ASSIGNMENT 2: Mutlithreaded Sort

Out: 2/7/02; Due: 2/21/02
Programming Parallel and Distributed Systems
Computer Science 178, Spring 2002
Steven P. Reiss

OBJECTIVE

The purpose of this assignment is to let you experiment with the design and implementation of a program using Java threads. .

THE PROBLEM

In the previous assignment you began exploring algorithms for file sorting (sorting data files where there is not enough memory to keep the whole file in memory at once). You might have chosen a particular algorithm or tried multiple algorithms to see which ran the fastest.

In this assignment we are going to continue those efforts. In particular, we are going to see if you can use a machine with multiple CPUS (fred, sunfire, mothra) to speed up the sort and if so, how much speed-up you can achieve.

SPECIFICATIONS

Just as in the last assignment, your program will be given the name of the file to be sorted and the name of the result file on the command line. It may also be given the name of a temporary directory to be used for any intermediate files. If no temporary directory is given, the program should create a subdirectory in the current directory, use that, and remove it when done. The file to be sorted will consist of lines of text (with newline (\n) terminators). Lines are to be compared lexicographically. The result file should be in ascending order. You can assume that all lines are shorter than 1024 characters long if necessary.

Your program will take two additional arguments. The first is the number of available CPUs that should be used (-c #). The second is the maximum amount of memory (in megabytes) that the program should use (-m #). You should use this second number to estimate how much of the file can be kept in memory at any one time.

Again as last time, you can assume that the file system has at least as much free space as the twice the size of the original file. You should also learn the use of the UNIX limit command to set the maximum number of file descriptors, and the program size. When using

Java, you should know both the -Xmx option to set memory size and the -d64 option (available with Java 1.4) that runs using a 64 bit data model and hence can use more the 2G of memory if that is appropriate to the given run.

TESTING

Sample files of various sizes will be provided in */map/aux0fred/cs178/rawdata.* You can create your own directories in */map/aux0fred/cs178/temp* for testing purposes. Your goal is to see how large a file you can sort within ten minutes of clock time. On UNIX you can use the system sort routine with the -c option to check if you sorted the file correctly. You should also record the times is takes you to sort various size files within this ten minute limit.

MECHANICS

You should hand in you source code and a graph showing how your program scales run time with file size (either total size or number of lines), memory size, and the number of processors used.