HIPAA Environment, AI and NLP Services at CHPC

Sean Igo
Center for High Performance Computing
Sean.Igo@utah.edu
September 23, 2010
Purpose of Presentation

• Brief overview of CHPC environment / access
• Overview of HIPAA-compliant secure environment
• Very brief overview of NLP / ML
• Presentation of available NLP/ML/etc software
  – Concentrating on command line utilities on batch-scheduled systems.
  – Interactive software is available
  – Programming to use parallelism
Overview of CHPC Resources

• **CHPC's mission:** Support Research!
  – Generally faculty/staff research

• CHPC can help if:
  – You need parallel-processing
  – You need remote access to a single high-powered computer
  – You have a large amount of data to process
  – You need an application you don't have on your computer
  – Your data is IRB-governed PHI
  – You have other needs...
Overview of CHPC Resources

- **Computational Clusters** – Arches (& Homer)
  - CHPC's specialty though other things available

- **Home directory** – separate for HIPAA & clusters
  - Based on your uNID
  - generally not backed up (there are exceptions)

- **Scratch systems** – cleaned after 60 days
  - some global to all HPC systems, some local for each cluster

- **Other services** -
  - Interactive, VM, Web, Grid, etc.
HPC Terms

- **Node** - a single computer (high-powered linux PC)
- **Core** - one hardware processor
  - nodes at CHPC have 2-12 cores
- **Cluster** - a set of nodes with high speed network connections and software to run in parallel
- **Job** - a batch processed command run on a cluster
- **Queue** - mechanism for ordering jobs
- **Interactive Node** - node you log on to to prepare jobs and submit them to the queue
Cluster Environment

Recommended approach:

- Edit/compile on interactive
- Create script for batch
- Submit script to scheduler
- Script does this:
  - Copy data to scratch
  - Process data
  - Write output to scratch
  - Copy output to account
  - Clean up scratch
CHPC Clusters

- General Clusters
  - delicatearch, tunnelarch, sanddunearch

- Restricted Clusters
  - updraft (256 nodes / 2048 cores), partially restricted
  - landscapearch, telluride

- HIPAA Environment - high-security for protected health info.
  - apexarch / homer – 5 nodes, 44 cores (more to come)
  - No general-access interactive node (real soon now)
  - Windows interactive nodes owned by researchers
Batch System

- All use of compute nodes go through a batch system using Torque (PBS) and Maui (Moab) for scheduling
  
  ```bash
  #PBS -S /bin/csh
  #PBS -l walltime=24:00:00,nodes=1:ppn=2
  ```

- Login (interactive) nodes only for prepping input files, analyzing results, compilations, etc.
  - no running of jobs – 15 min MAX
  - Except on HIPAA cluster...

- Walltime limits in place – usually 72 hours
Getting Started at CHPC

• Account application
  – online: https://www.chpc.utah.edu/apps/profile/account_request.php
  – You need to be doing research for a PI with a CHPC account
    • If your PI isn't signed up, let me know
  – If the online application is unsuitable, paper one is available
    • www.chpc.utah.edu/docs/forms/application.html

• HIPAA environment access
  – Application not yet automated
  – Mention in application notes that you need HIPAA
  – Access based on IRB number
Getting Started at CHPC

- Interactive nodes
  - 2/cluster (cluster.chpc.utah.edu) with round-robin access to divide load
  - HIPAA environment differs

- CHPC environment scripts
  - .tcshrc or .bashrc depending on what you prefer - edit to include apps

- Getting started guide
  - www.chpc.utah.edu/docs/manuals/getting_started

- Problem reporting system
  - http://jira.chpc.utah.edu or email to issues@chpc.utah.edu
Getting Started at CHPC

- HIPAA environment
- Hardware organization different
  - All interactive nodes owned by research groups
    - Plan to add a general interactive node RSN
  - Windows and Linux interactives; windows much like your desktop
- Same applications available on linux / cluster
- Different home directory filesystem
  - Additional security; physically secure backups, etc.
Security Policies (1)

• No clear text passwords - use ssh and scp
• Do not share your account under any circumstances
• Don’t leave your terminal unattended while logged into your account
• Do not introduce classified or sensitive work onto CHPC systems
  – Except Homer the HIPAA cluster...
• Use a good password and protect it – see gate.acs.utah.edu for tips on good passwords
Security Policies (2)

