LLSC Adoption of Slurm for Managing Diverse Resources and Workloads

Chansup Byun, Jeremy Kepner, William Arcand, David Bestor, Bill Bergeron, Vijay Gadepally, Michael Houle, Matthew Hubbell, Michael Jones, Anna Klein, Peter Michaleas, Lauren Milechin, Julie Mullen, Andrew Prout, Antonio Rosa, Siddharth Samsi, Charles Yee, Albert Reuther

MIT Lincoln Laboratory

SLURM User Group Meeting, September 24-26, 2017

This material is based upon work supported by the Assistant Secretary of Defense for Research and Engineering under Air Force Contract No. FA8721-05-C-0002 and/or FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Assistant Secretary of Defense for Research and Engineering.

Outline

- Introduction
- LLSC Environment
- Slurm Migration
  - LLSC software stack
  - Slurm unique features
    - Lua job_submit script
    - SPANK plug-in module
- Summary

Who We Are – a Little History


MIT Building 20

Mission: ***Development of radar systems and technology***

Main projects: Surveillance radar
Fire control radar
Navigation systems

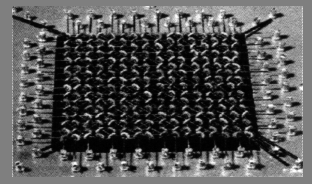
4000 employees
Designed half of all US WWII radars

Est. 1951: ***Air defense and technology development***

Main projects: Semi-Automatic Ground Environment (SAGE)

Major Innovations:

Real-Time Computing


Magnetic-core Memory


Light-pen CRT Interface

History of Supercomputing at Lincoln Laboratory

1951 Whirlwind

1963 Sketchpad

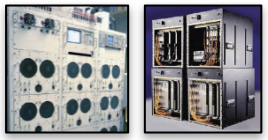
1956 TX-0

1970 Fast Digital Processor (FDP)

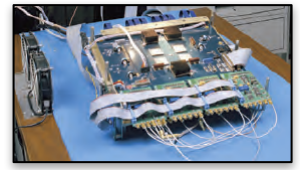
Late 1970s High-speed FFT pipelined processor

1977 : Lincoln Digital Signal Processor (LDSP)

1982 Compact LPC Vocoder



Early 1990s
- APT processor
- RAPTOR processor
- Space-Time Adaptive Processing Library (STAPL)



2002
- ISR Processing and Array Technology (IPAT) processor
- MatlabMPI

2004 Knowledge-Aided Sensor Signal Processing and expert Reasoning (KASSPER) processor

2016 Lincoln Laboratory Supercomputing Center (LLSC)

1950 1960 1970 1980 1990 2000 2010

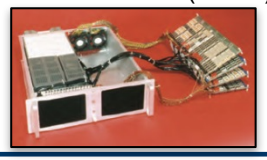
1953 Magnetic-Core Memory Array

1958
- AN/FSQ-7 (Whirlwind II)
- Average Response Computer (ARC)
- CG-24
- TX-2

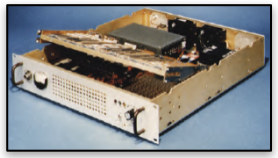
1962 Lincoln Instrument Computer (LINC)

1970 GENESYS

1974 Lincoln Digital Voice Terminal (LDVT)



1978 Micro-Processor Based LPC Vocoder (LPCM)



1992 Radar Surveillance Technology Experimental Radar (RSTER) processor

1999 Parallel Vector Library (PVL)

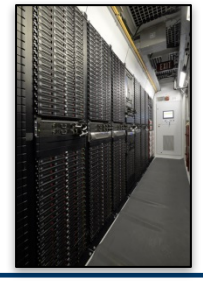
2003 pMatlab

2004
- pMapper gridMatlab
- LLGrid TX-2500

2007
- Parallel Vector Tile Optimizing Library (PVTOL)
- Real-Time Communication Layer (RTCL)

