Computational Linguistics Masters program est. 2001
Department of Linguistics established in 1963

25 ~ 30 new CLMS students each Fall

Classes can be attended on-line

Departmental cluster (~100 nodes) runs Condor (7.8.8)

Most class assignments and projects must use Condor

http://www.compling.uw.edu/
CLMS Courses using Condor

- LING 473: Computational Linguistics Fundamentals
- LING 570: Shallow Processing Techniques for Natural Language Processing
- LING 571: Deep Processing Techniques for Natural Language Processing
- LING 572: Advanced Statistical Methods in Natural Language Processing
- LING 573: Natural Language Processing Systems and Applications
Information Retrieval

Knowledge Base Population Task
Text Analysis Conference (NIST.gov)
TAC 2010
$ condor_submit myjob.cmd

```
universe    = vanilla
executable  = /usr/bin/python
getenv      = true
input       = myinput.in
output      = myoutput.out
error       = myerror.err
log         = mylogfile.log
arguments   = "myprogram.py -x"
transfer_executable = false
queue
```

The system will send you email when your job is complete.
Grading Student Programs

- Issues
  - Student programs must run using Condor as another user (TA)
  - Rubric points for “Runs as-is” and related requirements
  - Students don’t have a way to run their job that way

- Solutions
  - Dedicated grading user accounts
  - Scripts to run the jobs
  - Student accessible checking program

- CheckIt!

  $ ~/ling572_00/bin/check_it project1 <project1.tar >results.html
  $ lynx results.html
CheckIt! project2 for jimwhite
Copied 1365456 bytes successfully.

tar xf /home2/ling572_01/project1/jimwhite_8852931575133087009

Contents

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF.java</td>
<td>3371</td>
</tr>
<tr>
<td>TF.class</td>
<td>4466</td>
</tr>
<tr>
<td>compile.sh</td>
<td>24</td>
</tr>
<tr>
<td>error.txt</td>
<td>0</td>
</tr>
<tr>
<td>log.txt</td>
<td>1125</td>
</tr>
<tr>
<td>run.sh</td>
<td>37</td>
</tr>
<tr>
<td>TF$1.class</td>
<td>949</td>
</tr>
<tr>
<td>output.txt</td>
<td>3059227</td>
</tr>
<tr>
<td>condor.cmd</td>
<td>124</td>
</tr>
<tr>
<td>readme.txt</td>
<td>855</td>
</tr>
</tbody>
</table>

Submission Inventory

<table>
<thead>
<tr>
<th>Item</th>
<th>Present?</th>
<th>OK?</th>
<th>Pattern</th>
<th>Full Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exec</td>
<td>yes</td>
<td>ok</td>
<td>run.sh</td>
<td>project1/jimwhite_8852931575133087009.dir/content/run.sh</td>
</tr>
<tr>
<td>Condor</td>
<td>yes</td>
<td>ok</td>
<td>condor.cmd</td>
<td>project1/jimwhite_8852931575133087009.dir/content/condor.cmd</td>
</tr>
<tr>
<td>Compile</td>
<td>yes</td>
<td>ok</td>
<td>compile.sh</td>
<td>project1/jimwhite_8852931575133087009.dir/content/compile.sh</td>
</tr>
<tr>
<td>Output</td>
<td>yes</td>
<td>ok</td>
<td>output.txt</td>
<td>project1/jimwhite_8852931575133087009.dir/content/output.txt</td>
</tr>
<tr>
<td>README</td>
<td>yes</td>
<td>ok</td>
<td>(?i)readme.(txt</td>
<td>pdf)</td>
</tr>
</tbody>
</table>

Running Condor Job

/condor/bin/condor_submit condor.cmd
Submitting job(s).
1 job(s) submitted to cluster 111871.

/condor/bin/condor_wait -wait 3600 log.txt
All jobs done.

