ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΑΡΟΥΣΙΑΣΗ / ΕΞΕΤΑΣΗ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Βάρδας Ιωάννης
Μεταπτυχιακός Φοιτητής

Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης
Επόπτης Μεταπτ. Εργασίας: Καθηγητής, Μ. Κατεβαίνης

Τρίτη, 05/11/2019, 15:00
Αίθουσα Β106, Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης

" Βελτιστοποίηση τοποθέτησης διεργασιών και επεκτάσεις για ετερογενή συστήματα στο λογισμικό διαχειρισμού πόρων Slurm"

ΠΕΡΙΛΗΨΗ

Τα υπολογιστικά συστήματα υψηλών επιδόσεων (HPC), στην προσπάθεια τους να ικανοποιήσουν τις συνεχώς αυξανόμενες ανάγκες για περισσότερη απόδοση και υπολογιστικούς πόρους, αναπτύσσονται όλο και περισσότερο σε μέγεθος. Επιπλέον, αξιοποιώντας διαφορετικούς υπολογιστούς πόρους όπως οι "Accelerators" για περαιτέρω αύξηση της υπολογιστικής ισχύος τους γίνονται πιο ετερογενή.
Οι υπολογιστικοί πόροι αυτών των συστημάτων διαμοιράζονται μεταξύ πολλών χρηστών οι οποίοι σε πολλές περιπτώσεις μπορεί να ανέρχονται σε χιλιάδες και να εκτελούν εφαρμογές διαφόρων επιστημονικών πεδίων. Αυτό εγείρει δυο θέματα, το πρώτο είναι η παροχή μιας ποικιλόμορφης της στοίβας λογισμικού που είναι απαραίτητη για τις διαφορετικές εφαρμογές και το δεύτερο είναι η εγγυημένη απομόνωση μεταξύ των διαφορετικών χρηστών. Ένα επιπλέον θέμα είναι και διαχείριση των πολλών διεργασιών των χρηστών καθώς και διανομή των υπολογιστών

πόρων. Επίσης, οι εφαρμογές που επιζητούν περαιτέρω απόδοση βασίζονται σε προηγμένες τεχνικές παραλληλίας για να εκμεταλλευτούν τους πόρους τους συστήματος αυξάνοντας την πίεση στο δίκτυο του συστήματος.
Το πρόβλημα κατανομής των πόρων αυτών των περίπλοκων και μεγάλων HPC συστημάτων στους διαφόρους (πολλούς) χρήστες αντιμετωπίζεται με την χρήση ενός ειδικού ενδιάμεσου λογισμικού που συχνά καλείται Σύστημα Διαχείρισης Πόρων και Εργασιών.
Το θέμα της εγγυημένης απομόνωσης των διαφορετικών χρηστών καθώς και παροχής μίας πιο ειδικά διαμορφωμένης στοίβας λογισμικού συνήθως επιλύεται με την χρήση Εικονικών Μηχανών. Επίσης, η αυξημένη επικοινωνιακή τυπικότητα των εφαρμογών μπορεί να μειώσει το κόστος της επικοινωνίας με αποτέλεσμα την μείωση του χρόνου εκτέλεσης των εφαρμογών καθώς και την μείωση της πίεσης που δέχεται το δίκτυο. Εκτός από το κόστος επικοινωνίας, ο χρόνος εκτέλεσης μπορεί να βελτιωθεί περαιτέρω μέσω της μείωσης του κόστους της διακοπής των MPI εφαρμογών των χρηστών λόγω σφαλμάτων στους κόμβους του συστήματος.

