Towards a Quality of Service Aware Public Computing Utility

Muthucumaru Maheswaran, Balasubramaneyam Maniymaran, Shah Asaduzzaman, and Arindam Mitra
McGill University
Montreal, QC H3A 2A7
Canada

Abstract

This paper describes a design for a quality of service aware public computing utility (PCU). The goal of the PCU is to utilize the idle capacity of the shared public resources and augment the capacity with dedicated resources as necessary, to provide high quality of service to the clients at the least cost. Our PCU design combines peer-to-peer (P2P) and Grid computing ideas in a novel manner to construct a utility-based computing environment. In this paper, we present the overall architecture and describe two major components: a P2P overlay substrate for connecting the resources in a global network and a community-based decentralized resource management system.

1 Introduction

Computing utilities (CUs) much like Grid computing [7] are based on the idea of constructing very large virtual systems by pooling resources from a variety of sources. However, unlike Grid computing, utility computing focuses on providing a *utility* like interface to the virtual pool similar to that provided by common public utilities such as electricity or water. One of the defining feature of a utility is the *commoditization* of the resources that makes them provider-neutral and simplifies activities such as metering and billing. In a distributed computing system, commoditization means we categorize the computing resources into virtual resources that provide predefined sets of services. The benefits and challenges of a utility computing lie on efficiently realizing the commoditization process in distributed computing systems.

Typically, computing utilities are built using resources that are supplied by a single or few providers. These resources are installed for the exclusive use of the computing utility. This approach naturally limits the scalability and geographical scope of the computing utility. However, one of the advantages of this approach is that resource behavior is well managed resulting in predictable *quality of service* (QoS). Here, we consider an extended notion of computing utility called *public computing utility* (PCU) that opens up the membership to *public resources* much like the *peer-to-peer* (P2P) file sharing systems. This enables the PCU to leverage vast amounts of idle resources spread throughout the Internet. In addition to lowering the cost of participation, the PCU prevents provider monopoly and creates a geographically distributed resource base that is capable of satisfying location specific resource requirements.

This paper introduces a QoS aware PCU design called *Galaxy*. Delivering QoS while solely relying on public resources is hard to accomplish for the PCU because a public resource working for a client can defect from the PCU at any time. One way to compensate for this uncertainty is to employ redundant public resources. Another approach is to switch the client to dedicated resources once the public resources are determined to perform below the expected performance level. Our PCU design supplements the capacities harnessed from public resources with dedicated resources to meet the performance expectations of the clients.

Section 2 introduces the overall architecture of Galaxy. The routing substrate of Galaxy called the *resource addressable network* (RAN) is discussed in Section 3. The *Galaxy resource management system* (GRMS) is described in Section 4. Section 5 briefly examines the services and applications that can be supported by the Galaxy architecture, respectively. Other research works related to Galaxy are presented in Section 6.

2 A Public Computing Utility Architecture

The proposed architecture for the Galaxy PCU is shown in Figure 1. The lowest layer of the architecture is the P2P overlay network called the *resource addressable network* (RAN). All the resources that participate in the PCU plug into the RAN. The RAN provides the resource naming, discovery, and access services to the PCU. The next upper layer is the *Galaxy resource management system* (GRMS). Similar to the RAN, the GRMS is also organized as a P2P overlay network of managerial entities called *Resource Brokers* (RBs). In the RAN, the peers are virtualized resources whereas in the GRMS the peers are RBs. The trust/incentive management is a collaborating module to the GRMS. It controls the behavior of resources, especially the public resources, in the system. The next upper layer is the Galaxy services. Although the architecture does not impose any restriction on the organization of this layer, it could be orga-

nized as a P2P network. Example Galaxy services include application level QoS managers, shell interfaces, and network file systems. Security is a layer in this Galaxy middleware that spans all the other layers in parallel to provide the system from malicious activities (external and internal to the system).

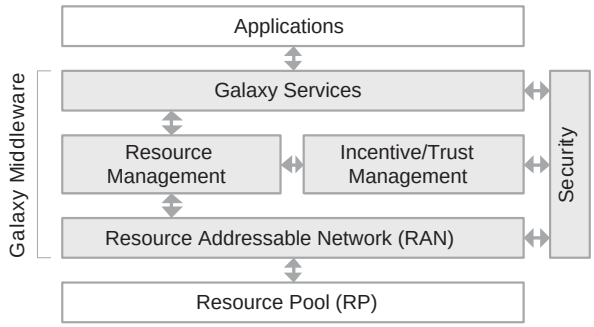

Figure 1. The Galaxy Architecture.

