Introduction

1. Description of the cluster
2. Preparing and running a job
3. Using the cluster effectively
4. Demo(s)
1. The Cluster
What is a Compute Cluster?

a group of computers ('nodes') connected by a dedicated high-speed network

- high performance
- high availability
How is it used?

serial computing

parallel computing
How is it used?

serial computing

parallel computing
How is it used?

- Log in
- (Prepare code)
- Prepare job for a *workload scheduler*
  - at PIK: LoadLeveler
- Submit
The PIK Compute Cluster
- hardware

**Interactive Nodes**
- `login01`
- `login02`

**Compute Nodes**
- `dx001`
- `dx002`
- `dx079`
- `dx080`
- `dx081`
- `dx082`
- `dx159`
- `dx160`
- `dx161`
- `dx162`
- `dx239`
- `dx240`
- `dx241`
- `dx242`
- `dx319`
- `dx320`

**Disks**
- `disks`

**SSH login**
- `your PC`
The PIK Compute Cluster
- hardware

<table>
<thead>
<tr>
<th>CPU</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>32GB</td>
</tr>
</tbody>
</table>

2x quad-core Intel Xeon E5472 @ 3.00GHz

dxnnn
The PIK Compute Cluster
- filesystems

- /home/(your username)
  - not for application data!

- /iplex/01
  - fast, parallel filesystem for project groups, backed up

- /scratch/01
  - fast, parallel filesystem. No backups, 30+ day old files deleted

- archiving and backups: IBM Tivoli Storage Manager (dsmc, dsmj)
The PIK Compute Cluster
- software

Development Tools
- Intel Cluster Studio (C, C++, Fortran, MPI, MKL)
- Matlab, Python, R
- source code management: Subversion
- standard Linux tools: GCC, sed, awk, make
The PIK Compute Cluster
- software

Code profiling and tuning
- Intel VTune Amplifier XE2011
- Intel Trace Analyzer and Collector
- GNU profiler
The PIK Compute Cluster
- software

Debugging
● Totalview debugger
● Intel debugger
● GNU debugger

Most software: /iplex/01/sys/applications
The PIK Compute Cluster

- so far this year:
  - 200000 jobs run
  - 9 million hours of CPU time
9M hours = 1026 years

985AD: Erik the Red colonises Greenland
2. Preparing & running a job

Running a program
- log in

● Login via Secure Shell (SSH) public key authentication
  ○ see HPC documentation

● From Linux or a Mac
  ○ ssh cluster on the command line / Terminal
  ○ ssh cluster.pik-potsdam.de

● From Windows
  ○ PuTTY ([http://putty.org](http://putty.org))

Running a program
- log in

LAST DOWNTIME
Cluster was down due to scheduled maintenance
August 14th 6:00 p.m. to August 15th 6:00 a.m.

UPCOMING MAINTENANCE
The cluster will be out of service from January 2nd to January 5th 2012
due to the annual maintenance of computer room facilities.
Running a program
- preparing a job

The LoadLeveler Job Command File (JCF)

#!/bin/ksh

# @ class = short
# @ group = its
# @ notify_user = linstead@pik-potsdam.de
# @ output = example.out
# @ error = example.err
# @ queue

/home/lininstead/examples/serial/hello_world
Running a program
- submitting a job

- llsubmit example.jcf

llsubmit: The job "cwsa.implex.pik-potsdam.de.102812" has been submitted.

- llq [ -u myusername ]

cwsa.102812.0 lininstead 11/7 12:10 I 50 short
Running a program
- submitting a job

---

From: loadl@pik-potsdam.de
To: linstead@pik-potsdam.de
Subject: cwsa.ipelex.pik-potsdam.de.102812
Date: Mon, 07 Nov 2011 12:11:28 +0100

---

From: LoadLeveler

LoadLeveler Job Step: cwsa.ipelex.pik-potsdam.de.102812.0
Executable: /home/linstead/examples/JCF/simple.jcf
Executable arguments:
State for machine: dx055.ipelex.pik-potsdam.de
LoadL_starter: The program, simple.jcf, exited normally and returned an exit code of 0.

