

Analysis of High Performance Applications Using Workload Requirements

Mariza Ferro^(✉), Giacomo Mc Evoy, and Bruno Schulze

National Laboratory of Scientific Computing, Petrópolis, Brazil
{mariza,giacomo,schulze}@lncc.br

Abstract. This short paper proposes two novel methodologies for analyzing scientific applications in distributed environments, using workload requirements. The first explores the impact of features such as problem size and programming language, over different computational architectures. The second explores the impact of mapping virtual cluster resources on the performance of parallel applications.

Keywords: HPC · Scientific computing · Performance prediction · Virtualization

1 Introduction

High Performance Distributed Computing is essential to improve scientific progress in many areas of science and to efficiently deploy a number of complex scientific applications. Also, the efficient deployment of High Performance Computing applications on Clouds offers many challenges, in particular, for communication- intensive applications. Benchmarks are good for comparisons between computational architectures, but they are not the best approach for evaluating if an architecture is adequate for a set of scientific applications. In this paper, we discuss two methodologies for evaluating the impact of the underlying infrastructure on observed performance, both from physical and virtual perspectives. The first methodology begins on scientific application characteristics, and then considers how these characteristics interact with the problem size, with the programming language and finally with a specific computational architecture. The second methodology focuses on the case of distributed applications in virtual clusters by analyzing the impact of different VM profiles and placements.

2 Methodology Based on Requirements

In this methodology, the performance evaluation is made considering the characteristics of the applications that will be used in the HPC infrastructure, under conditions as real as possible. It was developed based on Operational Analysis (OA) concepts [5] from where we extract the systematic model to evaluate complex systems and to provide a decision-making process to rationally choose an

architecture. Also, it was made a study about the requirements of the scientific applications, based on applications classes named Dwarfs [1]. These classes represent the behavior in terms of computational requirements. These requirements were studied, modeled and a set of parameters were defined for the methodology (Essential Elements of Analysis - EEA).

The methodology comprises a set of phases and respective steps, briefly described next. All phases and steps of the methodology are detailed in [2].

2.1 Description of Methodology Phases

The first phase is the *Definition Problem* in which the real problem and the objective of the methodology application are clearly defined. In sequence, the phase *Problem Detailing Analysis* details the user problem, searching the complete definition of requirements. It is very important here the knowledge acquired about each application, focus of the evaluation: the real problem sizes/workload executed, programming languages, applications executed sequentially or in parallel, etc. Further, the relative importance of each one is defined in a subjective way by researchers and converted in a set of numerical weights by means of Analytic Hierarchy Process (AHP). Beyond those critical issues, the Measures of Effectiveness (MOEs) and EEA are defined. A MOE of a system is a parameter that evaluates the capability of the system to accomplish its assigned goal under a given set of conditions. The *implementation* phase is where the test planning is completed, based on both aforementioned phases. The methodology endorses that the real application and workloads must be used for performance evaluation, enabling an evaluation as real as possible. However, we know that it is not always possible, for example by confidentiality or software licenses. So, in this case the real applications are mapped to a Dwarf class. The model for mapping applications to Dwarf comprises a set of rules that enable us to define the class of an application based on the EEA measured under the execution tests. Based on the classification of each application, one or more benchmarks are defined to be executed as evaluation test. The last phase is *Communication of Results*, in which data collected on tests are confronted with MOEs and the data from different providers are compared. For this phase it was developed a Gain Function (GF) that enables the decision based on quantitative and qualitative parameters about the problem of the researcher. Using MOEs and the GF, it is possible to define the operational effectiveness and suitability of the infrastructure. The GF is briefly described in Eq. 1 [3].

$$G(k) = w_d \sum_{j=1}^n w_j D(j, k) + w_c C_{E_k}, k = 1, \dots, m \quad (1)$$

For each application j , $j = 1, \dots, n$, on each evaluated infrastructure E_k , $k = 1, \dots, m$, the execution time $t(j, k)$ is measured. For each application j it is assigned a weight w_j . Also, for each architecture is considered its cost c_k . Let w_c and w_d be the weights for cost and performance. From those operational values, the GF enables to consider the performance (execution time) of each scientific application for each architecture evaluated.