3 Resource Addressable Network

The RAN is a decentralized, self-organizing overlay that interconnects all resources in Galaxy. When a resource joins the RAN, it is analyzed to obtain a description of the resource characteristics as a set of attribute-value tuples. This set is then profiled into a resource *type* which is used as its name. This *profile-based* naming differs from the generally used *description-based* naming [10] where the attribute-values are used uncompressed to "name" resources. Even though the description-based naming provides flexibility in querying resources, it suffers from the overheads in managing databases of resource descriptions. On the other hand, profile-based naming labels the resource descriptions into unique names to avoid these overheads. However, labeling all the possible description can lead to an unmanageable name space. Therefore only the popular resource descriptions are considered for labeling. The naming is such that the un-labeled descriptions are mapped onto closest matching labeled profiles. The concept of profile-based naming is supported by the argument that, in practical conditions, popularity of the resources are largely skewed towards a small number of resource descriptions. Profiling only these descriptions can satisfy a large portion of user queries while keeping the system overhead low. Thus, the profile-base naming trades-off the performance for reduced overhead.

Once profiled, resources of the same types are collected into *type rings*(Figure 2). Rings are overlay structures created by at least two routing pointers in each node to its left and right neighbors. Type rings are connected by *neighborhood rings*. These rings are created as *space-filling curves* [11] that support a decentralized, scalable discovery mechanism that provides $\log n$ hop-complexity. The RAN routing tables are incrementally built and managed by each resource in the RAN. They are two layered, one for routing along the profile space to reach the required type ring and the other for routing along the type ring to reach the desired location. This *location-based* routing within the type rings provides a QoS-aware discovery substrate for the Galaxy.

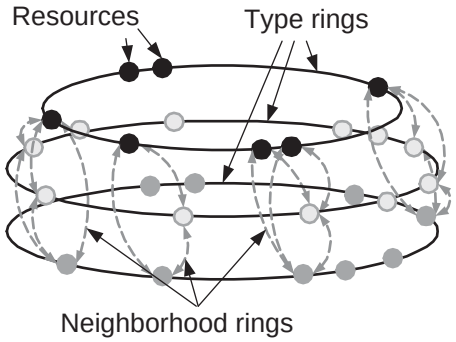

Figure 2. Ring arrangement of the RAN.

4 Galaxy Resource Management System

4.1 Overview

The GRMS layer is composed of two sets of entities: (a) *resource peers* (RPs) and (b) *resource brokers* (RBs). RPs are GRMS layer representatives of the resources present in the Galaxy. The resources RPs represent can be individual resources, cluster of resources, virtual resources, or software entities. RPs are the mediators in the GRMS resource allocation: they launch resource requests on behalf of the entities they represent and regulate the resource acquisition requests received for the resources under their control. Aggregating multiple resources under an RP has number of architectural benefits: it matches the administrative domains present in real world and provides an easy way of handling trust levels and incentive shares (Section 4.2).

The *Resource Brokers* (RBs) are the entities that coordinate most of the activities within the GRMS layer. Any RP can act also as an RB as long as it has sufficient reputation such that existing RBs accept the new one. RBs have their virtual representations in the RAN level, forming a separate RAN type ring of "RB type". This enables the RBs to use the RAN's scalable resource discovery mechanism to discover resources and other RBs. Two major functionalities of the RBs are to search and allocate resources as requests emerge from RPs and to assign and revalidate trust levels and incentive shares of the RPs. Each RP chooses an RB (probably the closest) to send the resource requests. The RB uses the RAN discovery mechanism to discover the appropriate resources. The discovered resources tell the RB

who their RP(s) are. The RB then mediates with the destination RPs to acquire the resources for the requesting RP. At the end of the resource utilization, the donor and client RPs report the performance during the utilization back to the RB and based on which the RB readjusts the incentive shares owned by the donor RPs.

4.2 Incentive Management

The core idea of the incentive management system is to make each client a *share holder* of the global utility by issuing shares of well defined values. The number of shares held by each resource denotes a client's eligibility to access the PCU's services and is directly dependent on two parameters: (i) capacity donated by the resource to the global utility and (ii) consistency of commitment of the resource towards the utility. The incentive is to perk the clients to consistently commit their capacities to the PCU.