2012 D4M

2014 LLGrid TX-Green



2015 BigDAWG

LLSC Approach

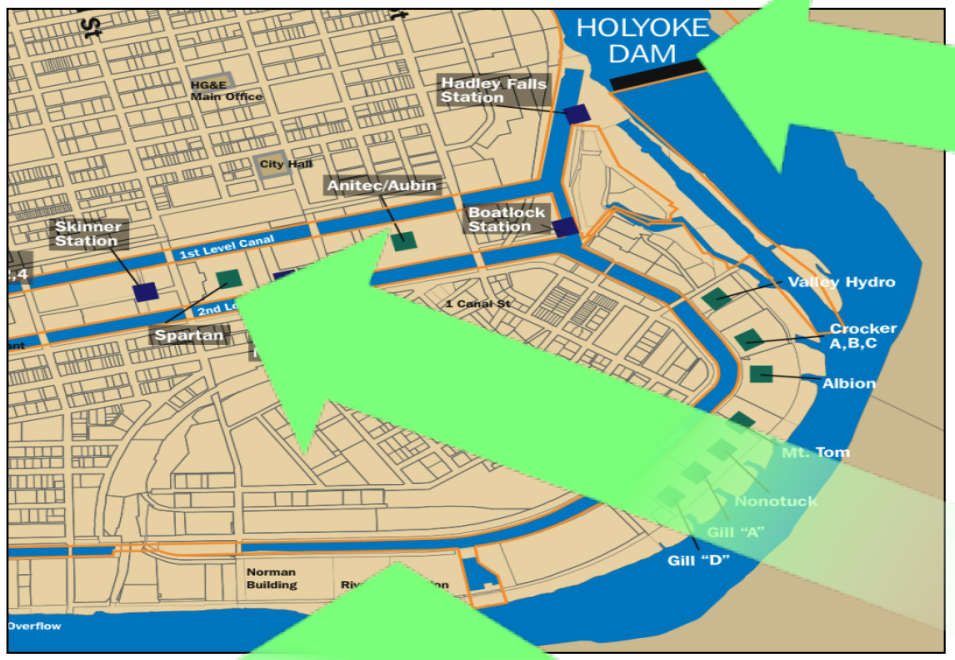
Approach

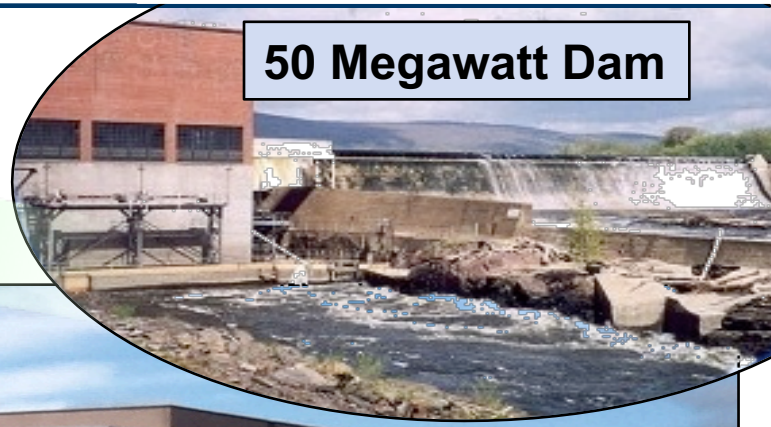
- LLSC develops and deploys unique, energy-efficient high performance computing that provides
  - Integrated HPC and Big Data capabilities
  - Data centers, hardware, software, and user support
  - 100X more productivity than standard HPC
  - 100X better performance than standard Cloud providers

