεδομένου ενός αναμενόμενου ρυθμού λήψης αιτήσεων. Προκειμένου να γίνει μελέτη της κλιμακωσιμότητας της εφαρμογής ο ρυθμός παραγωγής γεγονότων μεταβάλλεται κατά τη διάρκεια της εκτέλεσης.

Δεδομένου ότι το ίδιο σενάριο εκτελείται τέσσερις φορές, μια φορά για κάθε μία από τις τέσσερις παραλλαγές, η προσομοίωση εκτελείται σε δύο στάδια. Πρώτα δημιουργείται η ακολουθία χρονικών στιγμών γεγονότων αποστολής αίτησης. Η ακολουθία των γεγονότων αυτών ακολουθεί τις προδιαγραφές που αναφέρθηκαν. Στη

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

συνέχεια εκτελείται η πραγματική παραγωγή αιτήσεων προς την εφαρμογή ακολουθώντας τη χρονοσειρά που έχει προετοιμαστεί.

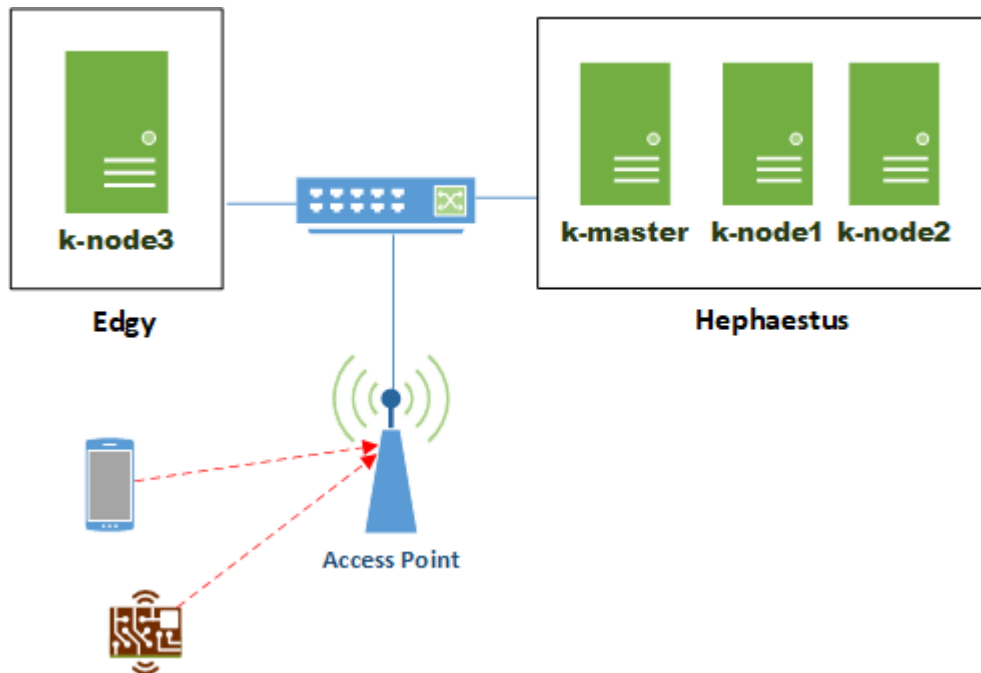
Το σενάριο που εκτελείται σε αυτό το πρόβλημα ξεκινά με ένα χαμηλό ρυθμό παραγωγής αιτήσεων ώστε το σύστημα να ξεκινήσει από το χαμηλότερο επίπεδο χρησιμοποίησης πόρων. Στη συνέχεια σταδιακά ο ρυθμός αυξάνεται με σκοπό τη σταδιακή κλιμάκωση της εφαρμογής αντίστοιχα. Μετά την κλιμάκωση στο μέγιστο επίπεδο λειτουργίας γίνεται σταδιακή μείωση του ρυθμού έως τον αποκλιμάκωση της εφαρμογής στο χαμηλότερο επίπεδο λειτουργίας με σκοπό την επίδειξη απελευθέρωσης υπολογιστικών πόρων.

3.6 Υπολογιστικό σύστημα

Το υπολογιστικό σύστημα που φιλοξενεί την εφαρμογή υλοποιείται σε τέσσερις εικονικούς υπολογιστές (Virtual Machines) οι οποίοι φιλοξενούνται σε δύο φυσικούς υπολογιστές. Οι φυσικοί υπολογιστές είναι ο Hephaestus και ο Edgy.

Ο Hephaestus φιλοξενεί τα k-master, k-node1 και k-node2 VMs ενώ ο Edgy φιλοξενεί το k-node3 VM όπως φαίνεται και στη σχηματική παράσταση που ακολουθεί. Και τα τέσσερα VMs βρίσκονται στο ίδιο τοπικό δίκτυο στο οποίο θα μπορούσαν να είναι συνδεδεμένες οι κινητές συσκευές που παράγουν τις αιτήσεις.

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes



Εικόνα 2: Διάταξη υπολογιστικού συστήματος.

Στα τέσσερα VMs εγκαθίσταται το λογισμικό Kubernetes μέσω kubespray [29] για τη δημιουργία K8s Cluster το οποίο αποτελείται από τα τέσσερα αντίστοιχα Nodes. Το k-master Node εκτελεί ρόλο Master Node στο Cluster.

Στη συνέχεια στο Kubernetes Cluster εγκαθίστανται οι οντότητες που αποτελούν το σύστημα παρακολούθησης που παρέχει το Kube-prometheus [24]. Οι οντότητες αυτές έχουν τροποποιηθεί προηγουμένως ώστε εκτός από την προκαθορισμένη συμπεριφορά να παρέχουν και τις εξής λειτουργίες:

- Τα Containers που εκτελούνται στο Cluster ως μέρος του συστήματος παρακολούθησης να προγραμματίζονται εφόσον είναι δυνατό στο k-master Node ή να διαμοιράζονται ισόποσα στα υπόλοιπα Nodes του, ώστε οι διαθέσιμοι πόροι που περισσεύουν για την εκτέλεση της εφαρμογής να είναι οι μέγιστοι δυνατοί και όσο το δυνατόν πιο δίκαια κατανομημένοι. Σε αυτά τα Nodes προγραμματίζεται η εκτέλεση των Pods της εφαρμογής.
- Ενεργοποιείται η υλοποίηση του custom metrics API του K8s API μέσω του οποίου συλλέγονται οι χρήσιμες μετρικές του παρέχει η εφαρμογή.

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

- Προστίθενται κανόνες μετατροπής μετρικών της εφαρμογής που καταγράφονται στο Prometheus ως μετρικές του Kubernetes custom metrics API.
- Προστίθενται κανόνες καταγραφής μετρικών υψηλότερου επιπέδου που προκύπτουν από συνδυασμό μετρικών χαμηλότερου επιπέδου της εφαρμογής (όπως περιγράφονται στη συνέχεια) ως μέρος των ρυθμίσεων του Prometheus.

Μετά την εγκατάσταση και του συστήματος παρακολούθησης το σύστημα είναι έτοιμο να δεχτεί τη σύνθεση των Kubernetes οντοτήτων που αποτελούν την εφαρμογή αυτού το προβλήματος.

3.7 Μετρικές εφαρμογής και παρατήρηση εφαρμογής

Πριν γίνει η περιγραφή της διάταξης που αφορά την εφαρμογή παρουσιάζονται οι μετρικές που εξάγονται από αυτή. Αυτές οι μετρικές μπορούν να ταξινομηθούν σε δύο κατηγορίες, αυτές που είναι χαμηλότερου επιπέδου και εκφράζουν απλά αριθμητικά μεγέθη κατά τη χρονική στιγμή της συλλογής τους από το Prometheus και αυτές που συντίθενται από τις τιμές των πρώτων και είναι και πιο χρήσιμες για την παρατήρηση λειτουργίας.

