High Throughput Computing at Indiana University

Abhinav Thota
Scientific Applications and Performance Tuning
Research Technologies
Indiana University

July 11, 2014
Outline

• Recap:
  • What is HTC?
  • Tools available
• Exercises
What is high throughput computing or HTC?

- Large number of independent serial or parallel jobs
- Generally in the hundreds or thousands
- Parameter sweeps, multiple branches of the same problem, etc.
- Traditionally run through high throughput grids
  - Examples include Open Science Grid (OSG), Condor, Boinc, Folding@home
- You can get started on any of these grids
  - Volunteer computing
  - If you are interested, can contribute spare desktop or Xbox/PS3 cycles
Where do you do this at IU?

• Quarry is best suited for this kind of workload
• The nodes are shared among users
  • If you request just one core on a node, the rest of the cores are used by someone else
• Just keep submitting single core jobs
• But Quarry has only so many cores
  • 218 nodes with a total of 1744 cores
    - 8 or 12 cores, 16 or 24 GB of memory
• You may on the other hand need more than what Quarry can offer you
• Big Red II is the other option
  • BR 2 is not designed for this purpose though
  • But there are ways to make this work
What’s so important about using all the cores on a node?

- Intra-node parallelism
- Quarry has 8 or 12 cores per node
  - All the nodes are shared, so cores are not wasted
- Nodes on BR 2 are not shared
  - 32 or 16 cores per node
  - Whether you use 1 core or 32 cores, you own the node
  - You will be wasting 95% of the node if you use only a single core
  - This is a big problem!
- Please be a good citizen and don’t do this!
On Big Red II

- Big Red II has more than 21,000 cores available
  - So that is attractive
- If you have a large bunch of serial jobs that you need to run, use all the cores on a node
- Let’s see how to do that
- Contact us if you are in doubt!
Tools available

- Bash scripting on Quarry or BR 2
  - This can be easily accomplished with some creative scripting within your PBS job script
- PCP on BR 2
  - We have a tool called PCP (parallel command processor) that takes care of the scripting (some if it) for you
- BigJob for more complex workflows
  - We also have Bigjob, comes with the python module on BR 2 and Quarry
  - If you have more complex workflows, where your jobs need to interact with each other, wait on output from one job to use it as input for a different job
  - Do load balancing between multiple jobs, etc.
- Swift
Bash Scripting

- You basically put all the commands you want to run in a file

- And run that file, as simple as this:
  - aprun -n 1 ./test_32.sh
  - Where test_32.sh is:
    #!/bin/bash
    ./binary > test_32.out1 &
    ./binary > test_32.out2 &
    ./binary > test_32.out3 &
    .
    .
    .
    ./binary > test_32.out32 &
    wait
PCP – Parallel Command Processor

• This is a nice little tool that lets you run multiple serial jobs on a single node
• Not all that different from bash scripting when it comes down to it
• You still have to create a script file containing all the commands that you want to run
• This works across nodes
• You don’t have to background all the command and wait for them to return
• Going to look something like this:
  • aprun –n 32 pcp script.sh
  • Where script.sh will look the same as the previous script, just without the &’s and wait at the end
• Will not work with applications that need CCM
Task 1: Use bash scripting to run 32 jobs on a single node

1. Generate a script containing the 32 commands that you want to run, without typing it yourself
   1. If you are looking for something to run, run this command 32 times, generates (pseudo) random numbers:
      - echo $RANDOM > output1.txt
      - Print 32 of these, while incrementing the output(i).txt file
      - Copy and paste the commands into a text file
      - Change the permissions of the text file to make it an executable
      - Write a PBS script
      - Submit the PBS script

   • Hint: The command is going to look something like this:
     • aprun -n 32 list32.txt
Task 2: Use pcp to run 128 jobs on 4 nodes

1. Generate a script containing the 128 commands that you want to run, without typing it yourself
   1. If you are looking for something to run, run this command 128 times, generates (pseudo) random numbers:
      - `echo $RANDOM > output1.txt`
      - Print 128 of these, while incrementing the output(i).txt file
      - Copy and paste the commands into a text file
      - Change the permissions of the text file to make it an executable
      - Write a PBS script
      - Submit the PBS script; make sure you load the pcp module!

   • Hint: The command is going to look something like this:
      • `aprun -n 128 pcp list128.txt`
Questions?