Data Technologies – CSC 2009

Lectures: Alberto Pace, Bernd Panzer-Steindl

Exercises: Andreas-Joachim Peters

Email Andreas.Joachim.Peters@cern.ch

Exercises Data Technologies

2 Performance Monitoring, Measurement & OS Caching

Introduction:

In this exercise we will first write programs to evaluate the performance of reading data from disk to memory & writing data from memory to disk.

You can implement the programs in a programming language of your choice (C, C++, PERL, Python, JAVA).

Later you will see, that these type of programs are in a similar way available as Linux commands which allow similiar measurements.

Warning: not all programming languages have a simple and good handling of byte, 32-bit or 64-bit oriented operations on memory buffers. Choose the language where you are confident to be able to implement the described program in the simplest way for you! This aspect is not yet too important for the exercises in this chapter, but in chapter 3 and 4 and you might reuse some code.

Exercises

2.1

MEMORY TO DISK WRITING

Write a program

Syntax: `memwriter <filename> <N> [<io block size=4k>]`

that writes a sequential file <filename> of <N> bytes copying with variable block size from a 1M allocated memory buffer. The buffer should be initialized with zero's. For files bigger than 1M use the memory buffer as a circular buffer restarting from the beginning!

DISK TO MEMORY READING

Write a program

Syntax: `memreader <filename> <N> [<io block size=4k>]`

that reads sequentially from the file <filename> <N> bytes with variable block size into a 1M memory buffer. Also in this case use the target buffer as a circular buffer.

Do now the following performance measurements:

2.1.1 Measure the time and calculate the rate to write sequential files with 4k IO blocks with a size of 1MB, 10MB, 100MB, 1GB, 10GB into the */tmp/* directory.

Do you see linear scaling? Explain your observation!

2.1.2 Repeat the measurements for a 1 byte block size for 1 MB and 10MB files.

What percentage difference do you observe? Can you explain it?

2.1.3 Measure the time to read back the 5 files starting with 10GB size upto 1MB two times each!

How would you explain a large variation from the 1st to 2nd execution?
Are all results compatible with the performance of a single hard disk?

2.1.4 Develop a strategy to measure the read performance of your hard disk and of the OS cache separately using only the following commands: *dd, cat, vmstat, time*

How fast is your hard drive in average doing sequential reads of a 1 GB file with 4k read block size?

How fast is access from the OS cache?

Hint: as root (unfortunately your are not) you can flush the OS caches in Linux by doing:

```
sync && echo 3 > /proc/sys/vm/drop_caches
```

as a user you can flush the cache by writing a new file bigger than the physical memory. ***dd*** is a nice tool to create files of predefined size!

2.1.5 Write a program that reads 1.000 randomly chosen bytes in one of your previously created 10 GB files.

Calculate the average seek time of your hard drive.

Hint: Take care not to be fooled by the buffer cache in case you repeat a measurement with the same 'random' blocks. If you don't use a new random seed with each program start you re-read always the same blocks!