We use the notion of *Ca*pacity *Sh*ares (CASH) to measure a client's share amount. When a client joins the PCU system, it is profiled by the PCU and the maximum number of CASHs (*maxCASH*) the client is entitled to hold is determined. Thereafter, a client's contributions are marked by assigning it a fraction of the *maxCASH* over some time periods (called epochs). The resource should work for the PCU for certain number of epochs to accumulate its full quota of CASH. Once a client earns it full quota of CASH, it gets no more shares. But to ensure the consistency of commitment, the client is expected to remain connected with the PCU in a continuous manner. If the resource departs the PCU it loses half of the shares held by it. The RBs award the clients CASH using time-limited certificates and it is the responsibility of the clients to renew their CASH certificates. At time of the renewal, a client client can present *participation certificates* or signed *work orders* to establish to the RB that it has contributed capacity to the PCU since the time the CASH certificate was issued. The work orders are issued by the RBs when the client's resource is allocated to another client in response to a resource request. More detailed descriptions for the CASH incentive mechanism is given in [2].

4.3 Quality of Service Management

Even though other layers are QoS aware, the GRMS layer implements the "core" set functionalities to manage the QoS delivered by Galaxy. The big part of the Galaxy resource pool made up of publicly owned resources that are guided by incentives and reputation. But, it has been previously shown that it is hard to guarantee QoS with uncontrolled public resources alone [8]. Galaxy significantly improves the QoS guarantees by augmenting the public resources with a fully controlled and reliable pool of dedicated resources.

We implement this idea in conjunction with the QoS guaranteed resource scheduling service provided by the RBs. An RB pre-allocates a pool of highly trustworthy resources and uses capacities from this pool as required to guarantee QoS for requests from client RPs. When an allocation is made by the RB to a client RP, an implicit *service level agreement* (SLA) is drawn between the client RP and the RB. The RB will reallocate the request from the client RP if it detects the SLA can be violated due to unreliable behavior from the provider RP that is engaged in the allocation. After the reallocation, the client RP might be served by resources from the pre-allocated resource pool associated with the RB. The RB wants to maximize the fulfillment of the SLAs of client RPs while maximizing the utilization of the pre-allocated resource pool. The prime objective of the RB is to utilize both the infinite public pool and the finite sized dedicated pool to deliver the best services to the client RPs. To do this, online scheduling algorithms are used to route and re-route the resource requests onto the proper resources.

5 Galaxy Services and Applications

Galaxy service is the highest layer of the Galaxy middleware. This provides generic capabilities for resources to perform activities such as launching resource acquisition commands, managing and monitoring acquired resources, and releasing resources. Some example of the Galaxy services are (a) a Unix shell like command line interface called *Galaxy shell* (GShell); (b) The *Galaxy network file system* (GNFS); and (c) *application-level QoS management*.

Galaxy is the suitable place for hosting many different applications: the Internet scale distribution of resources makes Galaxy suitable for web content distribution. Further, the public resources of Galaxy being always at the edge of the Internet supports an efficient edge delivery of the content. This property in addition to the immense storage and processing capacity of Galaxy makes it suitable also for streaming content distribution. Also the high-throughput computing applications can enjoy the virtually unlimited computing capacity of the Galaxy.

6 Related Work

Grid [7] is a distributed enterprise solution where different institutions pool their resources together to build a high-performance computing platform. Even though Grid concept originated as a high-performance computing platform, it has now evolved into a service-oriented *Open Grid Services Architecture* (OGSA) [4]. One of the important drawbacks of the Grid approach is its limited scalability due to the requirement for high trust among the participating institutions, which restricts the membership to a few institutions. In contrast to this "close" communities Grid presents,

the PCU is an *open* architecture that solicits wide participation.

Utility computing is based on the notion of "commoditizing" computer resources. It can simplify resource management and increase resource usage by introducing a *unified* interface to heterogeneous resources. HP Lab's *Utility Data Center* (UDC) [6] is one project that implements utility computing. It pools together the resources of an institution and provides mechanisms to create service-specific resource *farms* according to user specifications. Recent focus of UDC projects is to combine the UDC technology with Grid architectures to provide inter-enterprise resource sharing [5]. Besides UDC, there are other projects such as *Cluster on demand* (CoD) [3] and *Virtual appliances and Collective* [12] with similar goals. While Galaxy shares several key ideas with the utility computing systems, it differs from these projects because it is designed to implement (i) commoditization at the core and utilizes this notion to efficiently implement resource naming and discovery, (ii) relaxed participation models to induct public resources into the system, (iii) geographically scalable resource management architectures.