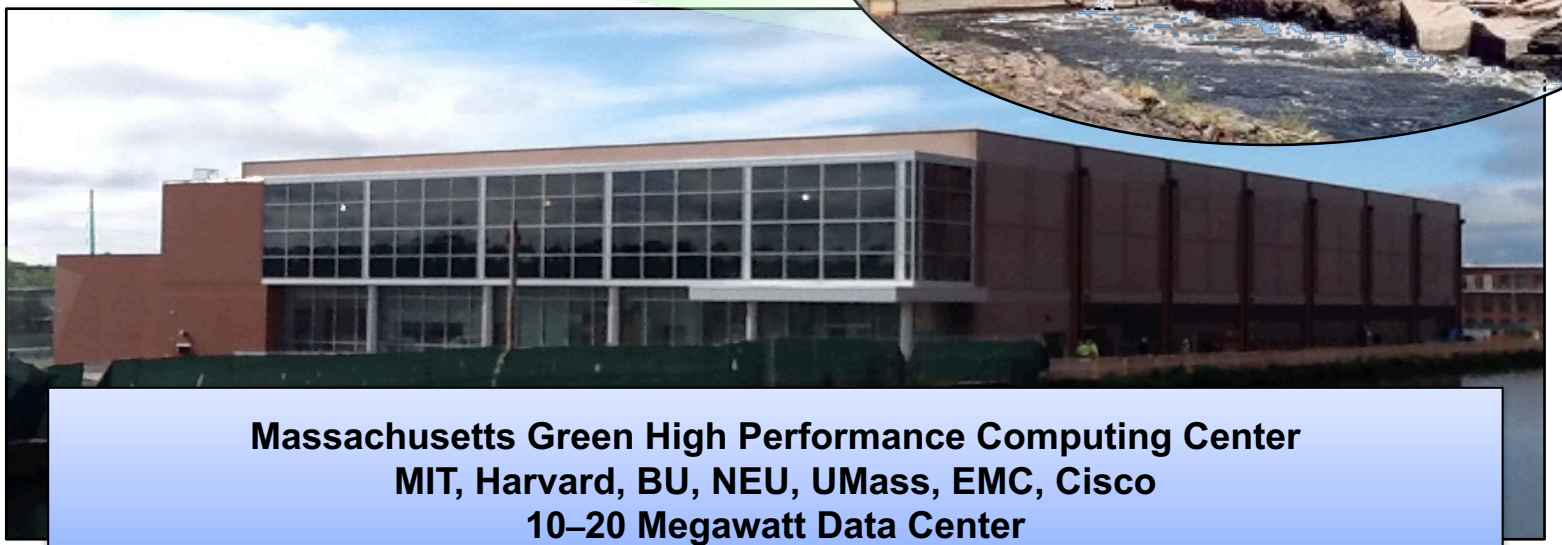
Carbon-free power

Diverse Locations

Hydro-Powered HPC in Holyoke, MA







Massachusetts Green High Performance Computing Center
MIT, Harvard, BU, NEU, UMass, EMC, Cisco
10–20 Megawatt Data Center

Lincoln Container Site
2.5 Megawatts

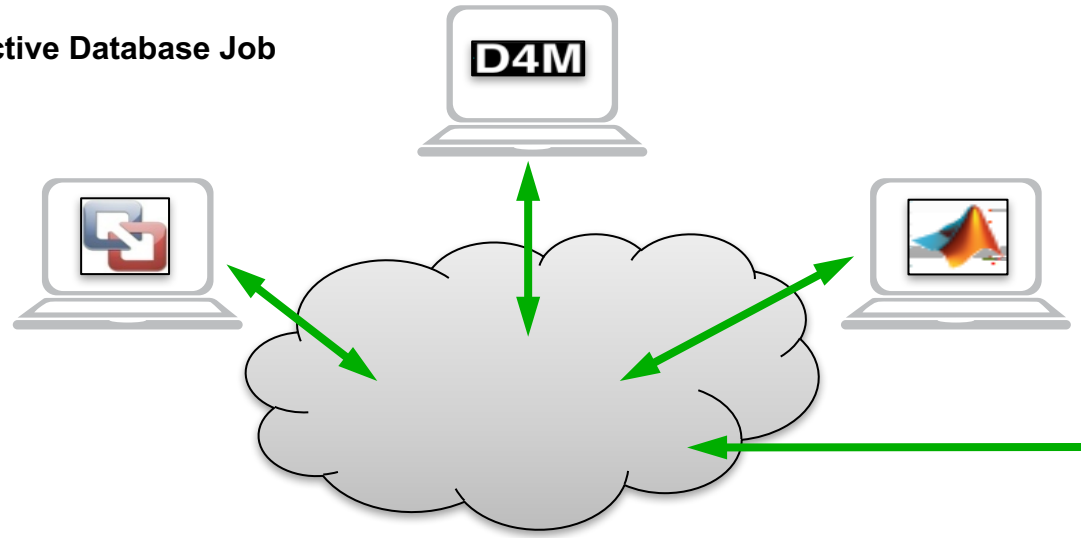
- **Energy efficient HPC is required for research at many institutions**
- **Strong statewide collaboration to put world class HPC in Holyoke, MA**

