High throughput computing

Research Computing Clusters: Kure and Killdevil

Research Computing
University of North Carolina

August 21, 2014

Table of Contents

Introduction

Initial Problem

Scheduler Usage.

File system usage.

Further directions.

Conclusions.

High throughput computing

- High Throughput Computing seen as distinct from *High Performance Computing*.
- Necessarily somewhat arbitrary.
- High performance computing: How to as quickly as possible complete one large calculation.
  - Focus on parallel, complicated inter-process communication.
  - Focus on quality hardware: specialized hardware, interconnects.
- High throughput computing: How to as quickly as possible complete a large number of small jobs.
  - Focus on parallel, trivial inter-process communication.
  - Focus on quantity of hardware: commodity hardware, file system for communication.

High throughput computing

- High Throughput Computing has developed a very specific meaning in some communities.
- Will consider a more general understanding of the term.
- Two primary problems:
  - Making efficient use of the scheduler.
  - Making efficient use of the shared file system.
- A large number of inter-related smaller problems.

(CYOA 3022) Be your own boss: Huge opportunities in the emerging field of bio-organic fertilizers

- You inherit a previously uncharted guano island in the middle of Jordan lake.
- Prior to synthetic petroleum based fertilizers, guano was a valuable national resource.
- The Guano Islands Act (11 Stat. 119, enacted 18 August 1856, codified at 48 U.S.C. ch. 8 §§1411-1419) is federal legislation passed by the U.S. Congress that enables citizens of the U.S. to take possession of islands containing guano deposits.
- Today organic farmers in Chapel Hill are willing to pay handsomely.
- The problem, of course, is getting your inheritance to market.

(CYOA 3022) Be your own boss: Huge opportunities in the emerging field of bio-organic fertilizers

- Experts estimate that you have inherited approximately 10000 tons of guano.
- However, the market for bio-organic fertilizer is constantly changing. Now there is demand and the price is high. Soon that might not be the case. You must act quickly.
- You decide to rent cargo ships to transport your inheritance. There are three classes of ships:
  - Ships that can carry 12 tons of guano and sail at 2.9 knots.
  - Ships that can carry 16 tons of guano and sail at 2.6 knots.
  - Ships with specialized Guano Processing Units (GPUs) to quickly load and unload the guano and sail at 2.9 knots.
- Which ships do you choose?

(CYOA 3022) Be your own boss: Huge opportunities in the emerging field of bio-organic fertilizers

- You have significantly more guano than can be transported on any single ship. The only way this is going to get done is with a whole guano fleet.
- Jordan Lake is not exactly the Pacific Ocean. The difference in sailing speed is largely irrelevant.
- Installing specialized loading machinery to support the GPUs is time-consuming and ultimately costly. Furthermore, there aren't many ships that sail Jordan Lake with this kind of special equipment.
- What you want to know is how quickly any ship can be chartered and how many ships are available to be chartered at any given time.

(CYOA 3022) Be your own boss: Huge opportunities in the emerging field of bio-organic fertilizers

- Chartering a cargo vessel for a week takes about a day.
- Loading and unloading 12 – 16 tons takes about 2.5 days.
- Sailing to and from your island takes about a day.
- Two extremes: neither is a good approach.
  - Charter one boat and move all 10000 tons with that vessel. This minimizes the total time spent chartering vessels, but takes forever to transfer the cargo.
  - Charter 10000 boats initially and move 1 ton each. This minimizes loading, unloading, and sailing time, but takes forever to start.
- You can get about 100 - 200 vessels chartered in a couple days.
- Each vessel could do two trips in a week.
- While the first fleet is sailing you can continue to charter more.

(CYOA 3022) Be your own boss: Huge opportunities in the emerging field of bio-organic fertilizers

- This is essential a high throughput computing calculation:
  - How many cores can I get with minimal pending?
  - At what point is it better to run fewer longer jobs than pend for more resources?
- Of course, this depends at any given time on the state of the cluster.
- Basically, you want to batch up your jobs so that each batch job take 20 minutes to two hours.
- You really only want to have about 1000 - 2000 pending jobs. There is a limit for all users.

(CYOA 3022) Be your own boss: Huge opportunities in the emerging field of bio-organic fertilizers

- The key observations:
- Get nodes working as quickly as possible: minimize pending time.
- Balance concurrency against job duration. Heed the ancient proverb:
  - Happiness lies between one ship with 10000 tons of guano and 10000 ships with one ton of guano.
- Minimize other bottlenecks: primarily the file system and network. On our clusters these are essentially the same thing.

The scheduler and the kernel only like you as a friend: Keeping your spirits high and your standards low.

- Things that affect pending time:
  - Availability: Nothing you can do about this. Plan ahead prepare for the worst.
  - The more restrictive your requirements, the fewer resources you have to choose from. Requirements that increase pending time:
    - Exclusive access.
    - Multiple cores.
    - More memory.
    - Specialized file systems.
  - Aim for the largest pool of possible nodes. Avoid special requirements.

The scheduler and the kernel only like you as a friend: Keeping your spirits high and your standards low.

- Balancing concurrency with job duration:
  - Once a job starts running you have the node for up to a hour, day, or week.
  - Longer jobs don't benefit from the parallelism inherent in the cluster: this is the one ship approach.
  - As the fraction of pending to running time increases, most of your time to completion is spent doing nothing: this is the 10000 ships approach.
- The ship with 10000 tons of guano sinks, but 10000 ships with one ton of guano won't fit on Jordan Lake.
- Minimizing bottlenecks: primarily file system or network.

