Decentralized Distributed Processing

Michael Tao
Thomas Jefferson High School for Science and Technology
Alexandria, Virginia

March 30, 2007

Abstract

With the enormous amount of data being collected every day, a single computer's CPU's computational ability to analyze the data and to utilize meaning behind the data is less than satisfactory. In order to mine the rough of the data within certain time constraints, a collection of computers is needed. The purpose of this project is to produce a medium for distributing the load of enormous tasks to networked peers with varying computing power in an efficient manner. This will distribute the work load from one computer to other computers within a network of peer computers by sending portions of the data and the proper analytical tools to all of the specified peers while also computing various peer's tasks. Peers can be running on multiple computer platforms such as Windows and Linux.

Keywords: High performance, Data analysis, decentralized, distributed processing

1 Introduction and Background

Though distributed servers and clusters have existed for a while, there is a lack of sharing, most distributing acts rely on a single task giver, and the peers being enslaved to the server, with little / no reciprocation As the quantity of data and complexity of analysis from individual groups becomes greater, the efficiency current distributed processing units will certainly become less than satisfactory.

Though distributed servers and clusters have existed for a while, there is a lack of sharing, most distributing acts rely on a single taskgiver, and the peers being enslaved to the server, with little / no reciprocation. As the quantity of data and complexity of analysis from individual groups becomes greater, the efficiency current distributed processing units will certainly become less than satisfactory. Some projects such as Boinc have been created to share the resources of many nodes on a certain set of registered tasks, but those nodes aren't able to share their own tasks. In the larger picture, of all of the data analyses needed to be done in the world, only a select few are being done efficiently. What if anyone could create their own analysis tool and have any number of peers aid them in generating results?

2 Development

Part of the point of this project was to give me a means to learn more about threads and networking in Java while also working on a topic that interested me. The first iterations involved learning how to get several computers to communicate "hello" messages, while the second large iteration involved making threads which are spanned in an organization similar to what I will probably use in the final iteration for this project. Currently it is able to recognize tasks being handed to it and will then start doing each task one at a time. It will initialize each task-doing client upon necessity and can start multiple clients to do the task as necessary, decided by a heuristic located within the file the task is contained in.

3 Results and Discussion

So far, the application is able of distributing itself into a server and client component. The server component will sit idle until it is given a task by a peer. There is a limited number of slots for tasks available on the computer, determined by its capabilities. I will soon write the code for transfering the data and initializing the analysis tool. The client component will involve a user interface which will wait for local users to upload tasks for the network to do. The server component's networking and communications are currently working as is the basis for the analysis but they are currently not working together.

References

[1] M. Wang, T. Madhyastha, N.H. Chan, S. Papadimitriou, C. Faloutsos, "Data Mining Meets Performance Evaluation: Fast Algorithms for Modeling Bursty Traffic"

[2] Khalil Amiri, David Petrou, Gred Ganger, and Garth Gibson, "Easing the Management of Data-parallel Systems via Adaptation", *Procedings of 9th ACM SIGOPS European Workshops & CMU-CS-99-140*, September 2000.

[3] Khalil Amiri, David Petrou, Gred Ganger, and Garth Gibson, "Dynamic Function Placement for Data-Intensive Cluster Computing", *Supercedes Carnegie Mellon University School of Computer Technical Report & CMU-CS-99-140*, June 2000.

[4] Michael Mesnier, Eno Thereska, Daniel Ellard, Gregory R. Ganger, Margo Seltzer, "File Classification in Self-* Storage Systems", *Supercedes Carnegie Mellon Univerity Parallel Data Lab Technical Report*, January 2004.

[5] Joao Pedro Sousa, David Garlan, "Aura: an Architectural Framework for User Mobility in Ubiquitous Computing Environments", *Procedings of the 3rd Working IEEE/IFIP Conference on Software Architecture*, August 2002.