Μεταξύ των μετρικών χαμηλού επιπέδου οι ακόλουθες είναι χρήσιμες για αυτό το πρόβλημα:

- Συνολικός αριθμός αιτήσεων που έχει δεχτεί η εφαρμογή. Αυτός είναι ένας μετρητής ο οποίος αυξάνεται καθώς νέες αιτήσεις λαμβάνονται από την εφαρμογή.
- Συνολικός χρόνος εξυπηρέτησης αιτήσεων που έχει δεχτεί η εφαρμογή σε δευτερόλεπτα. Είναι ένα μετρητής ο οποίος καταγράφει τον συνολικό χρόνο εξυπηρέτησης των αιτήσεων που λαμβάνονται.
- Συνολικός χρόνος χρήσης κεντρικής μονάδας επεξεργασίας σε δευτερόλεπτα. Καθώς γίνεται χρήση της ΚΜΕ ο χρόνος που αυτή χρησιμοποιείται προστίθεται στον χρόνο αυτό.
- Μέγεθος μνήμης σε bytes. Είναι ένα μετρητής που καταγράφει το μέγεθος μνήμης που χρησιμοποιείται από την εφαρμογή ανά πάσα χρονική στιγμή.

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

Από αυτές τις μετρικές με κατάλληλο συνδυασμό παράγονται οι ακόλουθες μετρικές υψηλότερου επιπέδου οι οποίες χρησιμοποιούνται άμεσα για την παρατήρηση της συμπεριφοράς της εφαρμογής. Αξίζει να σημειωθεί πως αυτές οι μετρικές καταγράφονται στο Prometheus μέσω κανόνων καταγραφής που ορίζονται όπως αναφέρθηκε προηγουμένως. Αυτές οι μετρικές είναι οι εξής:

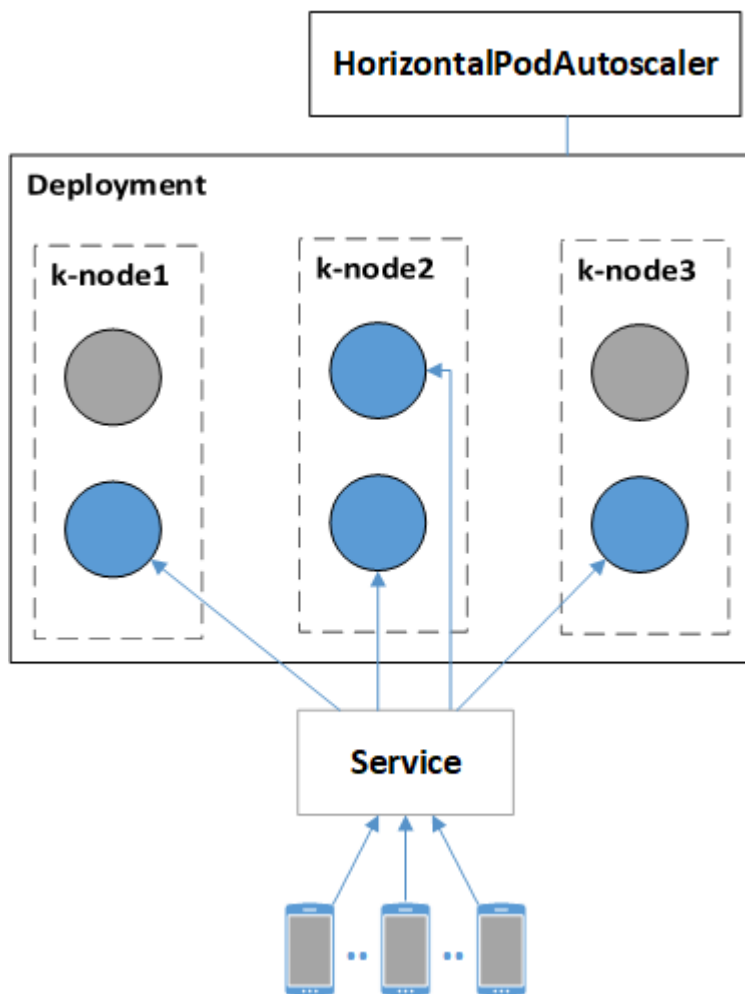
- Ρυθμός εξυπηρέτησης αιτήσεων. Προκύπτει από το μέση μεταβολή του συνολικού αριθμού εξυπηρετούμενων αιτήσεων κατά το τελευταίο χρονικό διάστημα
- Χρόνος επεξεργασίας αιτήσεων. Προκύπτει από τη μέση μεταβολή του συνολικού χρόνου εξυπηρέτησης ως προς τη μέση μεταβολή των εξυπηρετούμενων αιτήσεων κατά το τελευταίο χρονικό διάστημα.
- Χρησιμοποιούμενη ΚΜ. Προκύπτει από τη μέση μεταβολή του συνολικού χρόνου χρήσης της ΚΜΕ κατά το τελευταίο χρονικό διάστημα.
- Ο αριθμός των Pods που εκτελούνται. Κάθε Pod διαθέτει το δικό του μοναδικό αναγνωριστικό κατά την συλλογή των μετρικών του και έτσι δίνεται η δυνατότητα ανά πάσα στιγμή να προσδιορίζεται ο αριθμός των διαθέσιμων Pods.

Σημειώνεται ότι το κάθε Pod που εκτελείται παράγει τις δικές του μετρικές οι οποίες αποθηκεύονται με το αναγνωριστικό του Pod στο σύστημα παρακολούθησης. Από αυτές μπορούμε να εξάγουμε ανά περίπτωση τις συνολικές μετρικές υπηρεσίας που παρατηρείται από την λειτουργία όλων των Pods που ανήκουν σε αυτή. Συγκεκριμένα ο μέσος ρυθμός εξυπηρέτησης, η χρησιμοποιούμενη ΚΜΕ και η χρησιμοποιούμενη μνήμη προκύπτει από το άθροισμα των αντίστοιχων μεγεθών των επιμέρους Pods, ενώ ο μέσος χρόνος επεξεργασίας αιτήσεων προκύπτει ως μέση τιμή των αντίστοιχων χρόνων των επιμέρους Pods.

3.8 Σύνθεση εφαρμογής

Η σύνθεση του συστήματος που αποτελεί την εφαρμογή εξυπηρέτησης αιτήσεων αποτελείται από τις ακόλουθες οντότητες του Kubernetes.

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes



Εικόνα 3: Σύμβαση εφαρμογής. Οι κύκλοι συμβολίζουν τα πιθανά Pods που μπορούν να φιλοξενηθούν στα Nodes. Με μπλε χρώμα συμβολίζονται τα Pods που εξυπηρετούν αιτήσεις της εφαρμογής.

3.8.1 Deployment

Αυτή είναι η οντότητα μέσω της οποίας προγραμματίζεται η εκτέλεση ενός αριθμού πανομοιότυπων Pods του container της εφαρμογής. Εκτός των ρυθμίσεων που αφορούν την εφαρμογή, στο Deployment ορίζεται το που εκτελούνται τα Pods και οι υπολογιστικοί πόροι που παρέχονται σε αυτό. Ορίζεται έτσι όλα τα Pods του Deployment να εκτελούνται σε ένα εκ των k-node1, k-node2 και k-node3 Nodes.

Ως προς τους υπολογιστικούς πόρους που έχουν στη διάθεσή τους τα Pods ορίζονται δύο παραλλαγές του Deployment:

- Μία με Pods που έχουν στη διάθεσή τους όλους τους διαθέσιμους πόρους του Node που φιλοξενούνται. Σε αυτή την περίπτωση ο μέγιστος αριθμός Pods που μπορεί να εκτελεστεί στο Cluster είναι τρία λόγω των τριών διαθέσιμων Nodes

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

στα οποία μπορεί να εκτελεστούν τα Pods του Deployment. Στο εξής αυτή η παραλλαγή του Deployment θα ονομάζεται μεγάλο Deployment.

- Μία με Pods που έχουν στη διάθεσή τους τους μισούς διαθέσιμους πόρους του Node που φιλοξενούνται. Σε αυτή την περίπτωση ο μέγιστος αριθμός Pods που μπορεί να εκτελεστεί στο Cluster είναι έξι. Στο εξής αυτή η παραλλαγή του Deployment θα ονομάζεται μικρό Deployment.