File system secrets of the ancients.

- In the past twenty years:
  - CPU clock speeds have increased significantly .
  - Memory speeds have increased significantly.
  - Memory capacity has increased significantly.
  - Disk capacity has increased significantly.
  - Disk speed has effectively remained the same.
- Two ways people have dealt with this:
  - Solid-state drives: no spinning disk, no magnetic media.
  - File caching: at any given time significant amount of the memory on a machine is devoted to file cache.

File system secrets of the ancients.

- Best considerations to support efficient file caching:
  - Reading is much, much cheaper than writing.
  - Reading from file cache to significantly faster than reading from disk.
  - Flush buffers when you no longer need to access the file.
  - Per core, target about 500 MB to 2 GB of files.
- Networked file system: /netscr
  - Each node has a 1 Gb network connection.
  - Each node has 8 – 16 cores that could be competing.
  - There are times when few jobs actually is actually faster than more.
- In most cases, excessive testing is the only way to determine which configuration is best.

File system secrets of the ancients.

- Networked file systems. Consider two programs.
- Program 1:

```
zcat Test1.txt.gz > Test1.txt
zcat Test2.txt.gz > Test2.txt
...
zcat Testk.txt.gz > Testk.txt
```

- Program 2:

```
zcat Test1.txt.gz > Test1.txt &
zcat Test2.txt.gz > Test2.txt &
...
zcat Testk.txt.gz > Testk.txt &
wait
```

- Which is faster?

File system secrets of the ancients.

- Program 1 Serial ($k = 4$):

```
Started at Mon Aug 18 15:32:08 2014
Results reported at Mon Aug 18 15:48:41 2014
```

- Program 2 Parallel ($k = 4$):

```
Started at Mon Aug 18 15:12:33 2014
Results reported at Mon Aug 18 15:28:38 2014
```

- Program 1 Serial ($k = 8$):

```
Started at Tue Aug 19 07:00:57 2014
Results reported at Tue Aug 19 07:24:07 2014
```

- Program 2 Parallel ($k = 8$):

```
Started at Mon Aug 18 16:22:26 2014
Results reported at Mon Aug 18 16:49:39 2014
```

File system secrets of the ancients.

- File sizes and number of files per directory: the distributor cap program.
- Consider the following simple AWK program:

```
{
  id = (NR) % max_n;
  print $0 >> ‘‘Test’’max_n‘‘/Test-’’id‘‘.txt’’;
}
```

- For which values of 100, 1000, or 10000 will this program be the fastest when applied to a file of 13GB with roughly 25 million lines?

File system secrets of the ancients.

- Note that this is the same number of lines read and the same number of lines written
- 100: approximately four minutes and 20 seconds.
- 250: approximately six minutes and 20 seconds.
- 500: approximately 13 minutes and 30 seconds.
- 1000: approximately 30 minutes.
- 10000: greater than one hour. Estimated 15 hours.

File system secrets of the ancients.

- Sometimes, an easy way to improve throughput is a hierarchical directory structure.
- Notice how home directories are structured:

  `/nas02/home/o/n/onyen`
- Goal: No directory with more than 1000 files and/or sub-directories.
- This can be surprisingly effective for such a simple idea.
- Of course, there is a law of conservation of hard work. Sometimes it makes little difference.

Welcome to obsessive-compulsives anonymous: Hello, my name is Jeff?

- It is very easy to lose track of why you are using the cluster.
- The goal is to do less work, rather than more work.
- Scale. The 2-0 and Go Principle:
  - Test with about 20% of the jobs you ultimately want to run.
  - Scale up in increments of 20%.
  - Make sure that each stage works correctly before optimization.
- Optimization. The 8-0 and Go Principle:
  - Computer time is much cheaper than analyst time.
  - Typically, 10% of the optimization time is required to achieve 90% optimal results. The remaining 90% of the time is required to achieve the last 10 %.
  - Aim to get your runs 80 – 90 % optimal.

Things probably didn't go as planned when...

- Things to use to monitor:
  - Log on to the running node and run top.

    ```
    bjobs -X
    top -H -u onyen
    ```
  - Check your CPU occupancy:

    ```
    jle -u onyen | grep RUN
    ```
- Things to use to look for:
  - Excessive disk wait. The 'D' state on top.
  - Use 'H' to turn on or off threads. Check for over-threading.

Alternative configurations. Or how to turn your family room into a garage.

- Further directions in high throughput computing.
  - Specialized scheduler. Condor.
  - Specialized file systems. Lustre.
  - Specialized file formats: HDF, BerkeleyDB, SQLite, Kyoto Cabinet.
- Wild speculation, unnatural acts, unspeakable thoughts.
  - Local /tmp.
  - Hyper/Over threading.
  - Use of a RAM disk.
  - Kernel tuning.
  - Monotasking (no pre-emptive multitasking.)

Conclusions.

- Efficient use of the scheduler.
  - Avoid pending time as much as possible.
  - Balance concurrency against job length.
  - Batch up short running jobs into longer running batch jobs.
  - Aim for batch jobs to run about 20 minutes to two hours.
- Efficient use of the file system.
  - Favor a small number of larger files over a large number of smaller files.
  - A good place to start is the hierarchical or tree like directory structure.
  - Test out several configurationt explore the effects of file caching.