WID Computational Clinic

An Overview of Available Resources
Overview

- Core Computational Technology
- Campus Resources
- Building Resources (WIDs)
- WID Resources and the WID Compute Cluster
- Submitting with HTCondor
Core Computational Technology

- **Mission**: to stimulate new discoveries by providing scientists with effective and dependable access to state-of-the-art research computing capabilities
- Shared between WID and MIR

- **On-Campus partners**
  - Center for High Throughput Computing (CHTC)
  - HTCondor development team

- **Off-Campus partners**
  - Open Science Grid (OSG)
CCT and Your Research . . .

Cloud?

Open Science Grid
~700 million

CHTC
~100 million

Your Desktop:
~10k hrs per year

Computational Clinic 2/22/13
CCT and Your Research

Hours Served by Resource Collection

(values listed are weekly totals)
Current CCT Services

- Consultations regarding
  - submitting computing jobs
  - accessing additional computing resources
  - other computing desires
- Outreach sessions (like this one!)
- Office Hours
  - Tuesday and Thursday, 2-3:30pm
  - CHTC help desk, 2nd floor by the stairs
CCT Staff

- **Miron Livny**
  - Principal Scientist, CCT
  - CTO of WIDs, PI of HTCondor Project
  - Facility Coordinator for OSG

- **Brooklin Gore**
  - Senior Researcher, CCT
  - manages CHTC and HTCondor Project, COO of SWAMP

- **Michelle Craft**
  - Research IT Coordinator, CCT (WID)

- **Lauren Michael**
  - Research Computing Facilitator, CCT & CHTC
The Core Computational Technology research group, also known as CCT, is shared between WID and its partner the Morgridge Institute for Research.

CCT develops a cyberinfrastructure that provides high throughput computing and fosters collaborative work with global partners.

Miron Livny, a UW-Madison professor of computer sciences in the College of Letters and Science and director of the UW Center for High Throughput Computing, provides the advanced computing tools and infrastructure necessary to facilitate the leading-edge work of scientists in both institutes. Livny, who specializes in distributed computing, will continue his work with researchers around the world to advance computer and data intensive science.

“Our center brings together a unique ensemble of scientists whose goals are to stimulate new discoveries by providing scientists with effective and dependable
Select Campus Services

- **Online**
  - WiscMail (merged with WiscCal): [https://wiscmail.wisc.edu/](https://wiscmail.wisc.edu/)
  - UW Google Docs: [http://www.doit.wisc.edu/googleapps/](http://www.doit.wisc.edu/googleapps/)
  - UW Doodle Polls: [http://www.doit.wisc.edu/calendaring/doodle/default.aspx](http://www.doit.wisc.edu/calendaring/doodle/default.aspx)

- **Software**
  - Campus Licenses: Matlab, Mathematica, Autodesk, etc.
  - Free Software: Sysmantec AntiVirus, OneNote, Sharepoint Designer, etc.

- **DoIT Tech Store**
  - Physically located in the Comp Sci Building
  - [https://techstore.doit.wisc.edu](https://techstore.doit.wisc.edu)
  - Direct, rapid purchases of computers, accessories, printers

- **DoIT Support**
  - Laptops, software support, training
DoIT Services List

Support
- Help Desk Home
- Knowledge Base
- Computer Repair
- Laptop Checkout
- Equipment Rental
- Training
- Outages

Applications
- My UW-Madison
- WiscMail | WiscMail Plus
- WiscCal
- Learn@UW
- My Workspace
- WiscChat
- WiscList
- Qualtrics (survey tool)
- Google Apps
- Web Conferencing

Get Connected
- NetID
- Guest NetID
- Mobile Access
- Remote Access (WiscVPN)
- Computer Labs
- Computer Kiosks
- Voice/Telephone

Instructional Services
- Academic Technology Home
- Learn@UW
- Free Consultations
- Course Development

Security
- Report an Incident
- IT Policies
- Campus IT Security
- Free Antivirus

Enterprise Services
- Data Centers
- Middleware Services
- Network Services
- Report Distribution
- Email Lists
- Web Content Management
- File & Data Storage
- Manifest Group Access Management

Consulting Services
- Web Accessibility
- Web Design
- Site Hosting
- Application Development
- Software and Load Testing
- Custom Training