3.8.2 HorizontalPodAutoscaler

Αυτή είναι η οντότητα μέσω της οποίας εκτελείται η αυτόματη κλιμάκωση/αποκλιμάκωση της εφαρμογής. Αυτό πραγματοποιείται με έλεγχο του αριθμού των Pods που εκτελούνται ανά πάσα στιγμή ως μέρος του Deployment της εφαρμογής (προσδιορισμός του Deployment «replicas»). Επειδή κάθε Pod έχει στη διάθεσή του υπολογιστικούς πόρους σταθερού μεγέθους, μέσω της μεταβολής του αριθμού των Pods μεταβάλλονται και οι πόροι που είναι διαθέσιμοι για την εφαρμογή συνολικά. Οι ρυθμίσεις που παρέχονται είναι οι εξής:

- Το Deployment που επηρεάζεται από τον έλεγχο.
- Ο ελάχιστος και μέγιστος αριθμός Pods του Deployment.
- Μια σειρά από μετρικές μέσω των οποίων ελέγχεται ο αριθμός των Pods. Σε αυτό το πρόβλημα μόνο μία μετρική ορίζεται κάθε φορά παρόλο που συνδυασμός περισσότερων είναι εφικτός.

Ως προς τις παραλλαγές των Deployments που παρουσιάστηκαν και την μετρική μέσω της οποίας ελέγχεται ο αριθμός των Pods ορίζονται τέσσερις παραλλαγές HPA οντοτήτων για το ίδιο σενάριο, ως συνδυασμός των εξής ρυθμίσεων:

- Ο μέγιστος αριθμός Pods ορίζεται σε τρία Pods για το μεγάλο Deployment και σε έξι για το μικρό. Και στις δύο περιπτώσεις ο ελάχιστος αριθμός ορίζεται στο ένα Pod.
- Η μετρική έλεγχου ορίζεται είτε η χρησιμοποιούμενη ΚΜΕ από την εφαρμογή είτε ο μέσος ρυθμός άφιξης αιτήσεων στην εφαρμογή.

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

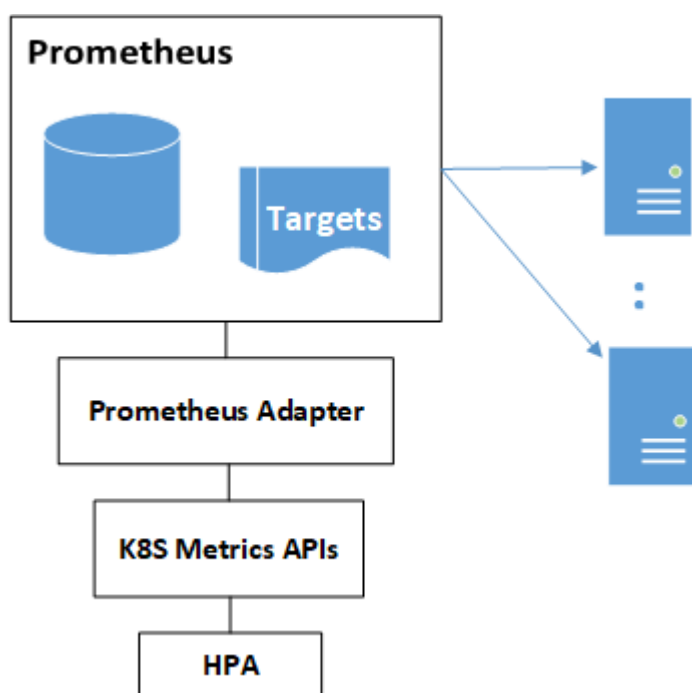
3.8.3 Service

Είναι η οντότητα η οποία αναλαμβάνει τον διαμοιρασμό των αιτήσεων στα Pods που Deployment που τις εξυπηρετούν.

3.8.4 Service Monitor

Είναι μια βοηθητική οντότητα μέσω της οποίας αυτοματοποιείται η παρακολούθηση των Pods του Service της εφαρμογής. Καθώς προκύπτουν νέα ή καταργούνται παλιά Pods, ανανεώνονται αυτόματα από αυτή την οντότητα οι στόχοι του συστήματος παρακολούθησης. Έτσι εξασφαλίζεται η διαθεσιμότητα των μετρικών που χρησιμοποιεί ο HPA για τον έλεγχο της εφαρμογής από όλα τα Pods που τα παράγουν.

Επίσης σε αυτή την οντότητα ορίζεται η περίοδος συλλογής των μετρικών χαμηλού επιπέδου των Pods της υπηρεσίας από το σύστημα παρακολούθησης.



Εικόνα 4: Σύστημα παρακολούθησης. Αποτελείται από το λογισμικό Prometheus που διαθέτει τη βάση δεδομένων και πραγματοποιεί περιοδική συλλογή μετρικών και τον Prometheus Adapter που εκθέτει μετρικές του Prometheus ως μετρικές των K8s metrics APIs. Από αυτά τα APIs ένας HorizontalPodAutoscaler ζητάει τιμές μετρικών προκειμένου να ελέγξει την κλιμάκωση μιας εφαρμογής.

3.9 Σχετικά με τους χρόνους μετρήσεων

Οι μετρικές υψηλού επιπέδου που παρουσιάζουν την κατάσταση της εφαρμογής καταγράφονται κάθε τριάντα δευτερόλεπτα. Επιλέγεται αυτή η περίοδος για την

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

καταγραφή νέων τιμών ως περίοδος αναφοράς για την μελέτη της εξέλιξης της λειτουργίας της υπηρεσίας.

Σύμφωνα με αυτή την περίοδο ορίζεται και η περίοδος συλλογής μετρικών χαμηλού επιπέδου στα δεκαπέντε δευτερόλεπτα, δηλαδή στο μισό χρόνο από την περίοδο αναφοράς.

Το μέγεθος του τελευταίου χρονικού διαστήματος πίσω στο χρόνο που χρησιμοποιείται σε υπολογισμούς μέσου ρυθμού μεταβολής μετρικών χαμηλού επιπέδου κατά τον υπολογισμό μετρικών υψηλού επιπέδου ορίζεται στα εκατόν είκοσι δευτερόλεπτα. Χρησιμοποιείται διάστημα τέσσερις φορές μεγαλύτερο από την περίοδο καταγραφής προκειμένου να περιορίζεται η επίδραση ακραίων περιστασιακών τιμών που μπορεί να παρατηρηθούν τη στιγμή της μεταβολής του επιπέδου κλιμάκωσης.

3.10 Σχετικά με το επίπεδο λειτουργίας της εφαρμογής

Για την εκτέλεση της παραλλαγής του σεναρίου με έλεγχο κλιμακωσιμότητας μέσου εξυπηρετούμενου ρυθμού αιτήσεων εκτελέστηκαν πειράματα λειτουργίας με μοναδικά Pods των δύο ειδών των Deployments που χρησιμοποιούνται σε αυτή την μελέτη. Παρατηρώντας τη χρήση των υπολογιστικών πόρων και τις αντίστοιχες τιμές λειτουργίας της εφαρμογής προσδιορίστηκαν εφικτά επίπεδα λειτουργίας που εξασφαλίζουν τον περιορισμό της ποιότητας υπηρεσίας της εφαρμογής. Το επίπεδο λειτουργίας που επιλέγεται για αυτή την εργασία είναι τέτοιο που διατηρεί το μέσο χρόνο επεξεργασίας αιτήσεων κάτω από τα δύομισι δευτερόλεπτα.

3.11 Σχετικά με την επεξεργαστική ισχύ των υπολογιστών

Οι φυσικοί υπολογιστές που χρησιμοποιούνται σε αυτή τη διάταξη είναι ανόμοιοι ως προς την επεξεργαστική ισχύ. Το Kubernetes βλέπει ως ισοδύναμο τον έναν πυρήνα KME του k-node1 που φιλοξενείται στον Hephaestus με τον έναν πυρήνα του k-node3 που φιλοξενείται στον Edgy. Στην πραγματικότητα, στο ίδιο επίπεδο λειτουργίας στα Hephaestus Pods επιτυγχάνεται υψηλότερος ρυθμός επεξεργασίας αιτήσεων από ό,τι στα Edgy Pods. Αντίστοιχα, με τον ίδιο ρυθμό επεξεργασίας αιτήσεων από όλα τα Pods, ο μέσος χρόνος επεξεργασίας αιτήσεων στα Hephaestus Pods είναι χαμηλότερος από ό,τι στο Edgy Pods. Στον πίνακα που ακολουθεί εμφανίζονται τα επίπεδα

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

λειτουργίας των Pods του μικρού και του μεγάλου Deployment όταν αυτά φιλοξενούνται στον Edgy ή στον Hephaestus προκειμένου να επιτευχθεί καθυστέρηση επεξεργασίας αιτήσεων μικρότερη των δύομισι δευτερολέπτων.

