ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΑΡΟΥΣΙΑΣΗ / ΕΞΕΤΑΣΗ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Κατσογριδάκης Παύλος

Μεταπτυχιακός Φοιτητής

Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης

Επόπτης Μεταπτ. Εργασίας: Καθηγητής Άγγελος Μπίλας

Πέμπτη, 10/11/2016, 13:00

Αίθουσα B108, Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης

" Εκτέλεση αναδρομικών ερωτημάτων στο Apache Spark"

ΠΕΡΙΛΗΨΗ

Τα περιβάλλοντα MapReduce επιτρέπουν την επεξεργασία τεράστιου όγκου δεδομένων με το να περιορίζουν το προγραμματιστικό μοντέλο σε τελεστές map και reduce. Αυτό το επίπεδο αφαίρεσης απλοποιεί πολλά δύσκολα προβλήματα που προκύπτουν στα κατανεμημένα συστήματα, όπως το συγχρονισμό και την ανοχή σε σφάλματα, και τα κρύβουν από τον προγραμματιστή. Παρ' όλα αυτά, υπάρχουν αλγόριθμοι οι οποίοι δεν μπορούν να εκφραστούν εύκολα σε MapReduce, όπως οι αναδρομικοί αλγόριθμοι.

Στην εργασία αυτή επεκτείναμε το Apache Spark (ένα σύστημα χρόνου εκτέλεσης MapReduce), ώστε να υποστηρίζει αναδρομικούς αλγορίθμους. Οι αναδρομικοί αλγόριθμοι MapReduce δημιουργούν μεγάλο αριθμό εργασιών, οι οποίες δυσκολεύουν το πρόβλημα της χρονοδρομολόγησης. Γι` αυτό εισάγουμε ένα νέο παράλληλο και πιο ελαφρύ αλγόριθμο χρονοδρομολόγησης. Ο αλγόριθμος αυτός είναι κατάλληλος για χρονοδρομολόγηση ενός μεγάλου αριθμού από εργασίες οι οποίες παίρνουν πολύ λίγο χρόνο. Υλοποιήσαμε τον παραπάνω αλγόριθμο και βρήκαμε ότι απλοποιεί την έκφραση

αναδρομικών ερωτημάτων, και παράλληλα μπορεί να πετύχει μέχρι 2,5 φορές καλύτερο χρόνο από τον ήδη υπάρχων αλγόριθμο του Spark σε κάποια είδη εργασιών.

Katsogridakis Pavlos
M.Sc. Thesis
Computer Science Department
University of Crete
Master's Thesis Supervisor: Professor A. Bilas

Thursday, 10/11/2016, 13:00
Room B108, Computer Science dept., University of Crete

"Execution of Recursive Queries in Apache Spark"

ABSTRACT

MapReduce environments offer great scalability by restricting the programming model to only map and reduce operators. This abstraction simplifies many difficult problems occurring in generic distributed computations like fault tolerance and synchronization, hiding them from the programmer. There are, however, algorithms that cannot be easily or efficiently expressed in MapReduce, such as recursive functions. In this work we extend the Apache Spark runtime so that it can support recursive queries. Those queries produce a very large number of tasks, making scheduling a difficult and time consuming problem.

To tackle this problem we also introduce a new parallel and more lightweight scheduling mechanism, ideal for scheduling a very large set of tiny tasks. We implemented the aforementioned scheduler and found that it simplifies the code for recursive computation and can perform up to 2.5 times faster than the default Spark scheduler for certain kinds of benchmarks.