

An Image Processing-based Test Bench for Performance Evaluation in Hybrid Clouds

Rafael D. C. Santos

LAC/CTE

Brazilian National Institute for Space Research

São José dos Campos, SP, Brazil

rafael.santos@inpe.br

Bernady O. Apduhan

Faculty of Information Science

Kyushu Sangyo University

Fukuoka, Japan

bob@is.kyusan-u.ac.jp

Abstract—Hybrid clouds are combinations of private, community and/or public clouds, and allow the creation of “cloud bursting” deployment models, in which part of the data processing is done in a private cloud and resources from public clouds are used when needed. This model is particularly interesting for CPU-intensive scientific data processing. In order to help determine in which conditions part of the whole of a large data processing task must be sent to a public or private cloud, dynamic benchmarks are required. In this paper we propose an image processing-based test bench that monitors the performance of the clouds resources to gather useful statistics and to determine which resource should be used for a specific task, in order to reduce costs and running time.

Keywords—Hybrid Clouds, Test Bench, Distributed Image Processing

I. INTRODUCTION

Cloud computing is a model of distributed computing that consists of shared computing resources (servers, storage, applications, networks, services, etc.) that are made remotely available to external users so these can use the services on an on-demand basis [1].

Additional advantages of the cloud computing model are the possibility of reducing costs associated with IT infrastructure, better use of underutilized computing resources, transparent replication of the services and portability over other clouds (that could lead to better manageability and reliability), among others.

There are three service models for cloud computing:

- *Infrastructure as a Service (IaaS)* is the most basic model, in which users create virtual machines and associated resources in the cloud, and must deploy all the applications on this virtual machine.
- *Platform as a Service (PaaS)* is an intermediate model, in which users can deploy applications on virtual machines that already have basic software installed (e.g. operating system, databases, web servers, application servers).
- *Software as a Service (SaaS)* is a model in which the user does not have the need to install software at all, besides his/her application, on a designed application

server. In this model users do not need to get involved with the management of the applications’ infrastructure.

Cloud computing environments can be also divided into four categories, according to the deployment method:

- *Public Clouds* are usually maintained by a large organization with resources that are made available to the general public, in a free or pay-per-use model. Infrastructure costs are the sole responsibility of the cloud maintainer.
- *Community Clouds* are used by groups of organizations to provide a pool of shared resources that are typically related to the organizations’ members needs. Infrastructure costs are shared by the organizations’ members.
- *Private Clouds* are pools of computational resources made available by and for a single organization. Alternatively private clouds may be managed by a third-party provider but the resources are dedicated to a single organization. Private clouds may be hosted internally on that organizations’ or externally.
- *Hybrid Clouds* are models that are composed by two or more clouds that allows different deployment approaches. An interesting deployment model that can be achieved with hybrid clouds is called cloud bursting, in which an application may use resources of a private cloud, “bursting” into public clouds when there is a demand for more computing or storage. This bursting is often temporary, so this model allows an organization to maintain reasonable resources for its daily usage, paying for the public cloud resources only when required.

Cloud computing can be considered a mature technology, with several commercial offerings hosting many applications that varies in size and scope [2].

We are interested in scientific applications that may use the cloud as the deployment environment. The reasons to do scientific computing on clouds are the same used by commercial computing: reducing costs and improving availability and scalability by sharing resources and centralizing the infrastructure.

We consider that for some categories of scientific comput-

ing hybrid clouds, composed by a private, on-site cloud with connections to external clouds (either community or public) may be the best approach when considering feasibility and cost-benefit, since most scientific applications may fall under two main general categories: data dissemination applications and data-intensive applications.

Data dissemination applications are applications that allow users to collect and query scientific data over the Internet, with some variants allowing some off-line processing of the data for specific purposes. These applications must be always online and are considered generally of low impact on computational resources.

Data-intensive applications, on the other hand, require large amounts of computational resources, often processing very large amounts of data in order to extract information from it. These applications' interfaces may be public, allowing participants from outside the hosting institution to run experiments on the data.

Some applications, which are the main point of interest for our research, may be considered a mixture of these two models: large amounts of data are processed offline and the results of this processing are available online, but are subject to reprocessing, processing with new algorithms, enrichment with other data sources, etc. in such a way that there is the need for both an always-on data dissemination solution and a sporadic, high-throughput data processing solution, which conceptually matches hybrid clouds deployment models.