Φυσικός Υπολογιστής	Deployment	Μέσος Χρόνος Επεξεργασίας Αιτήσεων (s)	Μέσος Αριθμός Αιτήσεων ανά 30s	Ελάχιστος Αριθμός Αιτήσεων ανά 30s	Μέγιστος Αριθμός Αιτήσεων ανά 30s
Edgy	μικρό	2.5	3.4192	0	8
	μεγάλο	2.5	8.6115	8	15
Hephaestus	μικρό	2.5	6.129	0	10
	μεγάλο	2.5	15.2	10	20

Το Kubernetes Service που κατανέμει το φορτίο στα διαθέσιμα Pods δεν παρέχει τη δυνατότητα για κυκλική δρομολόγηση με βάρη. Άρα δεν υπάρχει η δυνατότητα να γίνει κατανομή φορτίου με βάση το Node στο οποίο φιλοξενείται το Pod στόχος, προκειμένου να εξασφαλίζονται οι μέσοι χρόνοι επεξεργασίας αιτήσεων στο ίδιο επίπεδο για όλα τα Pods. Για αυτό το λόγο επιλέγεται αλγόριθμος κατανομής αιτήσεων με βάση τον αριθμό των ανοιχτών συνδέσεων του κάθε Pod. Σύμφωνα με αυτόν τον αλγόριθμο οι νέες αιτήσεις κατευθύνονται προς το Pod που διαθέτει τον μικρότερο αριθμό ενεργών αιτήσεων. Αυτό έχει ως αποτέλεσμα μια σχεδόν ισομερή κατανομή στα Pods χωρίς να επιβαρύνονται ιδιαίτερα τα Pods του Edgy. Οι μέσοι χρόνοι επεξεργασίας αιτήσεων εμφανίζονται υψηλότεροι στον Edgy από ό,τι στον Hephaestus όμως το επίπεδο λειτουργίας κάθε φορά θα είναι τέτοιο ώστε ο μέσος χρόνος επεξεργασίας αιτήσεων στα Edgy Pods να μην υπερβαίνει τα δύομισι δευτερόλεπτα.

3.12 Τιμές στόχοι για τον ελεγκτή κλιμάκωσης

Για κάθε τύπο μετρικής ελέγχου ορίζεται η τιμή στόχου έτσι ώστε η παρατηρούμενη τιμή την ύστατη στιγμή αλλαγής κλιμάκωσης να μην υπερβαίνει το επίπεδο λειτουργίας που έχει οριστεί. Λαμβάνοντας έτσι υπόψιν:

- τα επίπεδα λειτουργίας των Pods της εφαρμογής,
- το ότι είναι διπλάσια η πιθανότητα ένα Pod να εκτελείται στον Hephaestus από ό,τι στον Edgy,

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

- ο ελεγκτής έχει ρυθμιστεί στο να κλιμακώνει σε τιμές μετρικής ελέγχου που απέχουν τουλάχιστον 15% από την τιμή στόχου,

ορίζονται οι παρακάτω τιμές στόχοι του ελεγκτή στα τέσσερα διαφορετικά σενάρια.

Σενάριο	Μέγιστή τιμή (στόχος +15%)	Τιμή στόχος	Μονάδα μέτρησης
Μικρό Ρυθμός αιτήσεων	5,23	4,54	Αιτήσεις ανά 30s
Μεγάλο Ρυθμός αιτήσεων	13,0	11,3	Αιτήσεις ανά 30s
Μικρό Χρησιμοποιούμενη ΚΜΕ	97,85%	85%	Ποσοστό %
Μεγάλο Χρησιμοποιούμενη ΚΜΕ	97,85%	85%	Ποσοστό %

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

4 Πειραματικά Αποτελέσματα

Σύμφωνα με όσα αναφέρθηκαν εκτελείται ένα σενάριο κλιμάκωσης του λαμβανόμενου όγκου αιτήσεων. Το σενάριο αυτό εκτελείται τέσσερις φορές μία για τον κάθε συνδυασμό μεγέθους Deployment και τύπου μετρικής ελέγχου:

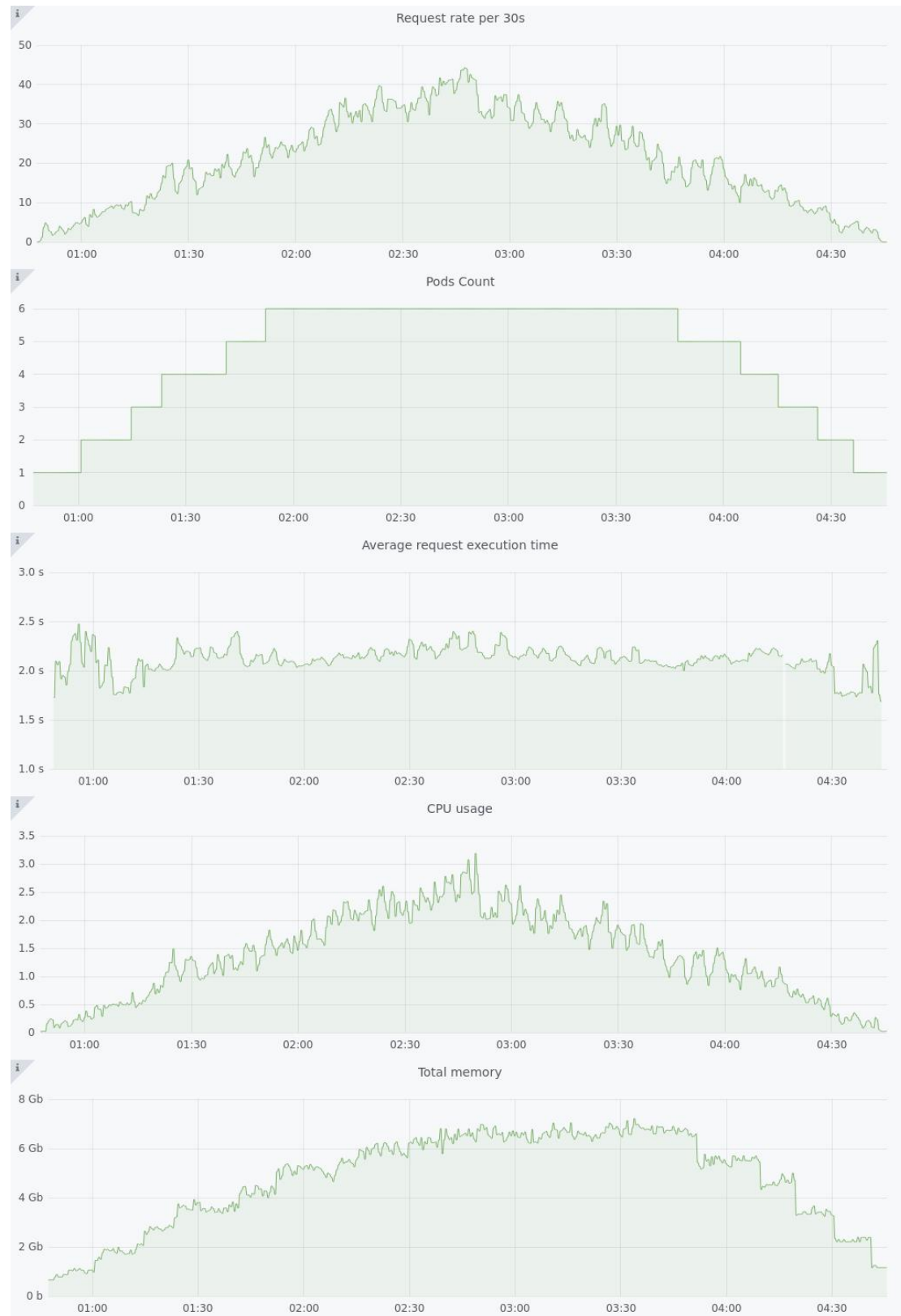
1. Μικρό Deployment με μετρική ελέγχου τον μέσο ρυθμό επεξεργασίας αιτήσεων. Ο μέγιστος αριθμός Pods είναι έξι (Εικόνα 5).
2. Μεγάλο Deployment με μετρική ελέγχου τον μέσο ρυθμό επεξεργασίας αιτήσεων. Ο μέγιστος αριθμός Pods είναι τρία (Εικόνα 6).
3. Μικρό Deployment με μετρική ελέγχου τη μέση χρησιμοποιούμενη ΚΜΕ. Ο μέγιστος αριθμός Pods είναι έξι (Εικόνα 7).
4. Μεγάλο Deployment με μετρική ελέγχου τη μέση χρησιμοποιούμενη ΚΜΕ. Ο μέγιστος αριθμός Pods είναι τρία (Εικόνα 8).

