Enabling High–Throughput Research on HPC Systems

Dr François Bissey
(francois.bissey@canterbury.ac.nz)

Outline

Introduction

Two types of fundamental workflow for HPC

- Large distributed memory jobs (MPI)
- High-throughput jobs

High-throughput jobs covers

- Monte-Carlo simulations
- Parameter space sweeps
- Processing of large data sets

It is characterised by doing a task over and over again with a different input.

The problem

MPI machines

Machine like the Power7 cluster at BlueFern and Fitzroy at NIWA are geared towards distributed memory jobs (MPI).

High-throughput job

Running high-throughput jobs efficiently on these systems requires a little bit of thinking ahead and care. We will "steal" the MPI workflow to achieve this. Machines like Pan in Auckland are geared more towards high-throughput and medium size MPI jobs but applying the techniques presented here could be beneficial on it as well.

Ehsan's Simulations

Ehsan is a PhD student in computer science. His problem is simulating wireless networks to improve the efficiency of the communication in mobile devices.

- first submission to the cluster was 40,000 simulations
- done 4 at a time
- first reaction: increase the number of simulations he can run (after all they are not big MPI jobs)
- Bad idea: the simulations were spread all over the cluster blocking big MPI jobs
- developed a custom solution to contain him to a set of nodes
- we later found out that it was equivalent to using MPMD

We now identify high-throughput project quickly and offer them better solutions as soon as possible.

Solutions

Presentation

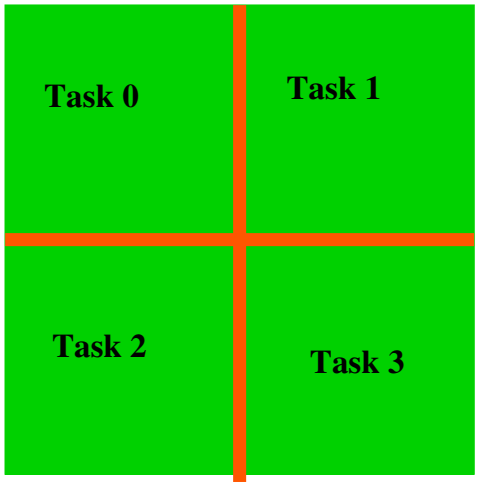
There are two main ways of packaging serial jobs to pass them as MPI distributed memory job, we'll look at them through the work of two researchers.

1. MPMD (Multiple Program Multiple Data): Nick's network
2. Batcher: Jing Wang DNA matching work

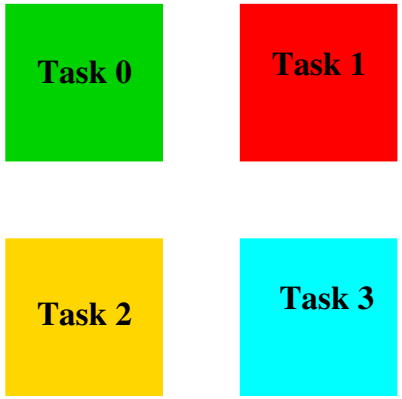
Nick's Networks

- Nick Baker is a PhD student in biological sciences with a background in mathematics. He studies ecological networks (complex food chains).
- His work focuses on analysing networks to find "keystone" species that are essential to the network. The work is what we call embarrassingly parallel, he has to go through a high number of networks and perform his analyses on each and everyone of them.
- I spotted him on the cluster running four jobs at a time out of close to a hundred. There was definitely room for improvement. He became a test subject for MPMD

MPMD: Multiple Program Multiple Data



SMPD: Instances of a program process the data distributed amongst them.



MPMD: Programs, not necessarily identical, process distinct data.

MPMD jobs at BlueFern (and Fitzroy)

At BlueFern and Fitzroy such jobs can be launched with the following

```
poe -procs N -pgmmodel mpmd -cmdfile cmdfile.txt
```

Where *cmdfile.txt* is a list of commands to be executed

```
program1 data1
program2 data2
...
programN dataN
```

Note that the number of commands matches the number of core requested. Now, Nick usually processes 32 networks per jobs instead of one.

Jing Wang, DNA matching with batcher

- Jing Wang is a researcher at ESR and came to us for several NeSI research projects
- The problem is to compare strands of DNA against a database to match sequences of genes
- The task can be further broken down by splitting the strands in sections and searching for match in these.
- Searching for sequences in one long DNA strand can take a long time. Doing it on a section is much quicker but now you have to do it on potentially thousands of sections. This has become a high-throughput task.

Batcher: Master and Worker

A batcher program uses a different MPI mode: the ability for one program to be a master that spawn workers to do various jobs on demand.
The batcher program like the MPMD will take a list of tasks. In MPMD all tasks are started at once but not with batcher. If you give batcher more tasks than you have cores, it will fill the cores given and as tasks finishes it will start the next one in the list until the list is finished.
Instead of building packets to match the number of cores requested (MPMD) you can package in units that makes sense.

Batcher jobs at BlueFern

At BlueFern such jobs can be launched with the following

```
poe batcher cmdfile.txt
```

Where *cmdfile.txt* is a list of commands to be executed

```
program1 data1 >> out1.log
program2 data2 >> out2.log
...
programN dataN >> outN.log
```

Pros and Cons

Batcher

Pros:

- Can group any number of tasks
- tasks that complete quickly will be replaced by new tasks while the longer one finishes

Cons:

- one core is used by batcher and not available to do tasks

MPMD

Pros:

- all cores are used

Cons:

- if some tasks are much longer than other some cores will be idle
- number of tasks must be equal to number of cores

Conclusions

- Lots of serial jobs are not a good thing on a machine tuned for MPI jobs
- We solve this by packing the serial jobs in a way that make look like MPI jobs
- It increases your throughput at BlueFern and Fitzroy
- It make the life of your cluster administrator easier
- MPMD is best suited for tasks of equal length (lattice QCD, Monte-Carlo)
- batcher is best suited for the rest
- Jobs can also be packaged on the Pan cluster in Auckland but the MPMD submission would be different.

Questions & Answers