Spare CPU Cycles as a Service – A New Cloud Computing Paradigm

Kireet Kokala, David Andersen October 30, 2009
Advanced Operating Systems, Dr. Yanqing Zhang, Georgia State University

ABSTRACT

We propose the merging of two technologies: the Folding@Home (FAH) distributed application and cloud computing. By relocating the utilization of spare CPU cycles from disparate and often unreliable distributed volunteer nodes, which now perform the protein folding calculations, to the large, secure, and dedicated server farms of large cloud computing vendors, we can demonstrate concrete increases in efficiency and throughput. As our novel proposal would still leverage only the currently unused spare computing power available from the many computers in the cloud, the costs for this model would still be incredibly reasonable and fit into the cloud's existing Software as a Service (SaS) business model.

1. INTRODUCTION

With the increasing demand for reusable and easily accessible data in domains like health-care, military fields, the commercial industry, and scientific research, the need for cost-efficient means of computation is at an all time high. Many scientific problems are computationally intensive but parallel in nature, allowing the necessary computations to be partitioned into many fully independent processes. One such problem includes protein folding.

In this paper we will discuss the merits and shortcomings of the current methods, and then introduce our novel solution to obtaining the computational resources needed to effectively tackle scientific problems such as protein folding [1].

1.1 BACKGROUND

At present, there are two sources of computing power available to protein researchers. The traditional approach is to directly purchase the hardware and software needed to meet the computing requirements of the problem. The advantage of this approach is the control over scheduling, security, and resource management. Cost, both in the initial purchase of infrastructure and ongoing maintenance, is of course, the primary downside to this approach. Current research into improving this model is the migration of the processing from dedicated infrastructure to the "cloud" [2]. Section 1.2.1 discusses the cloud computing paradigm.

A second approach to acquiring processing power is to enlist the public. Volunteer computing projects such as Folding@Home have been set up by researchers to distribute the computations among thousands of individuals who have agreed to donate the spare CPU cycles of their personal computer to the cause. In section 1.2.2 the current research into volunteer computing applications with regards to protein folding are discussed.

We introduce our novel solution and look to the future in Sections 1.3 and 1.4 respectively.

1.1.1 CURRENT RESEARCH – CLOUD COMPUTING

Cloud computing has garnered a great deal of attention in recent years. The attraction of this computing resource model for many technology departments, large and small, is threefold. First, by embracing the cloud as their information technology backbone, an IT department can eliminate the large initial capital expenditures required to set up a computing infrastructure and reduce the significant overhead costs of the department by offloading the software upgrades and hardware maintenance to a third party. Secondly, and perhaps more importantly, the department can optimize its budget by eliminating the need to overbuy technology to allow for future needs—the cloud provides instant scalability with no up-front costs. And finally, when the potential for the additional security and monitoring provided by the dedicated staff of the cloud vendor are added to the equation, the advantages of the cloud model are quite compelling. Simply put, the cloud offers the potential for not only a large capacity of spare computing power, but importantly a reliable source.

Current research in cloud computing includes enabling existing grid infrastructures to support the cloud model. In [8], a cloud application is proposed which would be made available by means of a business Service Oriented Architecture layered over a gLite Grid infrastructure. The proposed application aims to offer guaranteed business services by dealing with the unpredictability of service demand by allocating and deallocating resources according to current need. However, in spite of the increased reliability and cost savings offered by the cloud through this and other research, merely moving the processing platform from purchased to rented infrastructure doesn't fully leverage the fact that timeliness is secondary priority of protein folding problem.

Volunteer computing, by comparison, offers a low cost alternative to purchased or rented infrastructure, by sacrificing guarantees of timeliness.