Some examples of hybrid scientific applications are the Sloan Digital Sky Survey, which sporadically processes a very large amount of astronomical images to extract objects and store them on databases accessible by the public [3], [4]; 2DPhot, a similar image processing pipeline with different algorithms [5]; the Brazilian Space Weather Program, which monitors signals from the ionosphere and create several different types of views of the data [6]; the Brazilian Weather Forecast and Climate Studies Center (CPTEC) data site (<http://bancodedados.cptec.inpe.br/>), which allows the retrieval of historical time series from several different observed and simulated weather and climate data [7].

At the moment those applications are not hosted on clouds but on dedicated servers, with some studies [8] suggesting that migrating the whole applications and databases to a public cloud may not be feasible. We assume, nonetheless, that a hybrid cloud deployment may be viable for some of those applications, with processed data and services hosted by a private cloud and heavy-duty data processing performed on-demand in a community or public cloud.

In a deployment of a scientific, data intensive application in a hybrid cloud, several factors may determine whether a data processing task will be executed in a local workstation or cluster, in a virtual machine on a private cloud or bursted to a public or community cloud: the size of the data processing task (both related to the cost of transmitting data to and from a public cloud and related to the estimated

number of CPU cycles); the estimated advantages of doing so; privacy and disclosure issues, etc.

While some of those factors can impact definitively on the decision to run locally or remotely (for example, data size and privacy), in most cases other factors must be considered, and some of those cannot be determined statically. For example, the bandwidth required to transmit data between a local computer and a external cloud may suffer from sporadic variations; or a virtual machine in a cloud may provide less performance than configured for a number of reasons.

When deploying a scientific data processing task, in part or fully, on the scenarios that we're considering, it is important to have a simple metric that could quickly give an estimate of the cost/benefit of running it locally, in a private cloud or in a public cloud. This cost/benefit measure is very complex, since it depends on the network load, virtual machines availability and financial cost, but we assume even a quick and simple performance estimation measure can help the decision to execute locally or remotely.

In this paper we propose a test bench for a hybrid cloud that could help collect and analyze some features of the resources available to that cloud. These features could help decide on whether to execute a specific task locally (in a private cloud readily accessible to the final user) or in other private, community or public clouds. The test bench is based on simple reactive software agents and uses some basic image processing operations to measure performance in each resource available to the user.

This paper is divided into the following sections: this introduction presented the motivation for using a hybrid cloud for scientific applications and the expected benefits. Section II describes the details of the proposed test bench, including the underlying technologies (agents and image processing). Finally, section III presents some considerations on the development and deployment of such a test bench, the present status and future work.

II. ARCHITECTURE OF THE TEST BENCH

As mentioned in the introduction, when deploying tasks for processing over a hybrid cloud, it is important to have a simple measure that could help determine which resource in the hybrid cloud will be used for a particular task.

One simple way to get a quick estimate of the cost of executing a task locally or remotely is through monitoring the measurable variables that may impact on the execution of the task before submitting the task. The data collected by monitoring can be used for two purposes: quick decision on which execution environment will be chosen for a specific task (remote or local) and medium- to long-term evaluation of the available resources so one could determine measures to improve the general performance of the available infrastructure.