Ο ρυθμός αιτήσεων στο σενάριο αυτό ξεκινά από πολύ χαμηλό επίπεδο με απαιτούμενο αριθμό Pods στο ελάχιστο (ένα Pod σε κάθε περίπτωση). Ο ρυθμός αυξάνεται σταδιακά έως ότου να πραγματοποιηθεί κλιμάκωση στον μέγιστο αριθμό Pods. Σε αυτή την κατάσταση χρησιμοποιούνται οι μέγιστοι διαθέσιμοι πόροι του υπολογιστικού συστήματος. Στη συνέχεια ο ρυθμός ελαττώνεται σταδιακά ώστε αντίστοιχα να παρατηρηθεί αποκλιμάκωση έως το ένα Pod και στην ελάχιστη δέσμευση πόρων του συστήματος από την εφαρμογή.

Στη συνέχεια παρουσιάζονται οι γραφικές απεικονίσεις των σεναρίων αυτών. Συγκεκριμένα παρουσιάζονται για κάθε περίπτωση κατά σειρά οι τιμές των εξής μεγεθών κάθε στιγμή:

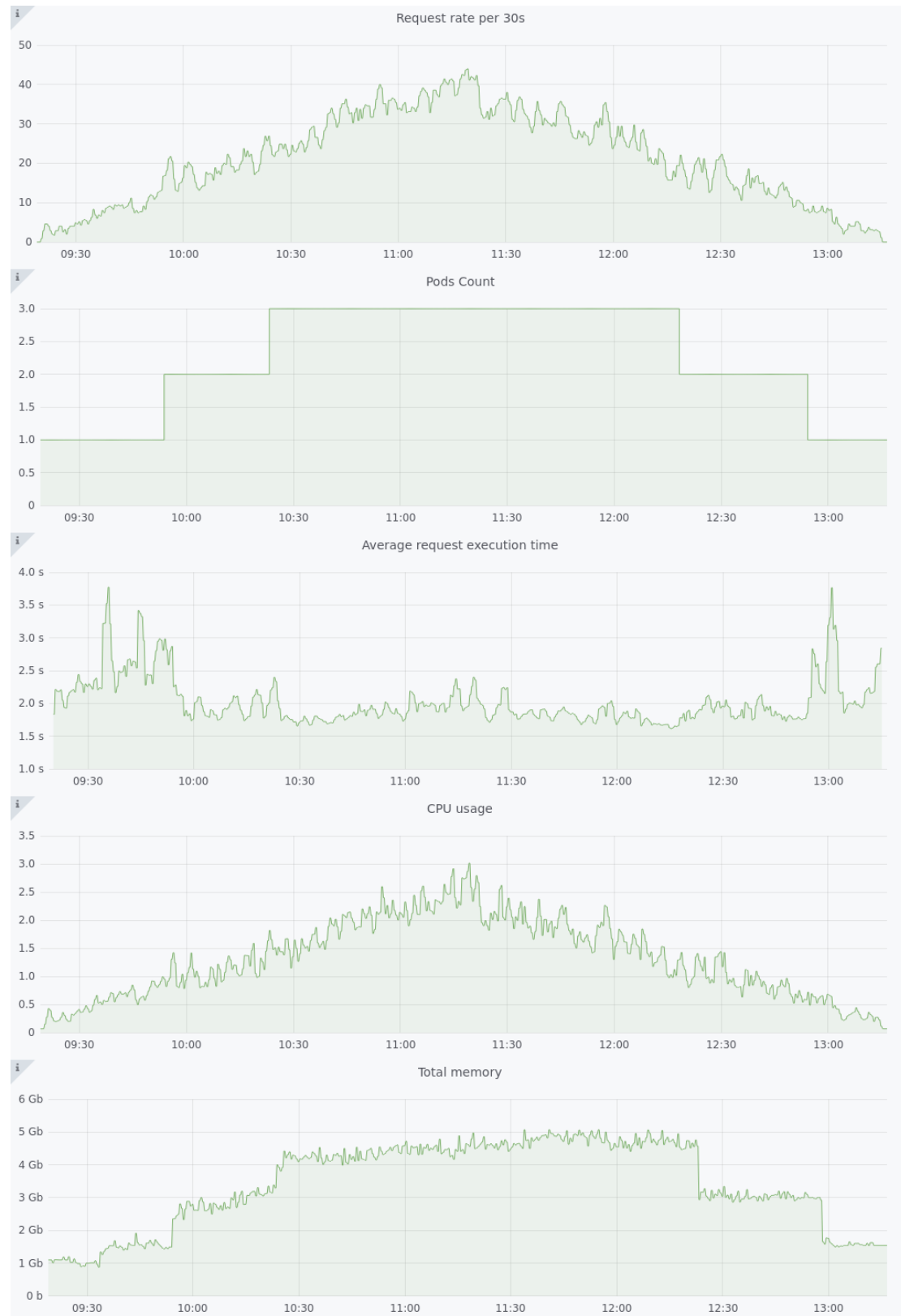
- Ο μέσος ρυθμός επεξεργασίας αιτήσεων ανά τριάντα δευτερόλεπτα.
- Το πλήθος των Pods της εφαρμογής που εκτελούνται.
- Ο μέσος χρόνος επεξεργασίας αιτήσεων.
- Η χρησιμοποιούμενη ΚΜΕ.
- Η χρησιμοποιούμενη μνήμη.

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes



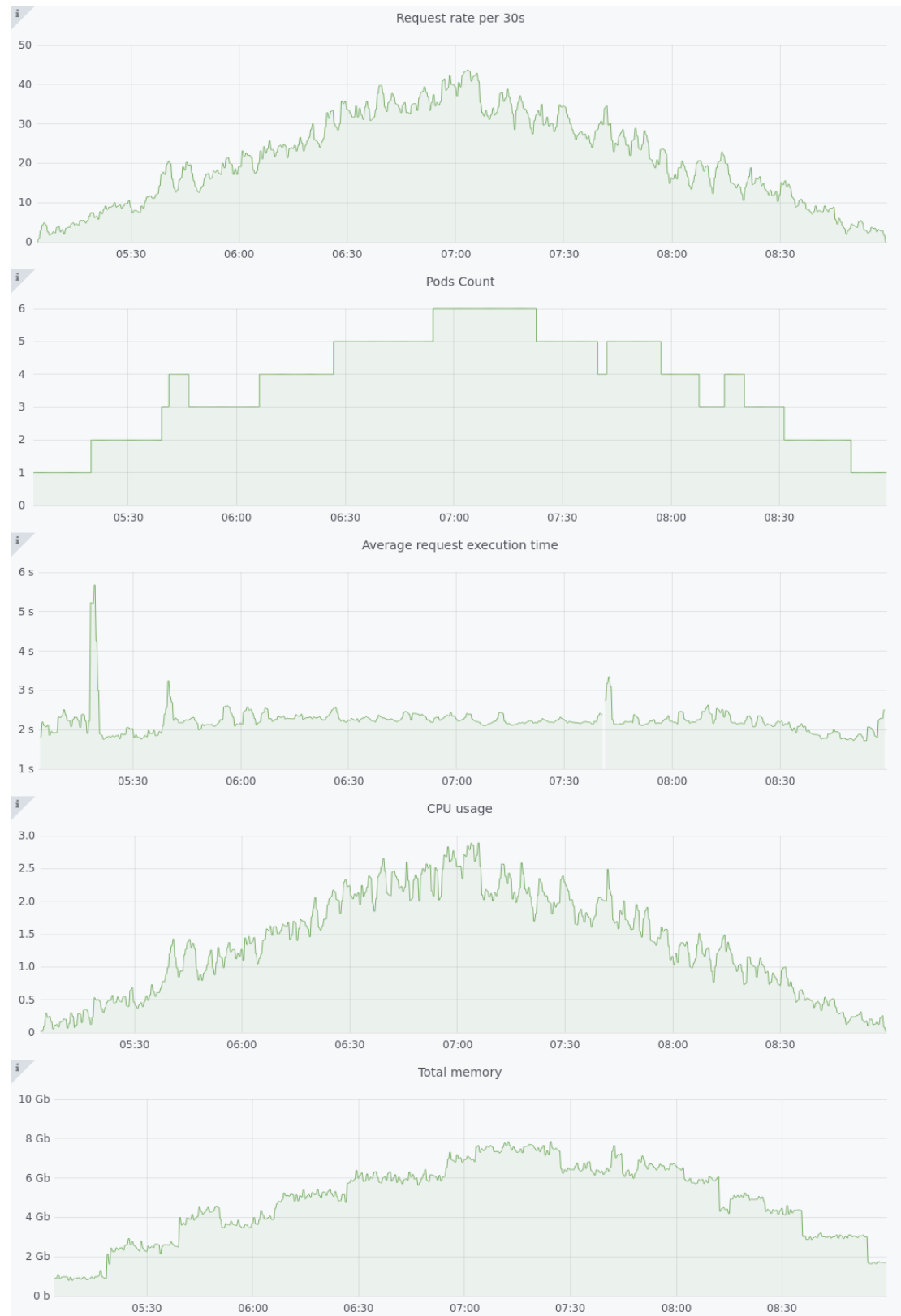
Εικόνα 5: Έλεγχος κλιμάκωσης μικρού Deployment μέσω του μέσου ρυθμού επεξεργασίας αιτήσεων.

