International Parallel and Distributed Processing Symposium, April 15-19, 2002.

Architecture of the Entropia Distributed Computing System

Andrew A. Chien*
Entropia, Inc, 10145 Pacific Heights, Suite 800, San Diego, CA 92121
and
University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093

Abstract

Distributed Computing, the exploitation of idle cycles on pervasive desktop PC systems offers the opportunity to increase the available computing power by orders of magnitude (10x - 1000x). However, for desktop PC distributed computing to be widely accepted within the enterprise, the systems much achieve high levels of robustness, security, scalability, unobtrusiveness, and manageability.

We describe the Entropia Distributed Computing System, as a case study, detailing its internal architecture and philosophy in attacking these key problems. In particular, key aspects of the Entropia system include the use of: 1) scalable web/database technology for system management, 2) network tunneling and application namespaces for logical connectivity, 3) binary sandboxing technology for security and unobtrusiveness, and 4) open integration model to allow applications from many sources to be incorporated.

These technologies are combined to produce a robust, flexible, high performance system which is in use in numerous enterprises supporting a wide range of applications.

Introduction

For nearly four years, the largest computing systems in the world have been based on "distributed computing" -- the assembly of large numbers of PC's over the Internet. These "grid" systems sustain multiple teraflops continuously by aggregating hundreds of thousands to millions of machines, and demonstrate the utility of such resources for solving a surprisingly wide range of large-scale computational problems in data mining, molecular interaction, financial modeling, etc. These systems have come to be called "distributed computing" systems and leverage the unused capacity of high performance of desktop PC's [1,2] (one to 2.2 Gigahertz machines with multi-gigaop capabilities), high speed local-area networks (100 Mbps to 1Gbps switched), large main memories (256MB – 1GB configurations), and large disks (60 to 100 GB disks). Such "distributed computing" systems leverage the installed hardware capability, and thus can achieve a cost per unit computing (or Return-On-Investment) superior to the cheapest hardware alternatives by a factor of five or as much as ten. As a result, distributed computing systems are now gaining increased attention and adoption within the enterprises to solve their largest computing problems and attack new problems of unprecedented scale. For the remainder of the paper, we focus on enterprise distributed computing.

While the tremendous computing resources available thru distributed computing present new opportunities, harnessing them in the enterprise is quite challenging. Because distributed computing exploits existing resources, to acquire the most resources, capable systems must thrive in environments of extreme heterogeneity in machine hardware and software configuration, network structure, and individual/network management practice. The existing resources have naturally been installed and designed for purposes other than distributed computing, (e.g. desktop word processing, web information access, spreadsheets, etc.), the resources must be exploited without disturbing their primary use. And to achieve a high degree of utility, distributed computing must capture a large number of valuable applications – it must be easy to put application on the platform – and secure the application and its data as it executes on the network. And of course, the systems must support large numbers of resources, thousands to millions of computers, to achieve their promise of tremendous power, and do so without requiring armies of IT administrators. In the remainder of this paper we describe:

- the long history of distributed computing activity which has led us to the large-scale distributed computing systems which are the focus of the Entropia system,
- the key technical requirements for distributed computing technology: robustness, security, scalability, manageability, unobtrusiveness, and openness/ease of application integration, and
- the Entropia system architecture, including its key elements and how it addresses the key technical requirements.

*Corresponding author: achien@entropia.com

In addition, to describing how distributed computing systems work, we will also expose how the challenges and opportunities for this technology differ from traditional parallel computing systems and desktop systems. Beyond these issues, we briefly present a perspective on the future potential of distributed computing.

Background

The idea of distributed computing has described and pursued as long as there has been computers connected by networks. Early justifications of the ARPANET [3], described the sharing of computational resources over the national network as a motivation for building the system. In the mid 1970's, the Ethernet was invented at Xerox PARC, providing high bandwidth, local-area networking. This invention, combined with the Alto Workstation, presented another opportunity for distributed computing, and the PARC Worm [4] was the result. In the 1980's and early 1990's the Condor Project [5,6] developed a distributed computing system that supported many Unix systems, and became popular amongst a number of academic institutions.

