Once more, with feeling!
A monitoring feedback loop for HTC jobs with unknown requirements

Ben Tovar
University of Notre Dame

btovar@nd.edu
Where we are

Scientist says:
"This demo task runs on my laptop, but I need much more for the real application. It would be great if we can run O(25K) tasks like this on this cloud/grid/cluster I have heard so much about."
Who we are

The Cooperative Computing Lab
Computer Science and Engineering
University of Notre Dame
Cooperative Computing Lab

Director

Staff

Director: Douglas Thain

Research Software Engr: Benjamin Tovar

Graduate Students

Patrick Donnelly

Peter Ivie

Haiyan Meng
First Responder

Nicholas Hazekamp
Outreach Coordinator

Nathaniel Kremer-Herman

Not shown, grad students: Tim Shaffer, Chao Zheng
CCL Objectives

- Harness all the resources that are available: desktops, clusters, clouds, and grids.
- Make it easy to scale up from one desktop to national scale infrastructure.
- Provide familiar interfaces that make it easy to connect existing apps together.
- Allow portability across operating systems, storage systems, middleware...
- Make simple things easy, and complex things possible.
- No special privileges required.
CCTools

• Open source, GNU General Public License.
• Compiles in 1-2 minutes, installs in $HOME.
• Runs on Linux, Solaris, MacOS, Cygwin, FreeBSD, ...
• Interoperates with many distributed computing systems.
  – Condor, SGE, Torque, Globus, iRODS, Hadoop...
• Components:
  – Makeflow – A portable workflow manager.
  – Work Queue – A lightweight distributed execution system.
  – All-Pairs / Wavefront / SAND – Specialized execution engines.
  – Parrot – A personal user-level virtual file system.
  – Chirp – A user-level distributed filesystem.
Long-tail of science

Individual researchers and small laboratories that:

Need to curate, manage, and analyse large amounts of data.

May not know how to access computational resources available to them.

May not have immediate access to the required resources.

(i.e., they know their discipline, but they do not have an HTC expert in their team)
Notre Dame's happy opportunistic situation

- ~25k cores at Notre Dame's Center for Research Computing (CRC)

- They belong to different individual PIs, but they are available through condor when not used by their owners.
## Notre Dame Condor Status

<table>
<thead>
<tr>
<th>Account</th>
<th>Slots</th>
<th>Cores</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:gcamargo@nd.edu">gcamargo@nd.edu</a></td>
<td>4428</td>
<td>4428</td>
</tr>
<tr>
<td><a href="mailto:mthomann@nd.edu">mthomann@nd.edu</a></td>
<td>635</td>
<td>635</td>
</tr>
<tr>
<td><a href="mailto:kwilli20@nd.edu">kwilli20@nd.edu</a></td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td><a href="mailto:nkirkpat@nd.edu">nkirkpat@nd.edu</a></td>
<td>343</td>
<td>343</td>
</tr>
<tr>
<td><a href="mailto:xli19@nd.edu">xli19@nd.edu</a></td>
<td>223</td>
<td>223</td>
</tr>
<tr>
<td><a href="mailto:jkinniso@nd.edu">jkinniso@nd.edu</a></td>
<td>14</td>
<td>112</td>
</tr>
<tr>
<td><a href="mailto:eoabadtc@nd.edu">eoabadtc@nd.edu</a></td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td><a href="mailto:tperkin1@nd.edu">tperkin1@nd.edu</a></td>
<td>73</td>
<td>73</td>
</tr>
<tr>
<td><a href="mailto:marangu1@nd.edu">marangu1@nd.edu</a></td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td><a href="mailto:rsmick@nd.edu">rsmick@nd.edu</a></td>
<td>28</td>
<td>28</td>
</tr>
<tr>
<td><a href="mailto:ochoudhu@nd.edu">ochoudhu@nd.edu</a></td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td><a href="mailto:kherring@nd.edu">kherring@nd.edu</a></td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td><a href="mailto:mvalenc2@nd.edu">mvalenc2@nd.edu</a></td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td><a href="mailto:ekrebs@nd.edu">ekrebs@nd.edu</a></td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td><a href="mailto:jdizor@nd.edu">jdizor@nd.edu</a></td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td><a href="mailto:nblanca@nd.edu">nblanca@nd.edu</a></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><a href="mailto:nsmith9@nd.edu">nsmith9@nd.edu</a></td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Display Options**
- Sort: [users, machines]
- Show: [users, states]
- Size: [bigger, smaller]
- Scale: [none]
Dialogue with our target users
Dialogue with our target users

- How much memory does it use?
- Eh....
Dialogue with our target users

- How much memory does it use?
  - Eh....