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes



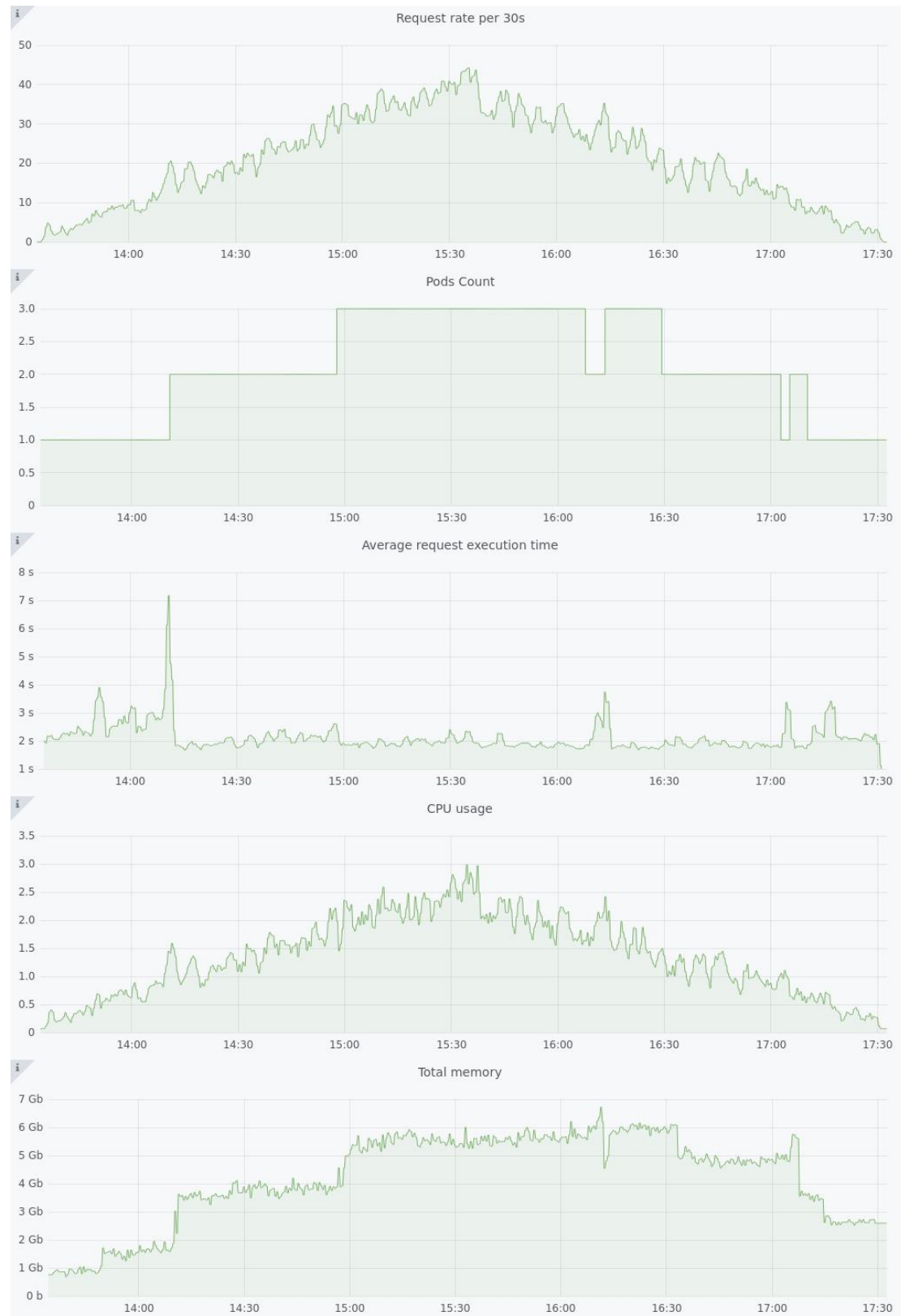
Εικόνα 6: Έλεγχος κλιμάκωσης μεγάλου Deployment μέσω του μέσου ρυθμού επεξεργασίας αιτήσεων.

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes



Εικόνα 7: Έλεγχος κλιμάκωσης μικρού Deployment μέσω της μέσης χρησιμοποιούμενης ΚΜΕ.

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes



Εικόνα 8: Έλεγχος κλιμάκωσης μεγάλου Deployment μέσω της μέσης χρησιμοποιούμενης ΚΜΕ.

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

Μερικές παρατηρήσεις που μπορούν να γίνουν πάνω στα αποτελέσματα παρουσιάζονται στη συνέχεια:

- Γενικά παρατηρείται η ζητούμενη κλιμάκωση και αποκλιμάκωση της εφαρμογής καθώς το φορτίο αιτήσεων μεταβάλλεται αντίστοιχα.
- Ο μέσος ρυθμός επεξεργασίας αιτήσεων μεταβάλλεται σχεδόν ανάλογα με τη χρησιμοποιούμενη ΚΜΕ. Αυτό δεν ισχύει και για τη χρησιμοποιούμενη μνήμη παρόλο που παρατηρείται κάποια σχετική κλιμάκωση και σε αυτό το μέγεθος. Αυτή η παρατήρηση είναι αναμενόμενη επειδή είναι τέτοιος ο τύπος της εφαρμογής που η απόδοσή της επηρεάζεται άμεσα από διαθέσιμη ΚΜΕ και όχι τόσο από την μνήμη.
- Ο μέσος χρόνος επεξεργασίας αιτήσεων, εξαιρουμένων περιορισμένων χρονικών στιγμών, διατηρείται στο επίπεδο που έχει τεθεί ως στόχος λειτουργίας, δηλαδή κάτω από τα δύομισι δευτερόλεπτα. Γενικά παρατηρείται ότι στις περιπτώσεις με μεγάλο Deployment ο χρόνος αυτός διατηρείται σε ελαφρώς χαμηλότερο επίπεδο από ό,τι στις περιπτώσεις μικρού Deployment.
- Στις παραλλαγές που χρησιμοποιείται μεγάλο Deployment παρατηρείται ελαφρώς χαμηλότερος μέσος χρόνος επεξεργασίας αιτήσεων από ό,τι με μικρό Deployment.
- Μπορεί να παρατηρηθεί ότι κατά την μεταβολή του αριθμού των Pods ο χρόνος επεξεργασίας αιτήσεων προσωρινά ξεπερνά το αναμενόμενο επίπεδο λειτουργίας. Αυτό οφείλεται στο ότι δεν δίνεται χρόνος στο νέο Pod να αρχικοποιηθεί προτού το K8s Service δρομολογήσει μέρος των αιτήσεων που λαμβάνει προς αυτό με αποτέλεσμα κάποιες από τις λαμβανόμενες αιτήσεις να επεξεργάζονται με καθυστέρηση ή να απορρίπτονται. Αυτή η συμπεριφορά μπορεί να αντιμετωπιστεί μέσω του μηχανισμού των Readiness Probes για τα Pods που προσφέρει το K8s.
- Κατά την κλιμάκωση μέσω του ρυθμού επεξεργασίας αιτήσεων παρατηρείται πιο ομαλή κλιμάκωση/αποκλιμάκωση από ό,τι μέσω της μέσης χρησιμοποιούμενης ΚΜΕ. Κατά τον έλεγχο με τον δεύτερο τρόπο παρατηρούνται προσωρινές αυξομειώσεις στον αριθμό των Pods που