Media Production & Services
- Publishing & Printing
- Digital Media Center
- Video Production
- Streaming Media

Access to Campus Data
- IT System Directory
- InfoAccess
- Query Library

Need Help?
- (608) 264-4357
- Email Us
- Live Chat
- Submit a Question
- Visit a Help Desk

http://www.doit.wisc.edu/services/
WIDs Building Resources

- **WIDs = Wisconsin Institutes for Discovery**
  - Contains both WID (Wisconsin Institute for Discovery) and MIR (Morgridge Institute for Research)

- **Meeting Rooms**
  - Public Rooms (via Town Center)
    - First Floor: Forum, Telepresence Room, Breakout Rooms
    - Teaching Labs on floors 2-4
    - Researchers’ Link on the second floor
  - Seminar rooms: green curtained rooms, 2 each on floors 2-4
  - Research Team Workspaces: corners of floors 2-4
  - All Rooms have supported AV Systems

- **Printers**
  - All printers are available to all building occupants
  - One color scanner/fax/printer per kitchenette
  - 44” HP Deskjet plotter for posters located in front of server room

- **Server Room (Lower Level, B1228)**
  - Limited physical access
  - Full Power, Cooling, Network

- **Building Power**
  - Red outlets on the building generator
    - ask your theme admin before connecting anything new
  - Servers on UPS (uninterruptable power supply)
WIDs IT Resources

- Active Directory Login (common username/password)
- Network
  - Wireless
    - Over 200 wireless access points
    - **WIDMIR** for building occupants
    - **WIDMIR-guest** network for guests, printer access blocked
    - **eduroam** for visitors from participating campuses, [http://www.eduroam.org/](http://www.eduroam.org/)
  - Local wired network available
  - VPN at vpn.wid.wisc.edu, separate account required
- **Intranet:** [http://intranet.discovery.wisc.edu/](http://intranet.discovery.wisc.edu/)
  - Primary source of facilities and IT information
  - Information and resources shared among building occupants
  - Only accessible from inside the building or through the VPN
- Data Sharing and Storage
  - Document Storage: Samba shares, 2TB per theme, backed up daily
  - Theme specific (NFS)
  - Working Data Storage (a.k.a. Gluster)
Working Data Storage

Any machine mounting the Working Storage

wid-submit ... wid-001 ... opt-a014

/allshared
/progs

Shared Directories for all of WIDs

/bionates
/c4
/opt
/sysbio
/widadmin

WID Theme Directories (10 TB Quota Per Theme)

/home
All users have their own home directory here:
/mnt/ws/<theme name>/home/<username>

/shared
Each group has shared project space:
/mnt/ws/<theme name>/shared/<project name>

Contact support@discovery.wisc.edu to upload large amounts of data with Globus FTP.

Available for laptops via samba. Mount points:
On Windows: \gfs.discovery.wisc.edu\ws
On Mac: smb://gfs.discovery.wisc.edu/ws

flow is recommended for use with Mac:
http://fivedetails.com/flow/
WIDs Server/Service Resources

- **Webhosting - some current sites:**
  - Drupal
    - [http://wid.discovery.wisc.edu](http://wid.discovery.wisc.edu) (used for events and surveys)
    - [http://www.evolution.wisc.edu/](http://www.evolution.wisc.edu/)
  - Wordpress
    - [https://events.discovery.wisc.edu/mip2013/](https://events.discovery.wisc.edu/mip2013/)
    - [https://blogs.discovery.wisc.edu/](https://blogs.discovery.wisc.edu/)
    - [http://wisconsinsciencefest.org/](http://wisconsinsciencefest.org/)

- **Virtual Machines**
  - as installed on WIDs servers, provides additional resources (databases, web presence, etc)
  - as installed on an individual or group machine, enables access to multiple operating systems

- **Document Storage and Revision Control**
  - svn1.discovery.wisc.edu: offers both svn and svn+ssh
  - git1.discovery.wisc.edu

- **Help Desk**
  - Please contact if any additional services are needed.
  - [support@discovery.wisc.edu](mailto:support@discovery.wisc.edu) or dial 4911
  - [http://kbox.morgridge.net](http://kbox.morgridge.net) to view/comment your current tickets
WID Resources

- **Brand New WID Website!**
  - [http://wid.wisc.edu](http://wid.wisc.edu)
  - Each theme can update their own pages

- **Publications Repository**
  - All researchers’ publications (after they joined WID) will be part of the WID Mendeley Group

- **Working Data Storage Backups**

- **WID Compute Cluster**
  - 6 Stonehaven A248, 512GB memory, 4 - 3.3GB 4-core Opteron
  - wid-001 -> wid-006.discovery.wisc.edu
  - Accessible through HTCondor
Using the WID Compute Cluster

User 1 ... User N

Interactive

Ssh to a submit machine

Job Type?