This job step was dispatched to run 1 time(s).
This job step was rejected by Starter 0 time(s).
Submitted at: Mon Nov  7 12:10:30 2011
Started at: Mon Nov  7 12:11:28 2011
Exited at: Mon Nov  7 12:11:28 2011
Real Time:  00:00:58
Job Step User Time: 00:00:00
Job Step System Time: 00:00:00
Total Job Step Time: 00:00:00
Starter User Time: 00:00:00
Starter System Time: 00:00:00
Total Starter Time: 00:00:00
Running a program
- but where's my output?

Unless I say otherwise, same place as my JCF

linstead@login01:~/examples/helloworld$ cat example.out
Hello world!
Running a program
- more examples

/iplex/01/sys/loadl/examples
3. Using the cluster effectively

Using the cluster effectively
- filesystems

$HOME not for application data!
Using the cluster effectively
- useful JCF parameters

# @ class = ...

- short
  - 1 day, 256 processes

- medium
  - 7 days, 128 processes

- long
  - 30 days, 4 processes

- largemem
  - 7 days, 1 core, 14GB memory

- dev, io
Using the cluster effectively
- useful JCF parameters

```
# @ wall_clock_limit = ...

- HH:MM:SS
- lets you jump the queue!
```
Using the cluster effectively
- useful JCF parameters

```bash
# @ job_name = ...
```

- easily identify your job in log files
- Recommended: use model name
  - e.g. LPJmL, Lagom, PISM-PIK etc
Using the cluster effectively
- useful JCF parameters

# @ group = ...

- run llgroups
- pick the right one for your job
Using the cluster effectively
- useful JCF parameters

● OK

    # @ output = example.out
    # @ error = example.err
Using the cluster effectively
- useful JCF parameters

● Better...

# @ output = /scratch/01/linstead/example.out
Using the cluster effectively
- useful JCF parameters

● even better...

```bash
# @ output = /scratch/01/linstead/example_${jobid}.out
```
Using the cluster effectively
- useful JCF parameters

● Better still...

```bash
# @ output = /scratch/01/$(user)/example_$jobid.out
```
Using the cluster effectively
- useful JCF parameters

● Best!

# @ job_name = example
# @ initialdir = /scratch/01/$(user)
# @ output = $(job_name)_$(jobid)_$(stepid).out
Using the cluster effectively
- useful JCF parameters

● other variables

$(host)$: The hostname of the machine from which you submitted the job.
$(domain)$: The domain of the host from which you submitted the job.
$(jobid)$: A sequential number assigned to the job by the submitting machine. Equivalent to the $(cluster)$ variable.
$(stepid)$: The sequential number assigned to a job step when more than one queue statement appears in the Job Command file. The $(stepid)$ and $(process)$ variables are equivalent.
$(executable)$: Contains the name of the executable if you set the executable keyword.
$(base_executable)$: Contains the name of the executable without the directory path if you set the executable keyword.
$(class)$: Contains the name of the job class that your job has been submitted to if you set the class keyword.
$(comment)$: Contains the comment text if you set the comment keyword.
$(job_name)$: Contains the job name text if you set the job_name keyword.
$(step_name)$: Contains the step name text if you set the step_name keyword.
Using the cluster effectively
- useful LoadLeveler commands

● `llcancel`
  ○ `llcancel cws02a.76469.0 -h cws02a`
  ○ `llcancel -u linstead -h cwsa -h cws02a`

● `llclass`
  ○ show available classes

● `llstatus`
  ○ show the load on the cluster
Using the cluster effectively
- common errors

Job stays Idle

# @ output = /scartch/01/$(user)/$(job_name)_$$(cluster). out
Using the cluster effectively
- common errors

Wrong or missing group

llsubmit: 2512-080 Class "short" is not valid for group "itss".
llsubmit: 2512-051 This job has not been submitted to LoadLeveler.

llsubmit: 2512-078 Group "esm" is not valid for user "linstead".
llsubmit: 2512-051 This job has not been submitted to LoadLeveler.
Using the cluster effectively
- how to get help

● documentation
  ○ http://www.pik-potsdam.de/services/it/hpc
  ○ /iplex/01/sys/loadl/examples

● email
  ○ cluster-admin@pik-potsdam.de

● submitting problem reports
  ○ include job ID, path to JCF
To conclude...

Try the examples.

If you get stuck:
- cluster-admin@pik-potsdam.de

Spring 2012:
- Advanced LoadLeveler - get your job to the top of the queue.
4. Demos