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

οφείλονται σε υψηλές μεταβολές της χρησιμοποιούμενης ΚΜΕ εκείνες τις χρονικές στιγμές (ιδιαίτερα με μικρό Deployment).

- Παρατηρείται ότι για όσο υπάρχει μόνο ένα Pod της εφαρμογής για να δέχεται αιτήσεις ο παρατηρούμενος μέσος χρόνος επεξεργασίας αιτήσεων είναι πιο ευμετάβλητος από όταν υπάρχουν περισσότερα Pods. Είναι πιθανό η εξισορρόπηση του φορτίου αιτήσεων καθώς και ο αλγόριθμος εξισορρόπησης με βάση τις ελάχιστες συνδέσεις που χρησιμοποιείται να παίζει κάποιο ρόλο σε αυτό.

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

5 Συμπεράσματα και μελλοντική εργασία

5.1 Συμπεράσματα

Μέσω των αποτελεσμάτων της προσομοίωσης που παρουσιάστηκαν αποδείχθηκε ότι είναι δυνατός ο έλεγχος κλιμακωσιμότητας εφαρμογής στα άκρα του δικτύου μέσω του λογισμικού Kubernetes. Η χρήση τέτοιας διάταξης επιτρέπει την αποδοτική εκμετάλλευση υπολογιστικών πόρων ενώ ταυτόχρονα εκπληρώνεται ο περιορισμός της ποιότητας υπηρεσίας, αφού η τιμή στόχος του ελεγκτή περιγράφει το προηγουμένως εντοπισμένο ζητούμενο επίπεδο λειτουργίας της εφαρμογής.

Μέσω του Kubernetes ο έλεγχος της κλιμάκωσης μπορεί το ίδιο εύκολα να επιτευχθεί είτε με απευθείας παρατήρηση των χρησιμοποιούμενων πόρων είτε μέσω παρατήρησης ποιοτικών μετρικών που σχετίζονται άμεσα με την αποστολή της εφαρμογής. Στο πείραμα αυτής της εργασίας, αυτές ήταν ο μέσος ρυθμός επεξεργασίας αιτήσεων και ο μέσος χρόνος επεξεργασίας αυτών. Η χρήση τέτοιων μετρικών επιτρέπει την αναγνώριση επιπέδων λειτουργίας που σχετίζονται άμεσα με τον στόχο της εφαρμογής και όχι εμμέσως μέσω των καταναλούμενων πόρων κάνοντας έτσι τη διαχείριση της εφαρμογής πιο εύκολη.

Το πλήθος των πόρων που διατίθενται σε κάθε Container της εφαρμογής δείχνει να επηρεάζει ελαφρώς την απόδοσή της. Για το λόγο αυτό, η εκτέλεση μετρήσεων με διάφορες εκδοχές παρεχόμενων πόρων ανά Container κρίνεται απαραίτητη ώστε να προσδιοριστεί η πιο κατάλληλη εκδοχή. Κατά των προσδιορισμό αυτό πρέπει να λαμβάνεται επίσης υπόψιν ότι είναι ενδεχομένως, λόγω περιορισμένων πόρων, πιο εύκολο για το Kubernetes να προγραμματίσει την εκτέλεση ενός μικρότερου Container σε κάποιο Node από ό,τι ενός μεγαλύτερου.

Τέλος, κατά τη κλιμάκωση εφαρμογής μέσω του Kubernetes μπορούν να χρησιμοποιηθούν τα εντοπισμένα επίπεδα λειτουργίας υπό συγκεκριμένους διαθέσιμους πόρους και να γίνει χρήση αυτών στον μηχανισμό κλιμάκωσης που αυτό προσφέρει. Έχοντας ως δεδομένο τα επίπεδα λειτουργίας ενός Container, η οριζόντια κλιμάκωση που υποστηρίζει το Kubernetes επιτρέπει την θεωρητικά απεριόριστη

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

κλιμάκωση της εφαρμογής μέσω παραλληλισμού και κατανομής αιτήσεων στα Containers που την εκτελούν.

5.2 Μελλοντική Εργασία

Αυτή η εργασία παρουσίασε μια απλή και μικρής κλίμακας χρήση του μηχανισμού κλιμάκωσης του λογισμικού Kubernetes. Υπάρχει πλήθος άλλων λειτουργιών που αυτό προσφέρει και θα μπορούσε να γίνει χρήση τους σε εφαρμογές που εκτελούνται στα άκρα του δικτύου.

Μια παραλλαγή αυτής της εργασίας θα μπορούσε να πραγματοποιεί τοπικά προσαρμοσμένο έλεγχο κλιμακωσιμότητας. Συγκεκριμένα μια εγκατάσταση θα μπορούσε να χωριστεί σε ζώνες και στη συνέχεια να εκτελείται μοντέλο κλιμάκωσης κοντά στην περιοχή που παρατηρείται μεταβολή ζήτησης. Έτσι, μια ζώνη μπορεί να έχει μεγαλύτερες μεταβολές στην ζήτηση από μία άλλη ή μια ζώνη να διαθέτει διαφορετικό επίπεδο λειτουργίας σε σχέση με άλλες.

Μια άλλη χρήση αυτού του μηχανισμού που θα μπορούσε να γίνει συμπληρωματικά είναι και η παρακολούθηση της κατάστασης των εφαρμογών. Το σύστημα παρακολούθησης που εγκαταστάθηκε για αυτήν την εργασία εκτός από την έκθεση μετρικών παρακολούθησης της κατάστασης των εφαρμογών, επιτρέπει την αποστολή ειδοποιήσεων όταν για παράδειγμα παρατηρηθεί κάποια απρόσμενη κατάσταση.

Ακόμη όσον αφορά την κλιμακωσιμότητα μέσω του λογισμικού Kubernetes, κάποιοι επιπλέον μηχανισμοί αυτόματης κλιμάκωσης μπορούν να δοκιμαστούν σε εφαρμογές που εκτελούνται στα άκρα του δικτύου. Οι επιπλέον μηχανισμοί που προσφέρονται από το Kubernetes είναι ο Cluster Autoscaler, ο Vertical Pod Autoscaler και ο Addon Resizer [30]. Σημειώνεται ότι το Cluster Autoscaler μπορεί να εφαρμοστεί μόνο σε δημόσιους παρόχους K8s Cluster. Για ιδιωτικά Cluster μια αντίστοιχη λειτουργία μπορεί να επιτευχθεί για παράδειγμα μέσω του μηχανισμού κλιμάκωσης που παρέχει το kubespray [29]. Ο Vertical Pod Autoscaler μπορεί να χρησιμοποιηθεί αντί του Horizontal Pod Autoscaler για τον έλεγχο των πόρων που έχουν στη διάθεσή τους τα Pods ενός Deployment. Μάλιστα είναι δυνατός ο συνδυασμός τους εφόσον ο δεύτερος δεν πραγματοποιεί έλεγχο μέσω των χρησιμοποιούμενων υπολογιστικών πόρων. Ο Addon Resizer είναι μια απλή παραλλαγή του Vertical Pod Autoscaler που μεταβάλλει

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

τα όρια των χρησιμοποιούμενων πόρων βάσει του αριθμού των Nodes που υπάρχουν σε ένα Cluster.

