Thesis Overview

Performance Evaluation of Applications for Heterogeneous Systems by means of Performance Probes

Alexandre Otto Strube*
Computer Architecture and Operating Systems Department,
Universitat Autònoma de Barcelona, Spain
PhD Thesis in High Performance Computing.
Advisor: Emilio Luque**
alexandre.otto@uab.es*, emilio.luque@uab.es**

This work describes a novel way to select the best computer node out of a pool of available potentially heterogeneous computing nodes for the execution of computational tasks. This is a very basic and difficult problem of computer science and computing centres tried to get around it by using only homogeneous compute clusters. Usually this fails as like any technical equipment, clusters get extended, adapted or repaired over time, and you end up with a heterogeneous configuration.

So far, the solution for this, was:

- To leave it to the users to choose the right node(s) for execution, or
- To make an extensive test battery, executing and measuring all possible tasks on every type of computing node available in the pool. In the typical case, where a large number of tasks would need to be tested on many different types of nodes, this could use a lot of computing resources, sometimes even more than the actual execution one wants to optimize. History of previous executions is a variant.
- Run a series of synthetic benchmarks that would create an abstract performance model of the available nodes.

In a specific situation (hierarchical multi-clusters - see figure 1), the situation is worse, as the configuration of the cluster changes over time, so that the execution tests would have to be done over and over, every time the configuration of the cluster is changed.

I developed a novel and elegant solution for this problem, named "Performance Probe", or just "Probe" for short.

These Probes are created by running the application to be characterized, discover, via instrumentation, its performance-wise relevant phases of the execution, and saving them in the form of checkpoints. This menas that, a probe is a striped-down version of a task that includes all important performance characteristics of the original, but can be executed in a much shorter time (seconds, instead of hours), is much smaller than the original task (about 5% of the original size in the worst cases), but allows to predict the execution time of the original within reasonable bounds (around 90% accuracy).

So, instead of running the application as a whole or benchmarks that may or may not represent any application performance at all, one runs a probe for some seconds and the probe will tell how long the whole execution of the application would take.

These results are very important: as scheduling is a basic problem of computer science, these results cannot only be used in the setting described by the thesis (of setting the right compute node for tasks in a hierarchical multi-cluster), but can also be applied in many different contexts every time scheduling and/or selection decisions have to be made: selecting where a computational task would run most efficiently (which cluster at which centre); picking the right execution nodes in a large complex (grid, cloud), workflows and many more.

This research started in order to complement previous research of [2] on the multi-cluster model, where the performance of the nodes was left out of the analytical model to be determined.

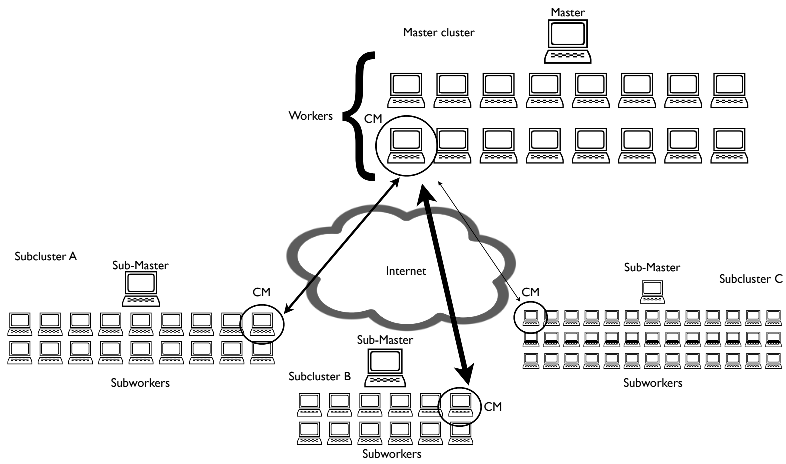

Figure 1: A hierarchical multi-cluster environment. Different machine sizes represent different processing power. Thickness of the arrows represent different network bandwidths.

Initial research demonstrated that the checkpoints' sizes made the Probes too big to be transported over the internet in feasible time. Further research reduced the Probes to a fraction of their original sizes, making them small enough to be transported to remote clusters in order to characterize their machines in viable time.

One characteristic of the current Probe model is that the prediction depends on the original application's input data. If the application is data-dependent, so will the Probe be. There are two aspects of this should be observed:

- Even when a data-dependent application yields wildly different results from one execution to another with different data sets of equivalent sizes, the Probes still gives good performance information to the scheduler: the same Probe while run in different machines gives different results proportional to those of the application.
- Being the Probes as fast as they are, it is possible to construct a probe collection, that correlates to the spectrum of possible performances. There is ongoing research on characterizing these different input data sets started by Fritzsche, Rexachs and Luque [4], and the union of both lines of research is natural.

Probes are useful for any scheduling logic that needs information about how a given computing element will perform while running an application can benefit from the use of Probes. Selection algorithms such as [1] and [3] can be simplified with the performance prediction given by our Probes. Environments such as grids and clouds can use our Probes directly to replace the "benchmark" parameter of the schedulers, as it is way more related to the real application. Different application models, such as workflows may also benefit from Probes, as we may run Probes for every node of the workflow on every machine available, and run the workflow nodes where they run best.

References

[1] I Al-Furiah, S Aluru, S Goil, and S Ranka. Practical algorithms for selection on coarse-grained parallel computers. *IEEE Transactions on Parallel and Distributed Systems*, Jan 1997.

[2] Eduardo Argollo and Emilio Luque. Performance prediction and tuning in a multi-cluster environment. *Ph.D. Dissertation*, Oct 2006.

[3] J Barbosa, J Tavares, and A.J Padilha. Linear algebra algorithms in heterogeneous cluster of personal computers. *hcw*, 00:147, 2000.

[4] P. Fritzsche. Podemos Predecir en Algoritmos Paralelos no Deterministas? *Ph.D. dissertation*, 2007.