3 Multi-dimensional Analysis on Virtual Clusters

This methodology proposes the utilization of Canonical Correlation Analysis (CCA) to find optimal virtual cluster settings of an application, accounting for its communication pattern. It is built upon three sources of information:

1. Characteristics about how the virtual cluster is defined and deployed;
2. Characteristics of the performance of the target application;
3. Characteristics about the nature of the workload using Dwarfs.

Extracting Characteristics: The Cluster Placement [4] was proposed to address the limitations of current descriptions of virtual clusters. Most representations focus solely on the dimensions of the virtual cluster. These elements can be directly observed by a parallel application running on the cluster. With our proposed model, it is not only possible to determine which VMs execute on which physical machine, but also know how each virtual core is mapped to underlying hardware using virtual core pinning (or lack thereof). This enriched information allow us to map virtualization characteristics to performance more effectively.

In order to understand the effect that Cluster Placement exerts on the performance of an application, we developed the VESPA (Virtualized Experiments for Scientific Parallel Applications) framework that manages the systematic execution of the application along several scenarios with different Cluster Placements. Executions were performed in a controlled environment to isolate resulting variability to characteristics of the Cluster Placement. The framework registers a series of performance metrics to be related to each execution, both (i) user-centric (runtime, application/kernel time, application-specific); (ii) system-centric (physical/virtual CPU and network utilization).

Mapping Characteristics to Performance: The nature of the workload is extracted by an equivalence to one of the Dwarfs [1], and at least one representative benchmark. The representative benchmarks are executed beforehand over several possible Cluster Placements (hundreds), and the relevant metrics are gathered, thereby creating a performance matrix.

For a given target application, a series of Cluster Placements (in our experience, at least 40) are proposed to create an initial profile for the application over virtualized environments. CCA enables us to find relationships between the datasets of the target and the representative application of the corresponding Dwarf. Within the space obtained through dimensionality reduction, we find linear regressions between performance and placement, and therefore we can predict performance for new placements using interpolations. For the Structured Grid Dwarf, we obtained accuracy higher than 90% in performance prediction, when at least 50 data points are known.

4 Summary

The methodology based on applications requirements can assists researchers to define what is the best to solve their set of scientific applications. The

methodology enables to define representative evaluation tests, including a model to define a representative benchmark, when the real application could not be used. Also, the GF allows a decision-making based on the performances of a set of applications and architectures and its relative importance. We made a case study for bioinformatics applications, in which some steps are detailed and where the methodology proved to be useful and relevant [3].

The proposed methodology based on Cluster Placement and VESPA was helpful in understanding how latency effects can be minimized by carefully constructing virtual clusters. The relationship between performance and Cluster Placement appears non-linear and complex, but by using CCA we were able to find linear relationships between two sets of relationships, enabling reasonably accurate predictions. The accuracy seems to depend on the type of Dwarf, whereby applications with higher frequency of communication are more difficult to predict.

References

1. Asanovic, K., Bodik, R., Demmel, J., Keaveny, T., Keutzer, K., Kubiawicz, J., Morgan, N., Patterson, D., Sen, K., Wawrzynek, J., Wessel, D., Yelick, K.: A view of the parallel computing landscape. *Commun. ACM* **52**(10), 56–67 (2009). <http://doi.acm.org/10.1145/1562764.1562783>
2. Ferro, M., Mury, A.R., Schulze, B.: Manual de metodologia de análise operacional de sistemas de computação científica distribuída de alto desempenho. Relatórios de Pesquisa e Desenvolvimento do LNCC 01/2015, Laboratório Nacional de Computação Científica, Petropolis (2015). www.lncc.br
3. Ferro, M., Nicolás, M.F., del Rosario, Q., Saji, G., Mury, A.R., Schulze, B.: Leveraging high performance computing for bioinformatics: a methodology that enables a reliable decision-making. In: 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2016, Cartagena, 16–19 May 2016, pp. 684–692. IEEE Computer Society (2016)
4. Mc Evoy, G., Porto, F., Schulze, B.: A representation model for virtual machine allocation. In: International Workshop on Clouds and (eScience) Applications Management - CloudAM 2012. 2012 IEEE/ACM Fifth International Conference on Utility and Cloud Computing (2012)
5. Wagner, D., Mylander, W., Sanders, T.: *Naval Operations Analysis*, 3rd edn. Naval Institute Press, Annapolis (1999)