Interactive login (for command prompt/shell access):

```
condor_submit -i
```

Batch

Start an HTCondor job as usual but add the following to the submit file:

```
+group = "WID"
```

Batch/HTCondor Job Information:
- Computing with HTCondor:
  http://research.cs.wisc.edu/htcondor/
- Getting started with Condor:
  http://chtc.cs.wisc.edu/helloworld.shtml

Submit Machines:
- wid-submit.discovery.wisc.edu
- opt-submit.discovery.wisc.edu
- roy-submit.chtc.wisc.edu

WID Compute Cluster:
- wid-001... wid-006.discovery.wisc.edu
- Stonehaven A248, 512G Memory, 4 3.3G 4C Opteron

Command Returns Login Prompt

Batch Job Runs

WID Compute Cluster

Computational Clinic  2/22/13
Submitting with HTCondor
HTCondor

- Resource Management System (RMS)
- Actively developed at UW-Madison
  - research.cs.wisc.edu/htcondor/
- Formerly just “Condor”

- Used for managing the WID Pool (Opt and WID clusters)
- Allows *interactive* and *batch* job submission
How HTCondor Works

- **Central Manager** (of the pool)
- **Submit Node(s)** (where jobs are submitted)
- **Execute Node(s)** (where jobs run)
How HTCondor Works

Central Manager (of the pool)

Queue
job1.1
user1
job1.2
user1
job2.1
user2

Submit Node(s) (where jobs are submitted)

input

Job ClassAd

Machine ClassAd

Execute Node(s) (where jobs run)

output

How HTCondor Works

Central Manager (of the pool)

Queue
job1.1
user1
job1.2
user1
job2.1
user2

Submit Node(s) (where jobs are submitted)

input

Job ClassAd

Machine ClassAd

Execute Node(s) (where jobs run)

output
The WID Pool

WID Pool
Central Manager

WID Machine
(where jobs run)

Shared file system
(Gluster)
input
output

WID
WID
Opt
Machine
Opt

WID
WID

roy-
submit

Queue
job1.1
user1
job1.2
user1
job2.1
user2

wid-submit
(Job
ClassAd)

Machine
ClassAd

opt-
submit
Submitting Batch Jobs to the WID Pool

- input/output accessed automatically from the shared file system; file don’t need to be transferred between nodes
- computing “slots” allocated dynamically:
  - jobs are allocated cores, RAM, and disk space on the execute nodes based upon your submit file specifications
* Test small batches (~3 jobs) first to determine your requirements and run time.

- Contact the CHTC (Lauren) when you’re ready to submit large batches (>10 jobs).
  - chtc@cs.wisc.edu
Steps for Running Batches

1. Make jobs batch-ready
   - prepare programs (automate, compile)
   - organize input data

2. Create a submit description file

3. Submit using:
   ```bash
   condor_submit <submitfile>
   ```

4. Monitor jobs
   - `condor_q <username>` to view your jobs in queue
   - `condor_rm <jobid>` to remove jobs
Making Jobs Batch-Ready

- Jobs should:
  - run in the background
  - require no pop-up windows or mouse-clicks
  - require no interactive input

- Programs should also:
  - be compiled OR have minimal dependencies
  - utilize parameter files as much as possible

- Common programs that run well as HTCondor batches:
  - binaries, shell scripts, C/C++, Fortran
  - MatLab, python, R (with some extra measures)
Basic HTCondor Submit File

# file name is cosmos.sub
# NOTE: commands on the left are not
# case sensitive, file names
# (on the right) are, and spaces
# on left don’t matter!