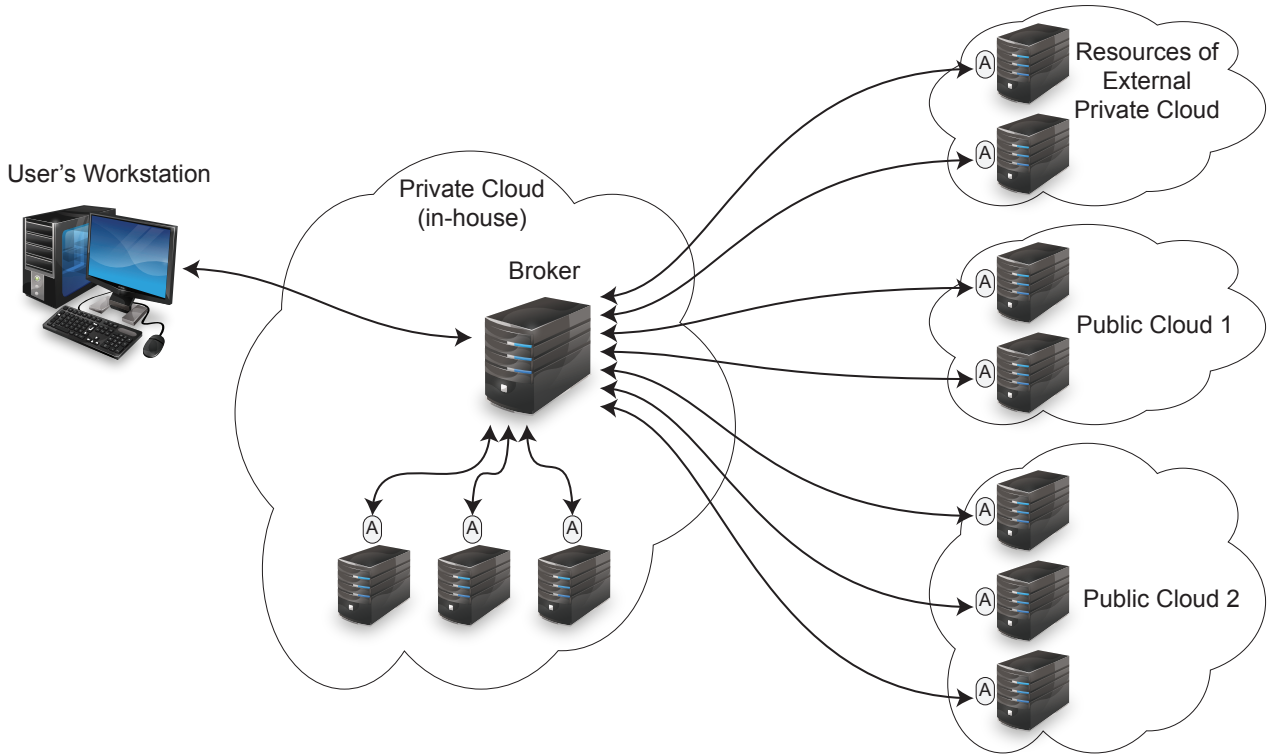


Figure 1. A Test Bench Architecture

In this paper we propose an image processing-based test bench that can be used to evaluate the responsiveness of a distributed computing system composed by local computers, a local private cloud and one or more external community or public clouds. Image processing operations were chosen since most of its operations are realistic examples of the common scientific data processing pipelines (CPU-intensive algorithms that can be applied in parallel to a dataset composed of independent data chunks).

Figure 1 shows the basic architecture for the test bench. The central component of it is a **broker**, an application server that is part of a private cloud that can be accessed from a user's workstation and which is able to connect with several simple software agents, hosted in every resource on the public and private clouds that are available to the user (identified by a circled "A").

The broker can execute several different but integrated tasks in the proposed test bench. Some of those are:

- Requesting information about the performance status in each resource of the clouds (*test mode*): the broker sends a signal to all agents so these will execute a simple, pre-programmed task, and record the timestamps for the different steps on this execution: sending and receiving the request, running the task, sending and receiving the response (Figure 2). Data collected by

this task will serve as a "ping"-like function to monitor the responsiveness of a resource.

- Executing a processing task in one of the resources of the clouds (*run mode*): this task is similar to the test mode task just described; but real data is sent to the resource for processing. All timestamps for the different steps will be recorded as well.
- Storing, tabulating and reporting on the results of execution of tasks in both test and run modes. Special treatment will be done regarding data collected from executing tasks in run mode, since these will be dependent on the amount of data to be transmitted and processed.
- Running data analysis algorithms on the execution of the tasks (i.e. timestamps for each step in each task). These algorithms may help identify best-fit resources for particular tasks, time-related issues with the resources (e.g. periods of low or high network congestion) and shed light on the dynamic nature of the whole infrastructure of the hybrid cloud.
- Suggesting one of the available resources for the execution of a particular task. This suggestion may be done based on the most recent information from a particular resource or on the analysis of performance of a resource over a medium to large period of time.

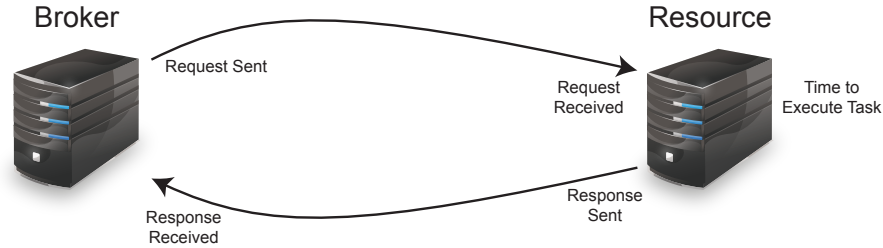


Figure 2. Data collected when the broker runs a monitoring task

It is expected that this suggestion may lead to a better performance of the data analysis system as a whole and to the distributed data processing policy in general.

The tasks that the broker must be able to execute are based on two technologies: reactive software agents and image processing, described in the next subsections.

A. Reactive Software Agents

Software agents are software systems that, in general, are able to execute tasks on behalf of others (users or agents). While this definition is one the simplest, and many authors expect software agents to exhibit other properties (autonomy, intelligence, mobility, learning) [9], we still consider as agents the software components on our test bench that will be deployed to the available resources in the clouds used on our setup.