- What about disk usage?
  - Well...
Dialogue with our target users

- How much memory does it use?
  - Eh...

- What about disk usage?
  - Well...

- A ballpark figure?
  - Mmm... It runs in my laptop...
Dialogue with our target users

- How much memory does it use?
- Eh....

- What about disk usage?
- Well...

- A ballpark figure?
- Mmm... It runs in my laptop...

- Surely you have a list of all the files used?
- ...
Dialogue with our target users

- How much memory does it use?
  - Eh....

- What about disk usage?
  - Well...

- A ballpark figure?
  - Mmm... It runs in my laptop...

- Surely you have a list of all the files used?
  - ...
Dialog with our target users

- Ok, I think we got the condor info right...  
SuperSequencer3000 seems to be working on the remotes nodes now.
- Yaaaay! I'll run our workflow shortly!
EARLY THE NEXT MORNING...
Dialog with our target users

- It does not work anymore? Did you change anything?
- No!
Dialog with our target users

- It does not work anymore? Did you change anything?
  - No!

- Your jobs are running out of disk. Nothing changed?
  - No!
Dialog with our target users

- It does not work anymore? Did you change anything?
  - No!

- Your jobs are running out of disk. Nothing changed?
  - No!

- Wait, that parameter looks different from last time.
Dialog with our target users

- It does not work anymore? Did you change anything?
  - No!

- Your jobs are running out of disk. Nothing changed?
  - No!

- Wait, that parameter looks different from last time.
  - Oh, that! Yes, we did change that...
Dialog with our target users

- It does not work anymore? Did you change anything?
  - No!

- Your jobs are running out of disk. Nothing changed?
  - No!

- Wait, that parameter looks different from last time.
  - Oh, that! Yes, we did change that...
Dialog with our target users

- It does not work anymore? Did you change anything?
  - No!

- Your jobs are running out of disk. Nothing changed?
  - No!

- Wait, that parameter looks different from last time.
  - Oh, that! Yes, we did change that...
    ...but we need to change that parameter often for our research...
Where we want to be

Run my scientific workflow!

Execution System (e.g. Condor)

Estimation

resources provision

resources historical data

Archive

resources measurements
Tasks with Unknown Resource Requirements

Tasks which size (e.g., cores, memory, and disk) is not known until runtime.

Available condor slots
Tasks with Unknown Resource Requirements

Tasks which size (e.g., cores, memory, and disk) is not known until runtime.

One task per slot: Wasted resources, reduced throughput.

Many tasks per slot (e.g. with pilot job): Resource contention/exhaustion, reduce throughput.
Task-in-the-Box

Condor slots
Task-in-the-Box

Allocations inside a slot

Condor slots
Task-in-the-Box

One task per allocation

One task per allocation

Condor slots
Task-in-the-Box

One task per allocation

Task exhausted its allocation

Condor slots
Task-in-the-Box

One task per allocation

Retry allocating a whole slot

Condor slots
Main Challenges

What is a good allocation size?

How do we measure the tasks?

How do we enforce the allocations?
One-guess policy result (guess once, then use max seen)

Real result from a production High-Energy Physics CMS analysis (Lobster NDCMS)

Histogram Peak Memory vs Number of Tasks
O(700K) tasks that ran in O(26K) cores managed by WorkQueue/Condor.

First-allocation that maximizes expected throughput (increase of %40 w.r.t. no task is retried)
And around it goes...

- Query/set historical information
- Informed guess on task size
- Monitor task. Kill if resources are exhausted
- Task successful?
  - Yes: Record success/failure
  - No: execution loop
    - hard limit hit?
      - Yes: main loop
      - No: execution loop
And around it goes...

What do we know?
- Historical data?
- Probability distribution?
- Perfect information?
- **Empirical distribution?**

Flowchart:
- Query/set historical information
- Informed guess on task size
- Monitor task. Kill if resources are exhausted
- task successful?
  - No
  - hard limit hit?
    - No
    - Yes: Record success/failure
    - Yes: main loop
  - Yes: main loop
And around it goes...

What do we know?
- Historical data?
- Probability distribution?
- Perfect information?
- Empirical distribution?

