Contribution ID: **11** Type: **Poster**

INCREASING CLUSTER COMPUTING PERFORMANCE THROUGH A DYNAMIC LOAD REARRANGEMENT

Monday, 11 April 2011 09:00 (8 hours)

Overview

The increase of the computational power leads necessarily to more complex approaches in the resources exploitation and management. One of the main problems nowadays for a computing center is given by the under exploitation of the available resources.
It may happens that in a heterogeneous batch queue system, available for both serial single core processes and parallel multi core jobs, one or more computational nodes of the cluster are serving a number of jobs lower than their actual capability.
A typical case is represented by more single core jobs running each one over a multi core server, while more parallel jobs - requiring all the available cores of a host - are queued. A job displacement, executed at runtime in order to stack up more single core processes over a single multi core server, is able to free extra resources able to host new processes - both single or multi core.
We present an efficient method to improve the computing resources exploitation.

Impact

In case of more single core processes running each one over a different host, the migration of the spread out jobs to the lowest number of computational nodes may allow to free a large number of otherwise busy hosts. In order to ensure a full hosts exploitation, and prevent at the same time the overload of one or more nodes in the cluster, the job migration takes place only under certain conditions. We also paid a special attention to avoid a too frequent job displacement, damaging the global performance: the system may act a job transition at scheduled time interval and under specific threshold conditions.
A secondary effect, probably not less appealing by the point of view of the “green computing”, is represented by the power efficiency improvement through a dynamic job rearrangement, with an energy saving up to 90% in some particular cases - more frequent than expected.
By the use of a remote controlled power supply, it is possible to switch off the unused hosts, waiting to be switched-on at request.

Description of the work

The idea is to pile up the maximum number of jobs over the minimum number of hosts, compatibly with the available CPU and memory on the single hosts, in order to fill as many contiguous job slots as possible. This way the running jobs do not suffer any performance loss, and at the same time the farm may gain contiguous job slot able to host new parallel multi-core jobs otherwise stuck in queue.
A prototype of job mover has been developed with the aim of freeing the best part of otherwise unavailable resources in a computing cluster. We started implementing a batch system and queue simulator, in order

to test the efficiency of several job rearrangement algorithms. Defining an exploitation parameter, strictly connected to the cluster load, we implemented two algorithms able to increase the computational resources load - limited for each server by the number of available cores.
The problem of the possible permutations achieved by moving a set of jobs, each one requiring a variable number of core, over a set of server is described by an NP-complete complexity class. Due to the difficulty in finding the best solution, we focused on searching a solution able to improve the current load status of the cluster, certainly not the optimum one.
The cluster and queue simulator may also be used to test other rearrangement algorithms, in order to achieve an even better result in the cluster exploitation.
To take advantage of the features provided by the job migration system, the only requirement is the job checkpoint capability - a complete disk and memory dump is needed for a job freeze and its immediately subsequent restart on another host. This capability is today guaranteed from the major batch queue system available: PBS, LSF, SGE.

Conclusions

The system, developed at Scuola Normale Superiore, in collaboration with the Computer Science Engineering Department at the University of Pisa [Italy], is able to provide an increase in the number of running jobs. The increase is from 15% in case of heterogeneous single and multi core running processes, to 90% in case of long term single core jobs running over several different hosts, and a large number of parallel multi core processes - requiring the entire number of processors of a single computational node - stuck in queue.

Primary authors: Dr CALZOLARI, FEDERICO (Scuola Normale Superiore - INFN); Dr VOLPE, SILVIA (Computer Science Engineering Department, University of Pisa [Italy])

Presenter: Dr CALZOLARI, FEDERICO (Scuola Normale Superiore - INFN)

Session Classification: Posters

Track Classification: Poster