1.1.2 CURRENT RESEARCH – VOLUNTEER COMPUTING

Amongst several approaches found in the ongoing research phase of the project, there has been a trend to establish a stable foundation to build distributed applications upon. Once such ambitious project, which has also been inspired in part by FAH and the SETI (Search for Extraterrestrial Intelligence) projects, has been presented by C. Pheatt of Empora State University. An easy to set-up and configurable distributed system suite for fairly large computational problems has been developed [6]. The paper introduces a client-server suite that focus on solving problems in a hybrid, peer-to-peer grid. Several considerations have been taken into account to make the suite easy for undergraduate students to use: namely, setup, ease of use, and documentation. The main tenets of their distributed system architecture address: a) the distributed paradigm, b) a main single distributed algorithm, and c) strong design and simulation/visualization implementation of said system [6]. While the approach extends well to unused and old Linux systems in order to build a computing cluster, a windows platform was targeted for the application. The client and server were developed in Visual C++ 2003 with modest server/client sizes of .25MB and .13MB.

Simplicity and robustness of the application was key during development. Standard TCP implementation was used wherein the protocol has been designed to minimize communication overhead using an MD5 message digest algorithm. Like FAH and SETI, the client program is a screensaver that displays a bitmap rendering of an image to showcase evolving computations via graphical transformations. Threads of execution are also used throughout the application. For example, one thread manages the OpenGL screensaver while another maintains a heartbeat message to the server [6]. Upon configuration, an administrator can access information about the cluster and view it's total status on completing a job, and also optionally launch the batch job scheduler—a vital necessity in distributed system computing. Sample and built-in applications are included within the suite to showcase performance: an obligatory hello world program, a FireStarter program, and an MD5 brute force attacker are some examples. Analysis for the program can also be performed by a built in results viewer.

Given the modesty of the application suite and relatively easily obtainable hardware, the results on 25 3.2GHz Windows XP computers and 20 200MHz to 1GHz mixed machines were benchmarked using LinPack [6]. The highest recorded performance measure was R = 17.8 Giga Flops, which pinned the Top 50 supercomputing system listings in 1995 by using spare CPU cycles. Since the research is readily adaptable to the clients application, it bears a very strong affinity our research proposal.

A similar effort has been put forward by Hiroshi Takemiya et. al in their paper on sustainable super-grid computing [5].

1.2 ANALYSIS

An analysis of these two approaches leads to conclusion that the reliability and dynamic resource allocation offered by the cloud is highly desirable for a computationally intensive problem.

Contrast this with the current volunteer computing distributed applications, which not only suffer from a continually varying processing capacity, but loses efficiency because several calculations must be restarted or verified and the application must recover from errors because of the inherent unreliability of data being returned by the volunteer nodes [6, 2].

Yet, in current cloud models, the price paid for a processing service is still at a premium as resources are specifically allocated for a given task. The cost model of volunteer computing is much preferable to the cloud.

1.3 NOVEL IDEA

Rather than specifically allocating CPU cycles to our protein folding problem, our proposal would leverage only the currently unused spare computing power available from the many computers in the cloud. By accepting the lowest priority standing the job queues, CPU cycles that would normally go unused could be purchased at a discount.

A further advantage of distributing our protein folding application in the cloud versus across volunteer computing infrastructure is that if throughput for a specific calculation does become an issue for a given problem (perhaps to meet a submission deadline), the cloud offers the inherent processing scalability necessary should an increase in priority for a given calculation be needed (and purchased) by the researcher.

1.4 FUTURE PLAN

Our proposal is a specific implementation or in a strict sense, a hybrid porting of functionality from a closed system to an open one. Full implementation of such an ambitious project is beyond the scope of this paper. However, in progressing towards an actual, future research on could be to learn from the architecture and implementation complexity of the system, and to establish an open framework that can be built cross-platforms, which can have several user interface, back-end functionality, and even decentralized management, which can in turn enable control and repository maintenance of FAH-type applications on cloud-distributed hybrid systems [6,7].

2. RESEARCH PLAN

As a first step towards proving the viability of our novel idea, the research problem to be addressed in this paper is the analysis of a theoretical implementation of a hybrid distributed-cloud. Specifically, our focus is to analyze the search complexity of a cloud-based computing method vs. a distributed computing method and directly compare and contrast it with that of the hybrid simulation.