Job Results: Log (log.txt)

000 (111871.000.000) 08/09 08:24:29 Job submitted from host: <192.168.100.50:53229>
    ...
001 (111871.000.000) 08/09 08:24:29 Job executing on host: <192.168.100.51:52838>
    ...
006 (111871.000.000) 08/09 08:24:38 Image size of job updated: 1
    3 - MemoryUsage of job (MB)
    2076 - ResidentSetSize of job (KB)
    ...
Job Results: Error (error.txt)

Empty

Job Results: Output (output.txt)

the 4388031
 a 1909523
to 1893178
of 1888349
and 1759666
in 1486078
that 814646
for 793612
is 712493
on 564756
by 559398
with 512396
he 494957
it 484400
at 463586
said 442322
was 439816
as 431831
his 373389
but 347712
be 337067
from 328710
are 328488
have 314716
i 307228

tecial 1
athenaeum's 1
encrusting 1
apostolidis 1
farsints 1
beatlemaniac 1
stelmakhova 1
rosser's 1
kafandaraki 1
tapahura 1

Condor Job Completed

This tar file conforms to the "Runs As-Is" rubric for the Condor Job portion of Project 1. This version of CheckIt! does not yet test your compile.sh (if any). Note that this is not any sort of check on whether your output is correct. Also note that if the file inventory showed missing items that you intend to include (such as README), then you should fix that before submitting.
Writing Condor Programs

flexible.job

```bash
file_ext = $(depth)_${gain}
universe = vanilla
executable = /opt/mono/bin/mono
getenv = true
output = acc_file.$(file_ext)
error = q4.err
log = q4.log
arguments = "myprog.exe model_file.$(file_ext) sys_file.$(file_ext)"
transfer_executable = false
```

$ condor_submit -append "depth=20" -append "gain=4" flexible.job

versus

$ mono myprog.exe model_file.20_4 sys_file.20_4 >acc_file.20_4
// If there are environment variables you want to copy from the current process, use clone_environment:
// gondor.clone_environment('PATH', 'ANT_HOME', 'JAVA_HOME')
// If you want to copy *all* of the current environment variables, omit the variable names (not recommended):
// gondor.clone_environment()

gondor.environment =
    [PATH:"/usr/local/bin:/bin:/usr/bin:/opt/git/bin:/opt/scripts:/condor/bin"
     , LC_COLLATE:'C'
     ]
workspace_dir = new File('/home2/jimwhite/workspace/parsers')

// Each parser has its own binary, but we'll use the one in base for them all.
bllip_dir = new File(workspace_dir, 'base/bllip-parser')

ensemble_dir = new File(workspace_dir, 'ensemble')

ycorpus_dir = new File(workspace_dir, 'ycorpus')

// first-stage/PARSE/parseIt -l399 -N50 first-stage/DATA/EN/ *
parse_nbest = gondor.condor_command(new File(bllip_dir, 'first-stage/PARSE/parseIt'), ['-K.flag', '-l400.flag', '-N50.flag', 'model.in', 'input.in'])

// second-stage/programs/features/best-parses" -l "$MODELDIR/features.gz"
"$MODELDIR/$ESTIMATORNICKNAME-weights.gz"
rerank_parses = gondor.condor_command(new File(bllip_dir, 'second-stage/programs/features/best-parses'), ['-l.flag', 'features.in', 'weights.in', 'infile.in'])
Universe   = vanilla
Environment= PATH=/usr/local/bin:/bin:/usr/bin:/opt/git/bin:
             /opt/scripts:/condor/bin;LC_COLLATE=C
Executable = /home2/jimwhite/workspace/parsers/base/bllip-parser/
             second-stage/programs/features/best-parses
Arguments  = -l $(_features) $(_weights)
Log        = jimwhite__home2_jimwhite_workspace_parsers_base_bllip-
             parser_second-stage_programs_features_best-parses.log
Input      = $(_MyJobInput)
Output     = $(_MyJobOutput)
Error      = $(_MyJobError)
Request_Memory=5*1029
Notification=Error
Queue
['brown-train.mrg'].each { String file_path ->
    ensemble_dir.eachFileMatch(~/parser_.*/) { File parser_dir ->
        def PARSER_MODEL=new File(parser_dir, 'first-stage/DATA/EN/')
        def MODELDIR=new File(parser_dir, 'second-stage/models/ec50spnonfinal')
        def ESTIMATORNICKNAME='cvlm-l1cl0P1'
        def RERANKER_WEIGHTS = new File(MODELDIR, ESTIMATORNICKNAME + '-weights.gz')
        def RERANKER_FEATURES = new File(MODELDIR, 'features.gz')
        
        def sysout_dir = new File(parser_dir, 'tmp/parsed')
        sysout_dir.deleteDir()
        sysout_dir.mkdirs()
        
        def nbest_output = new File(sysout_dir, file_path + '.nbest')
        def reranker_output = new File(sysout_dir, file_path + '.best')
        
        def charniak_input = new File(ycorpus_dir, file_path + '.sent')
        