These agents will be implemented as web services running on a local server on the clouds' resources, and may be able to execute basic tasks and send and receive data from the broker. Agents may also be able to monitor other environmental variables from its host, but these are not being considered at the moment.

These agents will be eminently reactive; performing tasks only when those are initiated by the broker; and with minimum proactivity (e.g. an agent may chose to terminate a process it is running when some conditions are reached, e.g. a timeout condition). The agents will also not be able to learn about the execution of its tasks, since learning, as a general function, may be done by the broker itself, which may be able to learn the best fit resources for a particular task.

On the proposed architecture, the agents are static, i.e. they already have a set of known operations they can execute, meaning that all timestamp collection and image processing algorithms must be coded beforehand and may not change after deployment. There are ways to deploy web services dynamically, which could add flexibility to our test bench and to the expected applications, that are under study [10].

B. Image Processing

In order to simplify the study and development of the test bench, at this point all data processing tasks that may be submitted to the hybrid cloud will be related to image-processing, which are related to the scientific data processing tasks categories being considered and commented in Section I.

Image processing is a research and development field that has applications in many fields of human endeavor [11]. Research in image processing is still in high demand, and there are some research topics that are highly related to high-performance computing due to the number and size of images that can be collected by sensors such as digital cameras, telescopes, microscopes, satellites, etc.

There exists several different image processing operations that are suitable either for generic processing (e.g. filtering, enhancing, detection of low-level elements) that can be applied to most images regardless of content [12]. Other algorithms and specialized operations are designed for a specific purpose [5], [13]. Since our test bench will use static agents, it is important to determine which algorithms will be implemented before the deployment of the agents.

In order to run the agents in test mode we consider that a simple algorithm that can be executed in time proportional to the image size (i.e. using CPU cycles proportional to the number of pixels) is sufficient. Instead of embedding the data and algorithm on the reactive agent, data will be passed from the broker to the agent and received back – this will enable to run monitoring tasks with varying image sizes to allow testing the resources with different work loads, which may be particularly useful to stress-test network loads when transferring data between the broker and resources.

Good candidates for the test mode tests are algorithms that iterate several times over the pixels of the image, e.g. edge detection algorithms, convolution algorithms, etc. These algorithms run time does not depend on the image contents or random initial conditions, differently from iterative algorithms such as region growing and clustering.

Image processing operators to be executed in the run mode will be highly dependent on the distributed data processing task at hand. When designing the broker and agents to deal

with image processing algorithms, one must consider the nature of the algorithm and how much of the image does it need to access to execute its task.

Image processing operations can be divided into three categories with regards to the extent of the image required for the processing:

- **Point or Pixel-based operations** are operations that can be applied independently to each pixel on the image. In other words, in order to perform an operation in a pixel there is no need for information on other pixels on the image. Some examples of this operation are color transformations, contrast and brightness transformations, pixel-based classification.
- **Area or Region-based operations** are operations that requires information on a pixel and on a surrounding region. Some examples of these operations are filters that consider the pixels' neighborhood and region growing-based image segmentation techniques, that join neighbor pixels based on their similarity in order to identify homogeneous regions. Operations such as region growing can have geometric limits applied so they won't require the whole image for completion: this approach is used for distributed image processing, but often require another step where the different image regions are joined together considering the segments generated by the previous step.
- **Geometric-based and Statistics operations** are operations that need information on all other pixels on the image in order to transform the image or to calculate some information based on its pixels. Some examples of geometric operations are image warping, image anamorphosis and geometric corrections in remotely sensed images. Some examples of statistics operations are extrema identification and statistical image segmentation (e.g. by clustering).

This classification is not complete, and its categories are not exactly mutually exclusive. For example. statistics operations can be performed by partitioning the image in blocks, calculating the statistics for each block and post-processing the results so the final values will be related to the whole image pixels. Similarly, some geometric-based operations may be reduced to area-based operations through some (e.g. flipping images vertically or horizontally may be done using repeated operations using two disjoint areas) but we will consider those as being geometric operations for generalization sake.

Usually an image processing task requires the processing of several images (i.e. files) that can often be considered independent of each other; therefore can be processed in parallel. Under these conditions, we can safely assume that the image processing operations developed and deployed will require the whole image for processing.

III. STATUS, FUTURE WORK AND CHALLENGES

This paper presents the architecture and general concepts of an ongoing project involving researchers from the Kyushu Sangyo University in Japan and the Brazilian National Institute for Space Research in Brazil.