Σε αυτή την εργασία παρουσιάζουμε τρεις επεκτάσεις στο λογισμικό διαχειρισμού πόρων Slurm το οποίο χρησιμοποιείται ευρέως από τα HPC συστήματα. Χρησιμοποιούμε το Slurm και το επεκτείνουμε με βάσεις τις παραπάνω μεθόδους για να επιλύσουμε τα προβλήματα που αναφέραμε. Η πρώτη επέκταση προσφέρει στο Slurm την δυνατότητα να υποστηρίζει FPGA-based accelerators καθιστώντας το Slurm καταλληλότερο για ετερογενή συστήματα. Το Slurm παρέχει ήδη υποστήριξη για GPUs όχι όμως για FPGA-based accelerators. Η επόμενη επέκταση που παρουσιάζεται προσδίδει την δυνατότητα στο Slurm να εκτελεί εικονικές μηχανές και μέσα στα εικονικά περιβάλλοντα αυτά να εκτελεί τις διεργασίες των χρηστών. Σε σύγκριση με παρόμοιες δουλείες η εν λόγω επέκταση προσφέρει ένα πιο απλό περιβάλλον για τον χρήστη καθώς και την δυνατότητα διαχείρισής των εικονικών μηχανών από το Slurm.
Η τελευταία επέκταση υλοποιεί την προτεινόμενη προσέγγιση για βελτιστοποίηση τοποθέτησης των διεργασιών με βάση την τοπολογία και παράλληλα λαμβάνοντας υπόψιν τους κόμβους που παρουσιάζουν σφάλματα. Σκοπός αυτής της προσέγγισης είναι να μειώσει το επιπλέον κόστος της επικοινωνίας των παράλληλων διεργασιών μιας MPI εφαρμογής αλλά και να μειώσει το επιπλέον κόστος της διακοπής εφαρμογών λόγω σφαλμάτων στους κόμβους.
Η προτεινόμενη προσέγγιση ακολουθεί μια γνωστή μέθοδο με βάση την υπάρχουσα βιβλιογραφία κατά την οποία το πρόβλημα της τοποθέτησης των διεργασιών στους υπολογιστους πόρους μοντελοποιείται ως ένα γραφοθεωρητικό πρόβλημα. Η τοπολογία του συστήματος και καθώς και το μοτίβο επικοινωνίας της εφαρμογής αναπαρίστανται ως δυο γράφοι. Με την χρήση της βιβλιοθήκης Scotch, η οποία λύνει το γραφοθεωρητικό πρόβλημα, παράγεται η αντιστοιχία των διεργασιών στους υπολογιστικούς πόρους του συστήματος.
Επιπλέον, η προτεινόμενη προσέγγιση ενημερώνεται για τους κόμβους που πιθανώς να παρουσιάσουν σφάλματα ώστε να αποφύγει την χρήση τους καθώς και τα μονοπάτια που επηρεάζουν.

Τέλος, αξιολογούμε την προτεινόμενη προσέγγιση προσομοιώνοντας πραγματικές εφαρμογές σε δύο διαφορετικά εικονικά περιβάλλοντα με και χωρίς σφάλματα. Τα αποτελέσματα δείχνουν σημαντική μείωση του χρόνου εκτέλεσης των εφαρμογών και στα δύο εικονικά περιβάλλοντα. Η προτεινόμενη προσέγγιση καταφέρνει να μειώσει σημαντικά τον αριθμό των απορριπτόντων διεργασιών λόγω σφαλμάτων καθώς τον χρόνο εκτέλεσης σε σχέση με την βασική προσέγγιση του Slurm από 10% έως 31%.

Ioannis Vardas
M.Sc. Thesis

Computer Science Department
University of Crete
Master's Thesis Supervisor: Professor, M. Katevenis

Tuesday, 05/11/2019, 15:00
Room B106, Computer Science Dept., University of Crete

"Process Placement Optimizations and Heterogeneity Extensions to the Slurm Resource Manager"

ABSTRACT

HPC systems keep growing in size to meet the ever-increasing demand for performance and computational resources. Additionally, by utilizing diverse resources such as accelerators to further improve their computational power the HPC systems become more heterogeneous. The resources of such systems are shared among many users, whose number can reach up to a few thousands, and execute a broad spectrum of applications from all scientific fields. This brings up two issues, the first is the diversity of the software stack required by the various applications and the complete platform isolation among the users. Another issue is the managing of the many resources and the various jobs issued by the users.

Additionally, applications that seek to lower their completion time rely on advanced parallelism for exploiting the system's resources, which leads to increased pressure on system interconnects. The problem of managing the resources of a complex and sizeable HPC system is tackled by the use of a special middleware often called Resource and Job Managing System.

In order to deal with the issue of platform isolation and provide a more flexible software stack the Virtual Machine or container technology is often employed. Furthermore, by increasing communication locality of the applications their communication overhead is reduced resulting in lower completion times which can also lower the pressure on system interconnects. Apart from the communication cost, the completion time of an application can be further improved by reducing the overhead of job abortions due to node failures.

In this thesis we present three extensions to Slurm which is a Resource and Job Managing System that is widely adopted in HPC systems that use the above methods to tackle the aforementioned issues. The first extension provides Slurm with support for FPGA-based accelerators that allows the users to select nodes with specific FPGA-based accelerators making Slurm better suited for heterogeneous environments. The second extension enables Slurm to run workloads in Virtual Machines. Compared to other similar approaches this extension to Slurm maintains a simple user interface and also integrates the VM management into Slurm.

The final extension implements a topology and fault aware process placement approach that reduces the communication cost while also taking into account node failures. The proposed topology and fault aware process placement approach follows a common approach which, according to the bibliography, models the process placement as graph mapping or graph embedding problem. Both the system's topology and the application's communication pattern can be expressed as two separate graphs and the mapping is derived by solving the corresponding graph mapping problem.

Additionally, unlike the common approach the proposed approach also takes into account node failures and attempts to avoid paths that include faulty nodes. Finally, we evaluate the proposed topology and fault aware approach in two different environments, with and without the presence of node failures. The results of our evaluation show a notable decrease in overall completion time of MPI jobs in both environments. Compared to the default process placement of Slurm, the proposed approach reduces significantly both the MPI job abortions and the overall completion time from 10% up to 31% for different MPI applications.