Τέλος, δεδομένου ότι οι πόροι που είναι διαθέσιμοι σε υπολογιστικά συστήματα σε εφαρμογές που εκτελούνται στα άκρα του δικτύου είναι περιορισμένοι, ιδιαίτερο ενδιαφέρον θα είχε μια μελέτη κλιμακωσιμότητας που θα χρησιμοποιούσε ένα υβριδικό μοντέλο παροχής Kubernetes Cluster [31]. Σε μια τέτοια περίπτωση θα μπορούσε να συνενωθεί το τοπικό στα άκρα του δικτύου Cluster, με ένα εξωτερικό ενός δημοσίου παρόχου που διαθέτει θεωρητικά άπειρους πόρους. Όταν οι πόροι στο τοπικό Cluster θα εξαντλούνταν, κατάλληλη μεταφόρτωση υπολογισμών θα μπορούσε αυτόματα να πραγματοποιείται προς το δημόσιο Cluster.

Παραπομπές

- [1] T. Yu-han, L. Ravindranath, S. Deng, and T. Y. Chen, “Continuous , Real-Time Object Recognition on Mobile Devices Categories and Subject Descriptors,” 2018.
- [2] P. Mach and Z. Becvar, “Mobile Edge Computing: A Survey on Architecture and Computation Offloading,” *IEEE Commun. Surv. Tutorials*, vol. 19, Aug. 2017, doi: 10.1109/COMST.2017.2682318.
- [3] N. Fernando, S. W. Loke, and W. Rahayu, “Mobile cloud computing: A survey,” *Futur. Gener. Comput. Syst.*, vol. 29, no. 1, pp. 84–106, 2013, doi: 10.1016/j.future.2012.05.023.
- [4] Z. Xiao, W. Song, and Q. Chen, “Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1107–1117, 2013, doi: 10.1109/TPDS.2012.283.
- [5] Kubernetes, “Kubernetes Container Orchestrator,” *kubernetes.io*. <https://kubernetes.io/>.
- [6] Docker, “Docker,” *docker.com*. <https://www.docker.com/>.
- [7] A. Yousefpour *et al.*, “All one needs to know about fog computing and related edge computing paradigms: A complete survey,” *J. Syst. Archit.*, vol. 98, no. February, pp. 289–330, 2019, doi: 10.1016/j.sysarc.2019.02.009.
- [8] T. Zhao, S. Zhou, X. Guo, and Z. Niu, “Tasks scheduling and resource allocation in heterogeneous cloud for delay-bounded mobile edge computing,” in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–7, doi: 10.1109/ICC.2017.7996858.
- [9] A. Mehta, W. Tarneberg, C. Klein, J. Tordsson, M. Kihl, and E. Elmroth, “How beneficial are intermediate layer data centers in mobile edge networks?,” *Proc. - IEEE 1st Int. Work. Found. Appl. Self-Systems, FAS-W 2016*, pp. 222–229, 2016, doi: 10.1109/FAS-W.2016.55.
- [10] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, “Fog computing may

- help to save energy in cloud computing,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1728–1739, 2016, doi: 10.1109/JSAC.2016.2545559.
- [11] P. Hao, Y. Bai, X. Zhang, and Y. Zhang, “Edgecourier: An edge-hosted personal service for low-bandwidth document synchronization in mobile cloud storage services,” *2017 2nd ACM/IEEE Symp. Edge Comput. SEC 2017*, 2017, doi: 10.1145/3132211.3134447.
- [12] A. Singh, N. Auluck, O. Rana, A. Jones, and S. Nepal, “RT-SANE : Real time security aware scheduling on the network edge,” *UCC 2017 - Proc. the10th Int. Conf. Util. Cloud Comput.*, no. July 2019, pp. 131–140, 2017, doi: 10.1145/3147213.3147216.
- [13] E. Ahmed *et al.*, “Bringing Computation Closer toward the User Network: Is Edge Computing the Solution?,” *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 138–144, 2017, doi: 10.1109/MCOM.2017.1700120.
- [14] H. Zhu and C. Huang, “Availability-aware mobile edge application placement in 5G networks,” *2017 IEEE Glob. Commun. Conf. GLOBECOM 2017 - Proc.*, vol. 2018-Janua, no. December, pp. 1–6, 2017, doi: 10.1109/GLOCOM.2017.8254591.
- [15] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar, “Mobility-Aware Application Scheduling in Fog Computing,” *IEEE Cloud Comput.*, vol. 4, no. 2, pp. 26–35, 2017, doi: 10.1109/MCC.2017.27.
- [16] M. Avgeris, D. Spatharakis, D. Dechouniotis, N. Kalatzis, I. Roussaki, and S. Papavassiliou, “Where there is fire there is smoke: A scalable edge computing framework for early fire detection,” *Sensors (Switzerland)*, vol. 19, no. 3, 2019, doi: 10.3390/s19030639.
- [17] H. Gupta, S. B. Nath, S. Chakraborty, and S. K. Ghosh, “SDFog: A Software Defined Computing Architecture for QoS Aware Service Orchestration over Edge Devices,” no. September, 2016, [Online]. Available: <http://arxiv.org/abs/1609.01190>.
- [18] N. Wang, B. Varghese, M. Matthaiou, and D. S. Nikolopoulos, “ENORM: A

- Framework For Edge NNode Resource Management,” *IEEE Trans. Serv. Comput.*, vol. X, no. September, pp. 1–14, 2017, doi: 10.1109/TSC.2017.2753775.
- [19] S. El Kafhali and K. Salah, “Efficient and dynamic scaling of fog nodes for IoT devices,” *J. Supercomput.*, vol. 73, no. 12, pp. 5261–5284, 2017, doi: 10.1007/s11227-017-2083-x.
- [20] M. Avgeris, D. Dechouniotis, N. Athanasopoulos, and S. Papavassiliou, “Adaptive resource allocation for computation offloading: A control-theoretic approach,” *ACM Trans. Internet Technol.*, vol. 19, no. 2, 2019, doi: 10.1145/3284553.
- [21] S. Cirani, G. Ferrari, N. Iotti, and M. Picone, “The IoT hub: A fog node for seamless management of heterogeneous connected smart objects,” *2015 12th Annu. IEEE Int. Conf. Sensing, Commun. Netw. - Work. SECON Work. 2015*, no. June, pp. 43–48, 2015, doi: 10.1109/SECONW.2015.7328145.
- [22] Z. Wen, R. Yang, P. Garraghan, T. Lin, J. Xu, and M. Rovatsos, “Fog orchestration for internet of things services,” *IEEE Internet Comput.*, vol. 21, no. 2, pp. 16–24, 2017, doi: 10.1109/MIC.2017.36.
- [23] F. van Lingen *et al.*, “The Unavoidable Convergence of NFV, 5G, and Fog: A Model-Driven Approach to Bridge Cloud and Edge,” *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 28–35, 2017, doi: 10.1109/MCOM.2017.1600907.
- [24] Coreos, “kube-prometheus,” *github.com*. <https://github.com/coreos/kube-prometheus>.
- [25] Prometheus, “Prometheus,” *Prometheus.io*. <https://prometheus.io/>.
- [26] Coreos, “prometheus-operator,” *github.com*. <https://github.com/coreos/prometheus-operator>.
- [27] DirectXMan12, “k8s-prometheus-adapter,” *github.com*. <https://github.com/DirectXMan12/k8s-prometheus-adapter>.
- [28] Maravger, “Tensorflow-based Image Classification Service,” *github.com*.

Μελέτη κλιμακωσιμότητας εφαρμογών στα άκρα του δικτύου με τη χρήση του λογισμικού Kubernetes

<https://github.com/maravger/ca-tf-image-classifier>.

[29] Kubespray, “Kubespray,” *github.com*. <https://github.com/kubernetes-sigs/kubespray>.

[30] Kubernetes, “Kubernetes autoscaler,” *github.com*, 2020. <https://github.com/kubernetes/autoscaler/tree/936eea18fb1d4236432fb27edfe90c4f8b2b4f04>.

[31] Google, “Heterogeneous Deployment Patterns with Kubernetes,” *cloud.google.com*, 2020. <https://cloud.google.com/solutions/heterogeneous-deployment-patterns-with-kubernetes>.