The growth of the Worldwide Web (WWW) [7] and exploding popularity of the Internet created a new, much larger scale opportunity for distributed computing. For the first time, millions of desktop PC's were connected to wide-area networks both in the enterprise and in the home. The number of machines potentially accessible to an Internet-based distributed computing system grew into the tens of millions of systems for the first time. The scale of the resources (millions), the types of systems (windows PC's, laptops), and the typical ownership (individuals, enterprises) and management (intermittent connection, operation) gave rise to a new explosion of interest in and a new set of technical challenges for distributed computing. In addition, the growing popularity of PC-based clusters for high-performance computing [8,9], provided clear evidence that distributed computing on PC's was ready to support major high performance computational needs.

In 1996, Scott Kurowski partnered with George Woltman to answer begin a search for large prime numbers, a task considered synonymous with the largest supercomputers. This effort, the "Great Internet Mersenne Prime Search" or GIMPS [10,11]. This effort has been running for over five years, currently includes over 200,000 machines, and has discovered the 35^{th}, 36^{th}, 37^{th}, 38^{th}, and 39^{th} Mersenne primes – the largest known prime numbers. The most recent was discovered in November 2001, and is over 4 million digits. The GIMPS project was the first project taken on by Entropia, Inc, a startup commercializing distributed computing. Another group, distributed.net [12], pursued a number of cryptography related distributed computing projects in this period as well. In 1999, the best-known Internet distributed computing project SETI@home [13] began, and rapidly grew to several million machines (typically about 0.5 million active). The publicity associated with these efforts led to the funding of a number of commercial startups which variously focus on enterprise and internet distributed computing with a variety of business models.

Requirements for Distributed Computing

Distributed Computing systems begin with a collection of computing resources, heterogeneous in hardware and software configuration, distributed throughout a corporate network, and subject to varied management and use regimens and aggregate them into an easily manageable and usable single resource. Further, a Distributed Computing system must do this in a fashion that ensures that there is little or no detectable impact on the use of the computing resources for other purposes. For end users of distributed computing, we believe the aggregated resources must be presented as a simple to use, robust resource. Based on our experience with corporate end users, the following requirements are essential for a viable distributed computing solution:

Robustness – The distributed computing system must complete computational jobs with minimal failures, masking underlying resource and network failures. In addition, the system must provide predictable performance to end users.

Security – The distributed computing system must protect the integrity of the distributed computation. That is, tampering with or disclosure of the application data and program must be prevented. In addition, the distributed computing system must protect the integrity of the computing resources that it is aggregating. Distributed computing applications must be prevented from accessing or modifying data on the computing resources.

Scalability – The distributed computing system must scale to the use of large numbers of computing resources. Because large numbers of PC's are deployed in many enterprises, scaling to 1000's, 10000's, and even 100000's are relevant capabilities. However, systems must scale both upward and downward – performing well with reasonable effort at a variety of system scales.

Manageability – The distributed computing system often involve thousands to hundreds of thousands of computing resources. Because of these large numbers, the management and administration effort for the distributed computing system cannot scale up with the number of resources. Typical rules of thumb such as requiring even one administrator for every two hundred systems would be unacceptable. We believe distributed computing systems must achieve manageability that requires no incremental human effort as clients are added to the system.

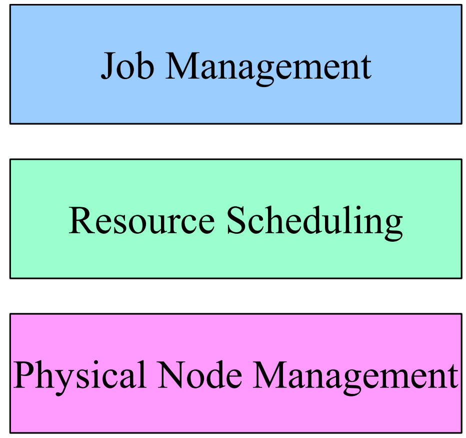
Unobtrusiveness – The distributed computing system typically shares resources (both computing, storage, and network resources) with other usage in the corporate IT environment. As a result, the usage of these resources should be unobtrusive, and where there is competition, non-aggressive. The distributed computing system must manage its use of resources so as not to interfere with the primary use of desktops by their primary owners and networks by other activities. The system must manage not only the usage due to system activities but also by all distributed computing application activities.

