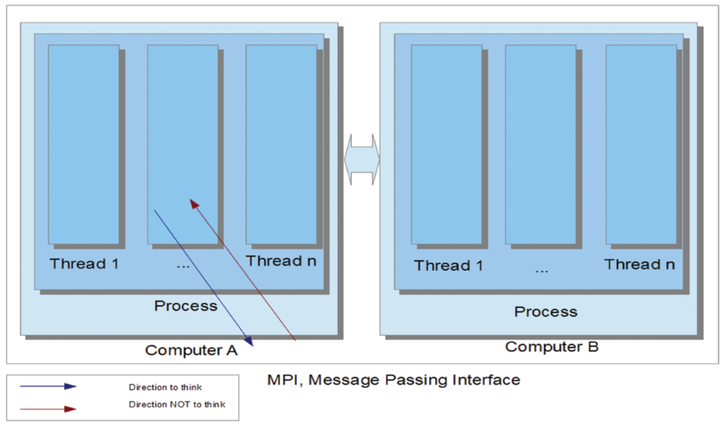
Forgotten Treasures: MPI, Algorithms, Data Structures and The C Language To Empower Distributed Web Crawler

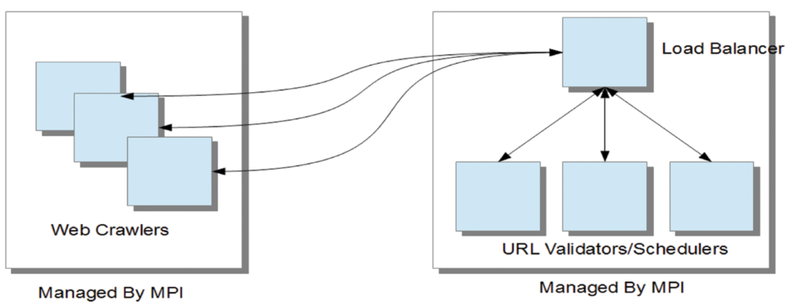
Seiya Kawashima, The University of Chicago, Radiology Department

Conclusion

a) MPI gives controls to design and manage multiple processes and machines anyway you imagine.
b) Efficiencies are not magical. They are from the right data structures and algorithms.
c) It works. It's consisted of 40,998 lines of C code so far.
d)No memory leaks so far verified by a memory detector called Valgrind.



Direction of How to think distributed computing to scale



The Architecture of Web Crawler Powered By MPI, Message Passing Interface

Introduction

This project answers the questions like below.
a) Is Map/Rduce from Google the only way to distribute tasks among computers ?
b) Would like to use your knowledge about data structures and algorithms to build something big ?
c) Do you feel that the languages, Java, C#, Node.js Ruby, Python are ok but wonder what C gives you at scale ?
d) Would you like to know how a web crawler is designed and implemented ?

Aim

a) Control threads,processes and multiple computers under MPI.
b) Think scale in a bottom up way but not top down.
c) Understand how important data structures and algorithms at scale.
d) Create zero memory leak.

Method

a) Understand the characteristics of threads and processes in C in user space and kernel space as well.
b) Understand the efficiency of text analysis including searching, storing and retrieving.
c) Optimize each component using the right data structures and algorithms
d) Use a memory detector to detect memory leaks and invalid memory access.

An example of bottom up optimization

Tag name	# of tags	Prs()	libxml2-2.9.1	jsoup-1.8.1
		Time (msec)		
<link>	14	0.525	5.714	265
	84	0.525	5.995	267
	282	0.646	6.008	265
<a>	382	0.645	6	260

Each of Prs(),libxml and jsoup reads,parses an about 79 KB HTML file and searches and counts the number occurrences of the tags.

Prs() - Implemented by the author to optimize HTML parsing as a bottom up optimization.
Libxml - The most popular XML,DOM HTML library in C.
Jsoup - The most convenient HTML library in Java as long as the author knows.

Approximate # of pages to visit a day

Crawl-Delay(sec)	# of times to crawl a day
1	86,400
30	2,880
60	1,440

Approximate number of pages to crawl a day based on web server's 'Crawl-Delay' policy. Any crawlers should not determine how often to visit a page but should be determined by the corresponding policy on their 'robots.txt' at the root paths. If the policy doesn't exist, the crawler visits every 60 seconds.

The numbers are approximate because it uses 'soft' real time as a timer to revisit the pages, so the interval is the corresponding policy or may be delayed depending on how busy it is at the moment.

Tips

a) To optimize at scale, make sure that each component is optimized before employing multiple threads,processes and machines.
b) Slow algorithms don't get any faster employing multiple threads,processes and machines.

Based on the technologies, the author is trying to create a web search engine. If you are interested in it, please contact the author at skawashima@uchicago.edu.

5841 S. Maryland Ave. | Rm. IB-012 | Chicago, IL 60637
Office: 773-834-1791