universe = vanilla
executable = cosmos
input = cosmos.in
output = cosmos.out
log = cosmos.log
queue

submit filename can be anything
most jobs are vanilla
executable is a single program or shell script calling a program(s)
input are any files or programs called by your executable and programs
output is where program output goes (that is not saved to a file by the program)
log can be any file name, will be generated by HTCondor

queue value indicates how many instances of the job you want to run; default is “1”
Basic HTCondor Submit File

# file name is cosmos.sub
# NOTE: commands on the left are not
# case sensitive, file names
# (on the right) are, and spaces
# on left don’t matter!

universe     = vanilla
executable   = cosmos
input        = cosmos.in
output       = cosmos.out
log          = cosmos.log
queue
Basic WID Pool Submit File

# file name is cosmos.sub

universe = vanilla
executable = cosmos
input = cosmos.in
output = cosmos.out
log = cosmos.log
+Group = "WID"
request_cpus = 1
request_memory = 1000
request_disk = 1000000
notification = never
queue

"WID" flag grants priority over outside users
one cpu for almost all jobs
memory (RAM) in MB
disk space in KB (including output)
email notification of job completion if not specified
More Advanced: absolute references, arguments, batches

# file name is cosmos.sub

universe = vanilla
executable = /bin/cosmos
args = cosmos_$(Process).in
output = cosmos_$(Process).out
log = cosmos_$(Process).log
+Group = "WID"
request_cpus = 1
request_memory = 1000
request_disk = 1000000
notification = never
queue 10

HTCondor will look for the executable in an absolute (non-relative) location: /bin/

In this (more common) case, the executable requires an argument that would usually be typed by the user when running the program interactively, as for: $ ./cosmos cosmos_1.in

We’ve asked for 10 repeat instances of the job, each with different input/output
More Advanced: absolute references, arguments, batches

# file name is cosmos.sub

universe      = vanilla
executable    = /bin/cosmos
args          = cosmos_$(Process).in
output        = cosmos_$(Process).out
log           = cosmos_$(Process).log
+Group        = "WID"
request_cpus  = 1
request_memory = 1000
request_disk  = 1000000
notification  = never
queue        = 10

Initial File Organization

test/
cosmos_0.in
cosmos_1.in
...
...
cosmos_9.in
cosmos.sub
The Best/Most Common Way: A Full Test-Batch Example

```bash
# file name is cosmos.sub
universe = vanilla
executable = /bin/cosmos
arguments = cosmos.in
output = cosmos.out
log = cosmos.log
InitialDir = $(Process)
+Group = "WID"
request_cpus = 1
request_memory = 1000
request_disk = 1000000
notification = never
queue 3
```
Submission

```
[lmichael@opt-submit test]$ condor_submit cosmos.sub
Submitting job(s)...
3 job(s) submitted to cluster 29747.
[lmichael@opt-submit test]$
```
Viewing the Queue

```
[lmichael@opt-submit test] $ condor_q lmichael

-- Submitter: opt-submit.discovery.wisc.edu : <144.92.142.159:9620?
   sock=3678_5c57_3> : opt-submit.discovery.wisc.edu

<table>
<thead>
<tr>
<th>ID</th>
<th>OWNER</th>
<th>SUBMITTED</th>
<th>RUN_TIME</th>
<th>ST</th>
<th>PRI</th>
<th>SIZE</th>
<th>CMD</th>
</tr>
</thead>
<tbody>
<tr>
<td>29747.0</td>
<td>lmichael</td>
<td>2/15 09:06</td>
<td>0+00:00:00</td>
<td>I</td>
<td>0</td>
<td>9.8</td>
<td>cosmos</td>
</tr>
<tr>
<td>29747.1</td>
<td>lmichael</td>
<td>2/15 09:06</td>
<td>0+00:00:00</td>
<td>I</td>
<td>0</td>
<td>9.8</td>
<td>cosmos</td>
</tr>
<tr>
<td>29747.2</td>
<td>lmichael</td>
<td>2/15 09:06</td>
<td>0+00:00:00</td>
<td>I</td>
<td>0</td>
<td>9.8</td>
<td>cosmos</td>
</tr>
</tbody>
</table>

3 jobs; 0 completed, 0 removed, 3 idle, 0 running, 0 held, 0 suspended
```

- `condor_q` view all users’ jobs
### Log Files

000 (29747.001.000) 02/15 09:29:17 Job submitted from host: <144.92.142.159:9620?
  sock=3678_5c57_3>

...  