Openness / Ease of Application Integration – The distributed computing system is a platform which supports applications. The number, variety, and utility of the applications supported by the system directly affect its utility. Distributed computing systems must support applications developed with all kind of models and with many distinct needs with minimal development effort. We believe that distributed computing systems must support both existing and new applications.

Together, we believe these six criteria represent the key requirements for distributed computing systems.

Entropia System Architecture

The Entropia distributed computing system groups the address of the critical requirements into three key elements: physical node management, resource scheduling, and job scheduling. We briefly describe these three layers, then describe the advantages of this approach in achieving a robust, scalable distributed computing resource. The Entropia system architecture is depicted here.



Physical Node Management – The distributed computing environment presents numerous unique significant challenges to providing a reliable computing capability. Many of these challenges are met by the physical node management layer which provides communication, physical node management, security, and application control for unobtrusiveness. The communication (and naming) services allow the system and applications to communicate securely, reliably, and conveniently, despite intermittent network connectivity, dynamic IP address assignment, and network features such as firewalls. The physical node management services capture a wealth of static and dynamic information about each physical node (e.g. physical memory, CPU speed, disk size, available space, client version, data cached, etc.), reporting it to the centralized system console. In addition, the physical node management layer ensures that nodes can be recovered from runaway applications, poor application behavior, and many damaged client states. The security services employ a range of encryption and binary sandboxing technologies to protect both distributed computing applications and the underlying physical node. Application communications and data are protected with high quality cryptographic techniques. A binary sandbox controls the operations and resources that are visible to distributed applications on the physical nodes, controlling access to protect the underlying machine. Finally, the binary sandbox also controls the usage of resources by the distributed computing application. This ensures that the application does not interfere with the primary users of the system – it is unobtrusive – without requiring a rewrite of the application for good behavior.

Resource Scheduling – The distributed computing system consists of resources with a wide variety of configurations and capabilities. The resource scheduling layer accepts units of computation, matches, then schedules them on appropriate resources. Because despite the resource conditioning provided by the physical node management layer, the resources may still be unreliable (indeed the application may be unreliable in its execution to completion), the resource scheduling layer must adapt to changes in resource status, unsuccessful execution of units of computation, and despite them ensure reliable and robust execution of each unit of work. To meet these challenging requirements, the Entropia system can support multiple instances of heterogeneous schedulers. This layer also provides simple abstractions for IT administrators which automate the majority of administration tasks with reasonable defaults, but allow detailed control as desired.

Job Scheduling – Distributed computing applications often involve large amounts of computation (thousands to millions of CPU hours) submitted as single large jobs. These jobs often arise from statistical studies (i.e. Monte Carlo or Genetic algorithm), parameter sweep, or database search (bioinformatics, combinatorial chemistry, etc.). Because such a large amount of computation is involved, higher level management of the aggregate job in order to provide short, predictable turnaround times, as well as to aggregate results and the status of individual computational units is provided by the job scheduling layer. This element of the system provides a simple abstractions for end users, delivering a high degree of usability in an environment where it is easy to drown in the data, computation, and just numbers of things going on.

The three layer approach provides a wealth of benefits in system capability, ease of use by end-users and IT administrators, and for internal implementation. The modularity provided by the Entropia system architecture allows the physical node layer to contain many of the challenges of the resource operating environment. The physical node layer manages many of the complexities of the communication, security, management, allowing the layers above to operate with simpler abstractions. The resource scheduling layer deals with unique challenges of the breadth and diversity of resources, but need not deal with a wide range of lower level issues. Above the resource scheduling layer, the job scheduling layer deals with mostly conventional job management issues. Complementarily, the upper layers of the system also capture upward interface issues, shielding the lower levels of the system from these concerns. End user and IT administrator presentation is a significant focus of the resource and job scheduling layers, as ease of use is a critical characteristic for distributed computing systems. Layers such as the physical node abstraction need not concern themselves with these issues. Finally, the higher level abstractions presented by each layer support the easy enabling of applications. Applications can be enabled unaware of low level system details, yet can be run with high degrees of security and unobtrusiveness. Distributed computing versions of applications can leverage existing job coordination and management designed for existing cluster systems because the Entropia platform provides high capability abstractions, similar to those used for clusters.