Some of the concepts for the test bench were already developed in other works [5], and some are under development. The motivation for the test bench is the description of similar approaches to evaluate cloud computing comparing public and private clouds and the operating system for deployment [14], evaluation of different aspects such as cost, complexity of migration and support [2] and some general approaches to testing and evaluating the performance of clouds [15] that did not fit our needs.

Some of the challenges and open issues on this research are:

- There may be a large difference between the complexity of the tasks which are run in test mode and run mode, moreover if we consider image processing tasks that need to work over large images (therefore being impractical for running in the test mode). Run mode tasks will be related to real image processing applications, so complex algorithms, that run in non-linear time may be used. Depending on the algorithm used, a smaller version of it that maintains its non-linear time may be developed to be executed in test mode.
- Deployment of the test bench in a real hybrid cloud scenario may be complex: in order to test with only private/community clouds we could use both institutes' clouds, but tests with commercially available public clouds will require the acquisition of resources in those public clouds, moreover if we want to test different offerings of public clouds.

ACKNOWLEDGMENT

Rafael Santos was supported by the INPE's PCI/CNPq grant number 551006/2011-0. Bernady O. Apduhan was supported in part by the Japan Society for the Promotion of Science Grants-in-Aid for Scientific Research, 24500100.

REFERENCES

- [1] P. Mell and T. Grance, "The NIST definition of cloud computing (draft)," *NIST special publication*, vol. 800, p. 145, 2011.
- [2] D. Högberg, "An applied evaluation and assessment of cloud computing platforms," Ph.D. dissertation, Umeå University, 2012.
- [3] H. Aihara, C. A. Prieto, D. An, S. F. Anderson, É. Aubourg, E. Balbinot, T. C. Beers, A. A. Berlind, S. J. Bickerton, D. Bizyaev *et al.*, "The eighth data release of the Sloan Digital Sky survey: first data from SDSS-III," *The Astrophysical Journal Supplement Series*, vol. 193, no. 2, p. 29, 2011.

- [4] A. R. Thakar, "The Sloan Digital Sky Survey: Drinking from the fire hose," *Computing in Science & Engineering*, vol. 10, no. 1, pp. 9–12, 2008.
- [5] F. La Barbera, R. De Carvalho, J. Kohl-Moreira, R. Gal, M. Soares-Santos, M. Capaccioli, R. Santos, and N. Sant'Anna, "2dphot: A multi-purpose environment for the two-dimensional analysis of wide-field images," *Publications of the Astronomical Society of the Pacific*, vol. 120, no. 868, pp. 681–702, 2008.
- [6] F. Becker-Guedes, E. de Paula, L. de Rezende, S. Stephany, I. Kantor, M. Muella, P. Siqueira, K. Correa, A. Dutra, C. Guedes *et al.*, "Monitoring, mapping and prediction of ionospheric scintillation over the Brazilian equatorial and low latitude regions," in *38th COSPAR Scientific Assembly*, vol. 38, 2010, p. 4237.
- [7] R. B. Andrade, L. Nunes, E. de Moraes Barbosa, N. L. Vijaykumar, and R. Coelho dos Santos, "A web service-based framework for temporal/spatial environmental data access," in *Computational Science and Its Applications (ICCSA), 2012 12th International Conference on*. IEEE, 2012, pp. 7–13.
- [8] A. Thakar and A. Szalay, "Migrating a (large) science database to the cloud," in *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*. ACM, 2010, pp. 430–434.
- [9] S. Green, L. Hurst, B. Nangle, and P. Cunningham, "Software agents: A review," 1997.
- [10] M. Kemps-Snijders, M. Brouwer, J. P. Kunst, and T. Visser, "Dynamic web service deployment in a cloud environment," in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, N. C. C. Chair, K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, J. Odijk, and S. Piperidis, Eds. Istanbul, Turkey: European Language Resources Association (ELRA), may 2012.
- [11] R. C. Gonzalez and E. Richard, "Woods, digital image processing," ed: *Prentice Hall Press, ISBN 0-201-18075-8*, 2002.
- [12] R. Santos, "Java Image Processing Cookbook," *Online: <http://www.lac.inpe.br/JIPCookbook>*, 2010.
- [13] R. H. Lupton, Z. Ivezić, J. Gunn, J. Knapp, M. Strauss, and N. Yasuda, "The sdss imaging pipelines," in *Proc. of SPIE Vol.*, vol. 4836, 2002, p. 351.
- [14] C. Pelletingas, "Performance evaluation of virtualization with cloud computing," Ph.D. dissertation, Edinburgh Napier University, 2010.
- [15] A. Antoniou, "Performance evaluation of cloud infrastructure using complex workloads," *Comput. Syst.*, vol. 27, pp. 1011–1026, 2011.