        parse_nbest(model:PARSER_MODEL, input:charniak_input, outfile:nbest_output)
        rerank_parses(features: RERANKER_FEATURES, weights: RERANKER_WEIGHTS,
                      infile:nbest_output, outfile:reranker_output)
    }
}
['brown-train.mrg'].each { String file_path ->
  ensemble_dir.eachFileMatch(~/parser_.*/) { File parser_dir ->
    def PARSER_MODEL=new File(parser_dir, 'first-stage/DATA/EN/')
    def MODELDIR=new File(parser_dir, 'second-stage/models/ec50spnonfinal')
    def ESTIMATORNICKNAME='cvlm-l1c10P1'
    def RERANKER_WEIGHTS = new File(MODELDIR, ESTIMATORNICKNAME + '-weights.gz')
    def RERANKER_FEATURES = new File(MODELDIR, 'features.gz')

    def sysout_dir = new File(parser_dir, 'tmp/parsed')
    sysout_dir.deleteDir()
    sysout_dir.mkdirs()

    def nbest_output = new File(sysout_dir, file_path + '.nbest')
    def reranker_output = new File(sysout_dir, file_path + '.best')

    def charniak_input = new File(ycorpus_dir, file_path + '.sent')

    parse_nbest(model:PARSER_MODEL, input:charniak_input, outfile:nbest_output)
    rerank_parses(features: RERANKER_FEATURES, weights: RERANKER_WEIGHTS, infile:nbest_output, outfile:reranker_output)
My Development Principles

- Work Independently
  - Hip, Hip, Hooray for Leo Singer and LIGO!
    HTCondor MacPort:
    ```
sudo port install htcondor
sudo port load htcondor
    ```
- Brevity is Beautiful
- Don’t Repeat Yourself (DRY)
- Integrate with Other Current and Future Compute Systems
- Compile-time Provenance
DRMAA Java Binding

- Been Around a Long Time
- Supported by Many Systems
- Constrain the Design to Ease Future Interoperability
- New Implementation for Condor sans JNI
  [https://github.com/jimwhite/condor-jrmaa](https://github.com/jimwhite/condor-jrmaa)
  - Generates Submit Description Files and uses condor_submit
- DAGman Workflow Extension
  - Generates DAGman DAG File (and Submit Files)
  - Uses DRMAA and pretends all jobs succeed
  - Add Dependency Method:
    ```java
    void addToParentJobIds(String childJobId, String parentJobId);
    ```
import org.ifcx.gondor.Command

@groovy.transform.BaseScript org.ifcx.gondor.WorkflowScript workflowScript

def parse_nbest = command(path:'first-stage/PARSE/parseIt') { 
    flag "-K" ; flag "-1400" ; flag "-N50"
    infile "model"
    infile "input"
    outfile "output"
    jobTemplate { softRunDurationLimit = 100 }
}

def rerank_parses = command(path:'second-stage/programs/features/best-parses') { 
    flag '-l' ; infile 'features' ; infile 'weights' ; infile 'stdin' ; outfile 'stdout'
}

def modelFile = new File("model.dat")
def inputFile = new File("input.txt")
def parsedFile = new File("output1.ptb")
def p = parse_nbest(n:15, model:modelFile, input:inputFile, output:parsedFile, m:2)

def RERANKER_FEATURES = new File('RERANKER_FEATURES')
def RERANKER_WEIGHTS = new File('RERANKER_WEIGHTS')

def reranker_output = new File("best_parse.ptb")

(parse_nbest(model: modelFile) << new File("in2.txt")) | 
    rerank_parses(features: RERANKER_FEATURES, weights: RERANKER_WEIGHTS) >> new File("out2.tree")
The Road Ahead for Gondor

- Self-describing Command Line Scripts
- Dynamic SubDAG Workflow Scripts
- Persistent Workflow Results
- Workflow Reduction
- Provenance
- Reproducible Research
Workflow Persistence & Reduction

- Put Everything in Git
- All Intermediate Artifacts including Condor Control Files
- Previous Results Reused If Desired Based on Object IDs
  - See for example Nix – The Functional Package Manager
    https://nixos.org/nix/
- File Transfer via Pull or Push As Desired
- `git-annex` (or similar) for very big blobs
Fig. 1. Mapping of Git operations to PROV concepts. Note that the Activity Start and End concepts of PROV are not depicted, and correspond to, respectively, the author time and the commit time of each commit.
Thank You!

http://depts.washington.edu/newscomm/photos/the-spring-cherry-blossoms-in-the_quad/