Summary and Futures

Distributed computing has the potential to revolutionize how much of large-scale computing is achieved. If easy-to-use distributed computing can be seamlessly available and accessed, applications will have access to dramatically more computational power to fuel increased functionality and capability. The key challenges to acceptance of distributed computing include robustness, security, scalability, manageability, unobtrusiveness, and openness/ease of application integration.

Entropia's system architecture consists of three layers: a physical node management layer, resource scheduling, and job scheduling. This architecture, provides a modularity which allows each layer to focus on a smaller number of concerns, enhancing overall system capability and usability. This system architecture provides a solid foundation to meet the technical challenges as the use of distributed computing matures – supporting the broadening the problems supportable by increasing the breadth of computational structure, resource usage, and ease of application integration.

Acknowledgements

We gratefully acknowledge the contributions of the talented engineers and architects at Entropia whose efforts contributed to the design and implementation of the Entropia system. We specifically acknowledge the contributions of Brad Calder, Kenjiro Taura, Scott Kurowski, Shawn Marlin, and Wayne Schroeder to the early definition and development of the system architecture.

Bibliography

1. Jay Lyman, Intel Debuts 2.2Ghz Pentium 4 Chip, The News Factor (online publication), http://dailynews.yahoo.com/h/nf/20020107/tc/15654_1.html, http://www.newsfactor.com, January 2002.

2. Tim McDonald, AMD Debuts Speedy new Athlon, The News Factor (online publication), http://dailynews.yahoo.com/h/nf/20020107/tc/15639_1.html, http://www.newsfactor.com, January 2002. Introduces 1.67 Ghz Athlon processor.
3. F. Heart, A. McKenzie, J. McQuillian, and D. Walden. "***ARPANET** Completion Report*". Technical Report 4799, BBN, January 1978.
4. John Shoch and Jon Hupp, The "Worm" Programs – Early Experience with a Distributed Computation, Communications of the ACM, March 1982, Volume 25, Number 3.
5. M.J. Litzkow, M. Livny. ***Condor**--A hunter of idle workstations*. Procs. 8th IEEE Int. Conf. Distr. Computing Systems, June 1988, 104--111.
6. A. Bricker, M. Litzkow, and M. Livny. Condor Technical Summary. Technical Report TR 1069, *Computer Sciences Department*, University of Wisconsin, Madison, WI, January 1992.

7. Gray, M. 1996. Internet Growth Summary. http://www.mit.edu/people/ mkgray/net/ ***internet-growth***-*summary*.html.
8. Andrew Chien, Mario Lauria, Rob Pennington, Mike Showerman, Giulio Iannello, Matt Buchanan, Kay Connelly, Louis Giannini, Greg Koenig, Sudha Krishnamurthy, Qian Liu, Scott Pakin, Geetanjali Sampemane. *Design and evaluation* ***HPVM***-*based Windows supercomputer*. Interna- tional Journal High Performance Computing Applications, 13(3):201--219, Fall 1999.
9. T. Sterling, ***Beowulf*** *Cluster Computing with Linux*. The MIT Press, 2001.
10. George Woltman. *Great Internet* ***Mersenne*** *Prime Search*. http://www.mersenne.org/
11. *Researchers Discover Largest Multi-Million-Digit Prime Using Entropia Distributed Computing Grid.* $2^{13,466,917}-1$ *is now the Largest Known Prime*. Press Release Dec 2001, http://www.mersenne.org/13466917.htm
12. distributed.***net****: The Fastest Computer on Earth*. http://www.distributed.net/
13. W.T. Sullivan III and D. Werthimer and S. Bowyer and J. Cobb and D. Gedye and D. Anderson. *A New Major SETI Project Based on Project Serendip Data and 100,000 Personal Computers*, Astronomical and Biochemical Origins and the Search for Life in the Universe, Proceedings of the Fifth International Conference on Bioastronomy, 1997.