

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΑΡΟΥΣΙΑΣΗ / ΕΞΕΤΑΣΗ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ζέρβας Γεώργιος

Μεταπτυχιακός Φοιτητής

Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης

Επόπτης Μεταπτυχιακής Εργασίας: Καθηγητής, Α. Μπίλας

Πέμπτη, 14 Ιουλίου 2022, ώρα 12:00 μ.μ.

Join Zoom Meeting

<https://us02web.zoom.us/j/88233996764>

**“ Genisys, ένας μηχανισμός διαχείρισης πόρων και τοποθέτησης εφαρμογών
υψηλών επιδόσεων κάτω από το σύστημα Kubernetes”**

ΠΕΡΙΛΗΨΗ

Σήμερα οι εφαρμογές του νέφους καθώς και τα φορτία εργασίας υπερυπολογιστικών συστημάτων τείνουν να χρησιμοποιούν διαφορετικές μεθόδους για την διαχείριση των υλικών πόρων πάνω από τους οποίους τρέχουν. Ωστόσο όλο και περισσότερες εφαρμογές χρειάζονται έναν συνδυασμό από βήματα τα οποία αφορούν υπολογιστικές διαδικασίες υψηλών επιδόσεων αλλά και επεξεργασία μεγάλου όγκου δεδομένων. Αυτό έχει σαν αποτέλεσμα, η ενσωμάτωση μηχανισμών διαχείρισης πόρων υπερυπολογιστικών συστημάτων στο υπολογιστικό νέφος να γίνεται ανάγκη. Η διαχείριση πόρων στο νέφος ,τείνει να προτιμάει τον διαμοιρασμό υλικών πόρων σε πολλαπλές εφαρμογές ταυτόχρονα ,προκειμένου να πετυχαίνει υψηλή και αποδοτική αξιοποίηση των υποδομών. Από την άλλη, οι μηχανισμοί διαχείρισης πόρων στα υπερυπολογιστικά συστήματα βασίζονται στην τοποθέτηση των εργασιών σε ουρές προτεραιότητας, ευνοώντας την αποκλειστική χρήση πόρων από κάθε εφαρμογή.

Σε αυτή την δουλειά, σχεδιάσαμε ένα σύστημα το οποίο συνδυάζει τις πλατφόρμες Slurm και Kubernetes. Το σύστημα αυτό είναι σε θέση να τρέξει υπερυπολογιστικά φορτία χωρίς τροποποιήσεις μαζί με άλλες εφαρμογές του νέφους, κάτω από την πλατφόρμα Kubernetes. Πρώτον, μεταφέρουμε όλο το περιβάλλον εκτέλεσης σε συστάδες από κοντέινερς τα οποία ονομάζουμε εικονικές συστάδες. Με αυτόν τον τρόπο κάθε χρήστης ο οποίος τρέχει υπερυπολογιστικά φορτία έχει έναν ιδιωτικό χώρο, ο οποίος περιέχει έναν Slurm καταναμητή εργασιών καθώς και όλα τα εργαλεία

και βιβλιοθήκες που ο χρήστης χρειάζεται. Δεύτερον, σχεδιάζουμε ένα ειδικό πρωτόκολλο επικοινωνίας μεταξύ Slurm και Kubernetes, το οποίο επιτρέπει στον Slurm να αιτείται υλικούς πόρους από τον Kubernetes με δυναμικό τρόπο. Με αυτόν τον τρόπο στο σύστημα μας, ο Slurm controller προωθεί τις αποφάσεις για την τοποθέτηση και κατανομή φορτίων εργασίας στον Kubernetes εξασφαλίζοντας έτσι ένα κεντροποιημένο τρόπο διαχείρισης πόρων. Τρίτον, υλοποιούμε έναν ειδικό κατανεμητή Kubernetes, ο οποίος εφαρμόζει διαφορετικές πολιτικές τοποθέτησης εργασιών ανάλογα με τον τύπο τους.

Αξιολογούμε την απόδοση του συστήματος μας, συγκρίνοντάς το με στατικά διαχωρισμένες συστάδες Kubernetes και Slurm καθώς δείχνουμε την ικανότητα του να τρέχει φορτία διαφορετικού τύπου κάτω από κοινή υποδομή με την λιγότερη δυνατή πτώση της απόδοσης τους.

University of Crete

Computer Science Department

M.Sc. Thesis

Zervas Georgios

Master's Thesis Supervisor: Professor A. Bilas

Thursday, 14 July 2022, 12:00 p.m.

Join Zoom Meeting

<https://us02web.zoom.us/j/88233996764>

**“Genisys, a Resource Management and Placement Mechanism for HPC and
Datacenter applications under Kubernetes”**

ABSTRACT

Today, Cloud and HPC workloads tend to use different approaches for managing resources. However, as more and more applications require a mixture of both high-performance and data processing computation, convergence of Cloud and HPC resource management is becoming a necessity. Cloud-oriented resource management strives to share physical resources across applications to improve infrastructure efficiency. On the other hand, the HPC community prefers to rely on job queueing mechanisms to coordinate among tasks, favoring dedicated use of physical resources by each application.

In this work, we design a combined Slurm-Kubernetes system that is able to run unmodified HPC workloads under Kubernetes, alongside other, non-HPC applications. First, we containerize the whole HPC execution environment into a virtual cluster, giving each user a private HPC context, with common libraries and utilities built-in, like the Slurm job scheduler. Second, we design a custom Slurm-Kubernetes protocol that allows Slurm to dynamically request resources from Kubernetes. Essentially, in our system the Slurm controller delegates placement and scheduling decisions to Kubernetes, thus establishing a centralized resource management endpoint for all available resources. Third, our custom Kubernetes scheduler applies different placement policies depending on the workload type.

We evaluate the performance of our system compared to statically partitioned Kubernetes and Slurm-based HPC clusters and demonstrate its ability to allow the joint execution of applications with seemingly conflicting requirements on the same infrastructure with minimal interference.