001 (29747.001.000) 02/15 09:33:59 Job executing on host: <144.92.142.153:9618?
  sock=17172_f1f3_3>

...  

006 (29747.001.000) 02/15 09:34:01 Image size of job updated: 10000

<table>
<thead>
<tr>
<th>MemoryUsage of job (MB)</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResidentSetSize of job (KB)</td>
<td>0</td>
</tr>
</tbody>
</table>

...  

005 (29747.001.000) 02/15 09:34:01 Job terminated.

(1) Normal termination (return value 0)

<table>
<thead>
<tr>
<th>Run Remote Usage</th>
<th>Usr 0:00:00:00, Sys 0:00:00:00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run Local Usage</td>
<td>Usr 0:00:00:00, Sys 0:00:00:00</td>
</tr>
<tr>
<td>Total Remote Usage</td>
<td>Usr 0:00:00:00, Sys 0:00:00:00</td>
</tr>
<tr>
<td>Total Local Usage</td>
<td>Usr 0:00:00:00, Sys 0:00:00:00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Run Bytes Sent By Job</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run Bytes Received By Job</td>
<td>0</td>
</tr>
<tr>
<td>Total Bytes Sent By Job</td>
<td>0</td>
</tr>
<tr>
<td>Total Bytes Received By Job</td>
<td>0</td>
</tr>
</tbody>
</table>

### Partitionable Resources:

<table>
<thead>
<tr>
<th>Resource</th>
<th>Usage</th>
<th>Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cpus</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Disk (KB)</td>
<td>10000</td>
<td>1000000</td>
</tr>
<tr>
<td>Memory (MB)</td>
<td>100</td>
<td>1000</td>
</tr>
</tbody>
</table>
Removing Jobs

```
[lmichael@opt-submit test] $ condor_rm 29747
Cluster 29747 has been marked for removal.
[lmichael@opt-submit test] $ condor_q lmichael

-- Submitter: opt-submit.discovery.wisc.edu : <144.92.142.159:9620?
sock=3678_5c57_3> : opt-submit.discovery.wisc.edu

<table>
<thead>
<tr>
<th>ID</th>
<th>OWNER</th>
<th>SUBMITTED</th>
<th>RUN_TIME</th>
<th>ST</th>
<th>PRI</th>
<th>SIZE</th>
<th>CMD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

0 jobs; 0 completed, 0 removed, 0 idle, 0 running, 0 held, 0 suspended
```

- `condor_rm 29747` remove a single job
- `condor_rm -a` remove ALL of your jobs (!)
Please contact us first if you . . .

- Haven’t used HTCondor before
- Have 10+ jobs in a batch
- Want to use
  - MatLab, R, Python
  - checkpointing
  - cumulative large disk or memory
- Have runtimes longer than 2 days per job
- Need more resources
  - “flocking”
  - helpful if you have >100 longer jobs (>12 hrs)
  - or even if you have >10,000 short jobs (mins to hrs)

Support: chtc@cs.wisc.edu
Website: chtc.cs.wisc.edu

Computational Clinic 2/22/13
Getting Help

- **DoIT**
  - [help@doit.wisc.edu](mailto:help@doit.wisc.edu)
  - (608) 264-HELP (4357)

- **Building IT or AV Support:**
  - [support@discovery.wisc.edu](mailto:support@discovery.wisc.edu)
  - (608) 316-4911
  (4911 from WIDs building)

- **Research IT Support**
  - [researchcomputing@discovery.wisc.edu](mailto:researchcomputing@discovery.wisc.edu)

- **CHTC Support**
  - [chtc@cs.wisc.edu](mailto:chtc@cs.wisc.edu)
  - [http://chtc.cs.wisc.edu](http://chtc.cs.wisc.edu)

- **Tea @3**
  - Weekdays at 3pm on the 3rd floor mezzanine
  - Meet your fellow researchers and support staff
  - Learn about scientific discoveries and collaborations
  - Enjoy free coffee, tea, and snacks

- If you are unsure who to contact, anyone in support can help you get what you need.