What do we want?
- Minimize retries?
- Minimize waste?
- Maximize throughput?
Slow-peaks model

Random variables to describe usage:
- Time to completion.
- Size of max peak

Resource usage:
- time x peak

Slow-peaks:
- Resource peaks at the end of execution (conservative assumption)
A LOT OF BORING MATH LATER...
Slow-peaks model

\[
E[\text{waste}(r, \tau, a_1)] = \int_0^\infty \left( \int_0^{a_1} (a_1 - r) \tau p(r, \tau) \, dr \right) \, d\tau \\
+ \int_{a_1}^{a_m} \left( (a_m + a_1 - r) \tau p(r, \tau) \, dr \right) \, d\tau \\
= a_1 \int_{a_1}^{a_m} \int_0^\infty \tau p(r, \tau) \, d\tau \, dr \\
+ a_m \int_{a_1}^{a_m} \int_0^\infty \tau p(\tau | r) \, d\tau \, p(r) \, dr \\
- \int_0^\infty \int_0^\infty r \tau p(r, \tau) \, d\tau \, dr,
\]

Choice of: maximum throughput minimum waste.

Optimizations over expectations

O(n) simple arithmetic expressions that use only information available during execution.
**Integrated in CCTools** (next major release)

```
makeflow --max-throughput -Tcondor myworkflow
```

Activate monitor and allocations

Submit jobs to condor. Allocations in terms of request_cpus, request_memory and request_disk.

```bash
# unix make style recipes

output.0: input.0 cmd
   ./cmd -i input.0 output.0

output.1: input.1 othercmd
   ./othercmd < input.1 > output.1
```
We need monitoring for all of this

Mechanisms available to unprivileged users
root permissions or loading kernel modules are a no go

Tasks as trees of processes
no whole systems or individual processes

High-throughput computing
measure so we can run many tasks at the same time,
not to profile a single instance to make it run faster
We need monitoring for all of this

Monitoring as an unprivileged user is hard!

- No permissions
- No ways to add needed kernel support
- What the user wants to measure is different to what a system administrator may care about. (E.g., cpu usage of a single task v.s. system load.)
- Tracking children processes is hard without wrapping the parent process.

Need to measure individual tasks, not individual users or systems.
Integrated in CCTools

resource_monitor -L"cores: 4" -L"memory: 4096" -- matlab

(c) 2016
(c) 2016

(resource_monitor -L"cores: 4" -L"memory: 4096" -- matlab)
Recent development: Monitoring Library

The resource_monitor main functionality was converted into a library, with C, Python, and perl interfaces. An application can poll its resources usage with a single library call. (unlike resource_monitor, does not track forks/exits)
ND CMS workflow distributions

- 681874 tasks on Lobster/WorkQueue/Condor
- Computing allocations takes \( \sim 0.05 \) seconds.
ND CMS workflow: Memory bottleneck

<table>
<thead>
<tr>
<th></th>
<th>Size</th>
<th>Retries</th>
<th>Proportion wasted</th>
<th>Throughput (norm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max peak always</td>
<td>3GB</td>
<td>0%</td>
<td>48%</td>
<td>1.0</td>
</tr>
<tr>
<td>Perfect information</td>
<td>-</td>
<td>0%</td>
<td>0%</td>
<td>2.0</td>
</tr>
<tr>
<td>Slow-peaks one-guess</td>
<td>1.9GB</td>
<td>9%</td>
<td>28%</td>
<td>1.41</td>
</tr>
</tbody>
</table>
### ND CMS workflow: Memory bottleneck

Things are even better if users give coarse information about the workflow. As simple as putting tasks into categories (e.g., merge, analysis recostep, parameter-X, etc.)

<table>
<thead>
<tr>
<th>Details</th>
<th>Size</th>
<th>Retries</th>
<th>Proportion wasted</th>
<th>Throughput (norm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max peak always</td>
<td>3GB</td>
<td>0%</td>
<td>48%</td>
<td>1.0</td>
</tr>
<tr>
<td>Perfect information</td>
<td>-</td>
<td>0%</td>
<td>0%</td>
<td>2.0</td>
</tr>
<tr>
<td>Slow-peaks one-guess</td>
<td>1.9GB</td>
<td>9%</td>
<td>28%</td>
<td>1.41</td>
</tr>
<tr>
<td>One-guess + categories (per category)</td>
<td>&lt; 1%</td>
<td>17%</td>
<td>1.64</td>
<td></td>
</tr>
</tbody>
</table>
Questions?

Acknowledgements:

Many thanks to ND CMS group:

Prof. Kevin Lannon
Anna Woodard
Mathias Wolf
Kenyi Hurtado

DOE Grant: ER26110

(Paper under current blind-review.
If you are a reviewer,
you are feeling very sleepy...
At the count of three
you will forget all of this...)