Implimenting A Pregel Based Multithreaded Graph Processing Library

[Extended Abstract]

Alex Ballmer
Illinois Institute of Technology
alexandersballmer@gmail.com

Figure 1: A sample black and white graphic that has been resized with the `includegraphics` command.

ABSTRACT

The emerging applications in big data science and social networks has led to the development of numerous parallel or distributed graph processing applications. The need for faster manipulation of graphs has driven the need to scale to large core counts and accross many parallel machines. While distributed memory parallel systems continue to be used for high performance computing, some newer systems make use of shared memory and larger core counts. I hope to explore ways to utilize these larger shared memory systems by implementing a framework for manipulating graphs at a large scale. This system should be able to leverage and scale to very large core counts and provide a framework for later incorporating distributed processing across multiple nodes.

CCS Concepts

•**Theory of computation** → **Shared memory algorithms;**

Keywords

Graph processing; Parallel Algorithms; Shared Memory

1. INTRODUCTION

As was the case with tables, you may want a figure that spans two columns. To do this, and still to ensure proper "floating" placement of tables, use the environment **figure*** to enclose the figure and its caption. and don't forget to end the environment with figure*, not figure!

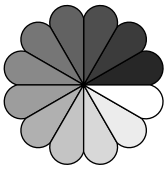

Figure 3: A sample black and white graphic that has been resized with the `includegraphics` command.

$$\log e^z = z$$

APPENDIX

ACM ISBN 978-1-4503-2138-9…$9999999.99
DOI: 10.1145/1235

Table 1: Some Typical Commands

Command	A Number	Comments
`\alignauthor`	100	Author alignment
`\numberofauthors`	200	Author enumeration
`\table`	300	For tables
`\table*`	400	For wider tables

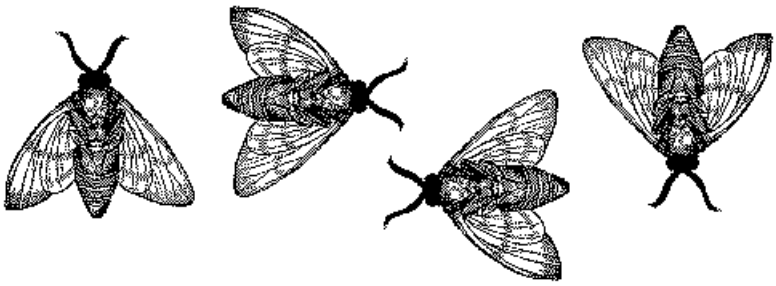

Figure 2: A sample black and white graphic that needs to span two columns of text.