- Do not try to break passwords, tamper with files, look into anyone else’s directory, etc. – your privileges do not extend beyond your own directory
- Do not distribute or copy privileged data or software
- Report suspicions to CHPC (security@chpc.utah.edu)
- Please see http://www.chpc.utah.edu/docs/policies/security.html for more details
- HIPAA-protected data subject to additional policies
Access to (linux) Interactive Nodes

• Need ssh client (recommend PuTTY)
  – http://www.chiark.greenend.org.uk/~sgtatham/putty/
  – cygwin ssh also works fine
  – As should Mac terminal ssh

• For Xwindowing –
  – XWin32 – free through OSL (www.osl.utah.edu)
  – Macs can do it natively (?)
  – Cygwin X server works too

• Access to HIPAA environment needs VPN from off campus
Default Login Scripts

- CHPC maintains default login scripts that set up environment for batch commands and many programs (not on HIPAA)
  - www.chpc.utah.edu/docs/manuals/getting_started/code/chpc.tcshrc
  - www.chpc.utah.edu/docs/manuals/getting_started/code/chpc.bashrc
- Copy to your home directory as .tcshrc or .bashrc
  - This is being done on new accounts
- Can comment out setups for packages not used
- Currently NLP/ML packages not included
- Can customize by creating .aliases file that is sourced at end of the CHPC script
Location of Software

- Currently we place most installations at:
  - /uufs/chpc.utah.edu/sys/pkg
    - Accessible on clusters and some desktops
    - all NLP / ML software is here
- HIPAA environment also uses the chpc.utah.edu tree
- That is for linux only; Windows interactive nodes have software installed natively
General Information

• Available on CHPC web pages
• Can get to from www.chpc.utah.edu
  – -> Software documentation -> More (for nearly complete list)
  – http://www.chpc.utah.edu/docs/manuals/software
• Available for most packages
  – Not NLP/ML packages, at the moment
  – A bit dated; moving to CHPC wiki (wiki.chpc.utah.edu)
• Has information on licensing restrictions, example batch scripts, where to get more information on a specific package
• Also has useful information on running of jobs
NLP Overview

- Get computers to understand human language
  - current CHPC software limited to text, not speech
  - also limited to understanding, not generation
- Started in the 1950s
- Turned out to be much harder than it seemed
- Now NLP consists mainly of distinct tasks
  - Some of the most common supported at CHPC
Typical Initial NLP Pipeline

Input Text

The man's dog was brown.

Tokenizer

The, man, 's, dog, was, brown, .

Tok. Stream

POS Tagger

Tok / Tag Str

The_DT, man_NN, 's_POS, dog_NN, was_VBD, brown_JJ, ._.

http://www.chpc.utah.edu
NLP Software at CHPC

• Stanford Tools:
  – Tagger, NER, Parser

• Berkeley Parser

• Pipeline development: UIMA and Eclipse

• Sundance (U of U Information extraction)

• Metamap (Biomedical NLP)
MetaMap – Biomedical NLP

- `/uufs/chpc.utah.edu/sys/pkg/metamap/std`
  - Current version metamap09v2
  - 09v2 is installed and has java api

- I have done parallelized runs with this
- Same as the online MetaMap utility
- To use, you must sign UMLS agreement
MetaMap – Biomedical NLP

Processing 00000000.tx.1: common cold

Phrase: "common cold"
Meta Candidates (6):
  1000 C0009443:Common Cold [Disease or Syndrome]
    Cold
  861 C0009264:Cold (Cold Temperature) [Natural Phenomenon or Process]
  861 C0205214:Common [Functional Concept, Quantitative Concept]
  861 C0234192:Cold (Cold Sensation) [Physiologic Function]
  827 C1949981:Colds [Pharmacologic Substance]
Meta Mapping (1000):
  1000 C0009443:Common Cold [Disease or Syndrome]

- Can e.g. extract CUI from output
- Output can be XML (structured but VERY verbose)
UIMA and Eclipse

• UIMA is NLP pipeline framework
• Eclipse:IDE w/ UIMA integration
• Installed at:
  /uufs/chpc.utah.edu/sys/pkg/eclipse/std
  /uufs/chpc.utah.edu/sys/pkg/uima/std
• Requires X-Windows server to be running
  – and ssh -Y to forward X calls from CHPC
Quick ML overview

• Typical Machine Learning task: Classification
• Gather a large number of training examples
• Have human experts classify them
• Represent as feature vectors
• ML algorithm trains a model
• Classify novel instances with the model
Quick ML overview