An L-Kynureninase enzyme mutation conducted during 2003-2005 in the University of Georgia biochemistry department will be used as the baseline for a real-time computation problem featuring multiple mutation sequences with appropriate depth searches, best, and worst case scenarios [3-4].

2.1 NEW RESEARCH IDEAS

To accomplish such a large-scale task, access to a ready network of computers or cluster computing resources is would be desirable. However, in lieu of such resources, we propose the following step-wise MATLAB simulation:

(i) Simulate a weighted and directed graph G defined by G(V,E) to represent an arbitrary distributed system Q in a cloud.
(ii) Assume Q_1...Q_2..Q_n are part of the cloud with similar computational capacities.
(iii) Add new vertices to G with weights being on a log scale where primes represent freeCPU cycles.
(iv) Compute the times for the following activities:
  (a) Minimizing the shortest path between newly joined vertex Z and V.
  (b) Planarity
  (c) The minimization also releases appropriate primes to be reused.

We will attempt to prove that an increase in vertices and minimization increases with time, different graphs joining the cloud, and no data losses. Though it is predicted that the complexity of the algorithm will be $W(n) = O(n^2)$, it is equally like that $O(\log n)$ might be the result, which is currently pending actual implementation and test results.

2.2 THEORETICAL WORKS

Some theoretical works have been analyzed during the inception and planning phases of the project. For example, frameworks [6,7] were studied to understand the architecture of large-scale distributed systems. However, since the project is largely homogeneous (i.e. the simulation runs on one machine) and because it does not leverage the actual computing capacity of several machines, the peer-to-peer and adaptive-grid algorithms have not been applied [6,7]. It is duly noted that such literature is of prime importance and bears much weight in Section 1.4 of this paper.

2.3 ANALYSIS

While many technologies, solutions, and, implementations are available for distributed system computing tasks, this paper explores a new cloud-based distributed computing paradigm, spare CPU cycles as a service. It is our belief that analysis of our simulation will show that an efficient implementation of our idea is indeed possible, and lay the groundwork for further research into its full implementation.

3. REFERENCES

[1] E. Stoneman, "Cloud-Distributed Computing," http://geekswithblogs.net /EltonStoneman/archive/2008/12/02/cloud-distributed-computing.aspx, 12-2-2008.

[2] http://en.wikipedia.org/wiki/Cloud_ computing, "Cloud Computing".

[3] Sriganesh V. Koushik et Robert S. Phillips, "Cloning, Sequence, and Expression of Kynureninase from Psuedomonas fluorescens," *Archives Biochemistry and Biophysics*, 1997, Vol. 344, pages 301-308.

[4] Gawandi, V., Liskey, D., Lima, S., and S. Phillips, "Reaction of beta-Benzoyl-L-alanine with Kynureninase from Pseudomonas fluorescens: Detection of a New Reaction Intermediate and a Change in Rate-determining Step," *Biochemistry*, 2004, Vol. 43, pages 3230-3237.

[5] Hiroshi Takemiya, Yoshio Tanaka, Satoshi Sekiguchi, Shuji Ogata, Rajiv K. Kalia, Aiichiro Nakano, Priya Vashishta, "Sustainable adaptive grid supercomputing: multiscale simulation of semiconductor processing across the pacific," *Conference on High Performance Networking and Computing Archive, Proceedings of the ACM*, No. 106, 2006.

[6] Chuck Pheatt, "An easy to use distributed computing framework," *Technical Symposium on Computer Science Education Archive, Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education*, 2007, pages 571-575.

[7] Jerome Verbeke, Neelakanth Nadgir, Greg Ruetsch, and Ilya Sharapov, "Framework for Peer-to-Peer Distributed Computing in a Heterogeneous, Decentralized Environment," Lecture Notes in Computer Science, *Proceedings of the Third International Workshop on Grid Computing*, 2002, Vol. 2536, pages 1-12.

[8] Carmelo Ragusa, Francesco Longo, and Antonio Puliafito, "Experiencing with the Cloud over gLite", *Proceedings of the 2009 International Conference on Software Engineering Workshop on Software Engineering Challenges of Cloud Computing* , 2009, pages 53-60.