Introduction

by William Stevenson

Parallel computing is the idea that one can accelerate the execution of a computer program by separating the program into segments that can be executed simultaneously, each on its own dedicated processor. Similarly, distributed computing refers to the idea that a parallel application can be run on multiple computers (called "nodes") that are interconnected by some communications medium, such as the Internet. Ideally, one could conceive of a program that when executed across *N* processors might execute *N* times faster then it would have if it had been executed as a single thread on a uniprocessor machine. In practice however, this is not exactly what happens, but nonetheless parallel and distributed computing have caused monumental gains in the performance of complex tasks. This is particularly true in the case of distributed computing applications such as the SETI@home project, (whose website is reviewed in this issue by Crossroads Reviewer Pragyansmita Paul) and other projects such as the contests of the distributed.net group [3]. The technology behind parallel computing has even come to the point that any suitably motivated (and financed) person could set up their own computing cluster using commonly available software such as Beowulf [1] or AppleSeed [2]. Furthermore, research such as that described in this issue continues to push the envelope of what is possible when one takes advantage of the advances in computer hardware coupled with the distribution power of the Internet.

We begin our investigation with Tobias Butte's research into distributed systems designed from the paradigm of mobile agents. While such technology is still in its birth stages, it most certainly will be put to good use in solving the problems of the future such as the detailed mapping of information on the Internet. Cory Quammen follows with an introduction to the art of programming parallel computers of both the shared-memory type and the distributed-memory type. Quammen sheds light on the differences between the two architectures and discusses OpenMP and MPI, two standards for programming for these different types of multiprocessors. Josiah Dykstra ties the issue up with an application for parallel computing - the verification and validation of computer code via automated theorem proving. In particular, Dykstra discusses Destiny, an application framework for theorem proving that is scalable for parallel processing.

We at Crossroads enjoyed preparing this issue for you, and invite you to visit our website at http://www.acm.org/crossroads/ to see some of the other articles on parallel and distributed computing that we hope you will find to be as interesting as we found them. We will be back this summer with our next issue which deals with the topics of Free Software and the Open Source Movement.

References

1

The Beowulf Project. http://www.beowulf.org (22 January 2002).

2

Decyk, V., Dauger, D., and Kokelaar, P. Project AppleSeed. http://exodus.physics.ucla.edu/apleseed/ (22 January 2002).

3

Distributed.net: Node Zero http://distributed.net (22 January 2002).

Biography

William Stevenson (billstevenson@acm.org, http://www.billstevenson.org) is a graduate student at Pennsylvania State University studying to get his PhD in Computer Science. His primary research interests lie in scientific computing and cognitive science, and he currently serves as Editor in Chief of ACM Crossroads Magazine. In his spare time, he enjoys reading and writing about science, and surfing the web on his PowerBook.