<table>
<thead>
<tr>
<th>outlook</th>
<th>temperature</th>
<th>humidity</th>
<th>windy</th>
<th>play</th>
</tr>
</thead>
<tbody>
<tr>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>FALSE</td>
<td>no</td>
</tr>
<tr>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>TRUE</td>
<td>no</td>
</tr>
<tr>
<td>overcast</td>
<td>hot</td>
<td>high</td>
<td>FALSE</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>mild</td>
<td>high</td>
<td>FALSE</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>cool</td>
<td>normal</td>
<td>FALSE</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>cool</td>
<td>normal</td>
<td>TRUE</td>
<td>no</td>
</tr>
<tr>
<td>overcast</td>
<td>cool</td>
<td>normal</td>
<td>TRUE</td>
<td>yes</td>
</tr>
<tr>
<td>sunny</td>
<td>mild</td>
<td>high</td>
<td>FALSE</td>
<td>no</td>
</tr>
<tr>
<td>sunny</td>
<td>cool</td>
<td>normal</td>
<td>FALSE</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>mild</td>
<td>normal</td>
<td>FALSE</td>
<td>yes</td>
</tr>
<tr>
<td>sunny</td>
<td>mild</td>
<td>normal</td>
<td>TRUE</td>
<td>yes</td>
</tr>
<tr>
<td>overcast</td>
<td>mild</td>
<td>high</td>
<td>TRUE</td>
<td>yes</td>
</tr>
<tr>
<td>overcast</td>
<td>hot</td>
<td>normal</td>
<td>FALSE</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>mild</td>
<td>high</td>
<td>TRUE</td>
<td>no</td>
</tr>
</tbody>
</table>

Classic toy ML problem – given the weather, should I play tennis?

(Decision support!)
Quick ML overview

@relation weather.symbolic

@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal}
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,hot,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,mild,high,FALSE,yes
rainy,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,mild,normal,FALSE,yes
sunny,mild,normal,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no
Quick ML overview

ML software trains a model

Here, J48 decision tree

J48 pruned tree

outlook = sunny
|   humidity = high: no (3.0)
|   humidity = normal: yes (2.0)

outlook = overcast: yes (4.0)

outlook = rainy
|   windy = true: no (2.0)
|   windy = false: yes (3.0)
ML Software at CHPC

- Weka
  - Multi-algorithm ML package; java
- SVMLight, SVMLin, SVMLib
  - Support Vector Machine software / library
- Orange
  - Python extension for ML
- MegaM, VW
Other Software at CHPC

- **R 2.8.1, 2.11.0**
  - Not parallelized versions; if you need it let me know

- **Matlab up to R2010a + many toolboxes / DCS**
  - single node 8-core shared-memory parallel
  - 64 proc. distributed memory (embarrassingly parallel only)

- **Birdsuite 1.5.3**

- **Python**
  - 2.6.4 with orange, numpy, scipy (broken?)
  - 2.6.5 numpy, scipy, matplotlib
Parallelizing Your Task

• Most packages I've shown are single-processor!
  – Dedicated parallel software would be nice...
  – Can write your own, not always practical

• Different ways to run existing sw on parallel clusters

• Many tasks are “embarrassingly parallel”
  – i.e., capable of being decomposed into sub-problems that do not depend on one another
  – e.g., parsing several text files; can just send $1/n$ of the files to parsers running on $n$ processors
Parallelizing Your Task

• Embarrassingly Parallel is nothing to be ashamed of!

• However, dividing tasks naively may not give each processor the same amount of work

• That is, 1/n of the documents/sentences/etc. may not be 1/n of the work

• Initial tests with MetaMap show that dividing over documents is a terrible way to use it
Parallelizing Your Task

• Slightly better: “deal” tasks to processors as they become available
  – with fine granularity should be very efficient

• Other things to consider:
  – Per-invocation overhead
  – Integrating results

• I'm creating scripts to wrap / distribute software calls (command lines)
  – Talk to me if you need something like this
Finally...

- Let us know if there is some other package that does something that our current packages do not; we can look into the possibility of getting it.
  - Factors: cost, hardware/OS requirements, licensing
  - Also new approaches to running existing software!

- Any questions – contact me!
  - Sean.Igo@utah.edu
  - Phone: (I don't have a phone!)
  - Office: 405-16 INSCC (also a BMI cubicle)
Questions?