There is an emerging class of projects that use public resources in an "online" manner. These projects attempt to extend P2P system to generalized computing platforms. With a P2P architecture base, the resource management becomes complex as incentives and trust are introduced into the system. *Cluster computing on the Fly* (CCoF) [9] is an example P2P-based generalized computing system. Even though CCoF is very similar to the PCU, one of the fundamental differences is that PCU is focused on delivering QoS to its clients requiring it to have a stronger resource management system. As part of Galaxy, we present a community-oriented architecture for the RMS. *Xenoservers* [1] shares the same view of a PCU as our Galaxy. It addresses the issue of incorporating public resources into the system by designing incentive and trust mechanisms. However it can be seen that Xenoservers lacks a highly scalable discovery substrate like RAN. The RBs in Xenoservers (called *Xenocorps*) are overloaded with naming and discovery services. With unstructured peer arrangement among Xenocorps and the usage of description based naming, it is hard to expect high scalability.

7 Conclusion

We presented the design of Galaxy a QoS-aware PCU. We presented the principles that govern the overall design and a detailed description of the Galaxy architecture. Two major components of the overall architecture were examined in more detail. One was the P2P overlay substrate called the RAN. It provides the connectivity among a variety of Galaxy elements that have a requirement to name and discover each other in a location sensitive manner. Another was the GRMS that provides various services including resource allocation, incentive management, and trust management. One of the unique aspects of GRMS was its community-oriented architecture. In this architecture, the resources are associated with Galaxy management elements in an on-demand basis. This paradigm is well suited for PCU where the resources including the ones running the Galaxy management elements can leave or join the Galaxy at any time.

Currently, the Galaxy system is under development. We are focusing on the development of various Galaxy components such as the RAN, GRMS, and GShell. We intend to deploy an initial version of Galaxy on the Planet-Lab for testing and benchmarking purposes.

References

[1] Xenoservers. http://www.cl.cam.ac.uk/Research/SRG/netos/xeno/.

[2] A. Mitra and M. Maheswaran. Resource Pariticipation Models for Public-Resources in Computing Utilities. Technical report, School of Computer Science, McGill University, May 2004.

[3] J. Chase, L. Grit, D. Irwin, J. Moore, and S. Sprenkle. Dynamic virtual clusters in a grid site manager. In *The 12th International Symposium on High Performance Distributed Computing (HPDC-12)*, June 2003.

[4] I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The physiology of the grid: An open grid services architecture for distributed systems integration. In *Open Grid Service Infrastructure WG, Global Grid Forum*, June 2002.

[5] S. Graupner, J. Pruyne, and S. Singhal. Making the utility data center a power station for the enterprise grid. Technical Report HPL-2003-53, HP Labs, Mar. 2003. Tech Report.

[6] HP Labs. HP utiltiy data center: Transforming data center economics. http://www.hp.com/go/udc, 2004. White Paper.

[7] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the Grid: Enabling scalable virtual organizations. *International Journal on Supercomputer Applications*, June 2001.

[8] C. Kenyon and G. Cheliotis. Creating services with hard guarantees from cycle harvesting resources. In *Proceeings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'03)*, 2003.

[9] V. Lo, D. Zappala, D. Zhou, Y. Liu, and S. Zhao. Cluster computing on the fly: P2p scheduling of idle cycles in the internet. In *The 3rd International Workshop on Peer-to-Peer Systmes (IPTPS 2004)*, Feb. 2004.

[10] R. Raman, M. Livny, and M. H. Solomon. Matchmaking: Distributed resource management for high throughput computing. In *The 7th IEEE International Symposium on High Performance Distributed Computing*, Chicago, IL, July 1998.

[11] H. Sagan. *Space-filling curves*. Springer-Verlag Telos, New York, Aug. 1994.

[12] C. Sapuntzakis and M. S. Lam. Virtual appliances in the collective: A road to hassle-free computing. In *The 9th Workshop on Hot Topics in Operating Systems (HOTOS IX)*, pages 55–60, May 2003.