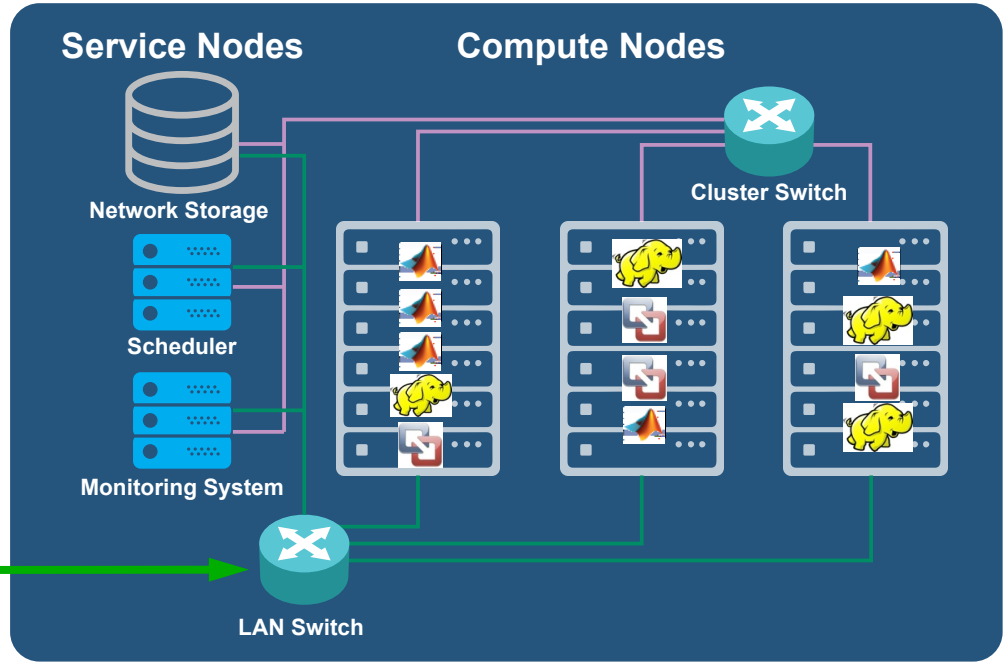
LLSC Advantage: Interactive Supercomputing

Interactive Compute Job

Interactive VM Job

Interactive Database Job





- LLSC provides a software platform that allows users to
  - Launch interactive compute jobs from their desktop
  - Share large volumes of project data
- The LLSC experience provides
  - Reference datasets pre-positioned in databases
  - Software modules and training to reduce user ramp up up time

LLSC Experience with Schedulers

- LLSC has been using various schedulers from the beginning of its grid computing efforts
  - OpenPBS and Condor were used in the past
  - In January 2012, LSF was switched to Grid Engine
    - Costs for licensing LSF become significant
    - Open-source GridEngine provides what LLSC used with LSF.
    - In-house GridEngine expert available
- Open-source GridEngine development is dormant
  - Supporting new hardware such as Intel KNL systems becomes problematic.
- LLSC has been adding more computing power and storage capacity
  - Need a new resource manager to support new hardware and software
- Slurm is selected in 2016 after re-evaluating the currently available open-source scheduler projects

Outline

- Introduction
- LLSC Environment
- Slurm Migration
  - LLSC software stack
  - Slurm unique features
    - Lua job_submit script
    - SPANK plug-in module
- Summary

Unique LLSC Interactive Supercomputing Capabilities

- **Parallel Matlab: world's most productive parallel computing environment**

```
>> eval(pRUN('myCode',256,'grid'))
```

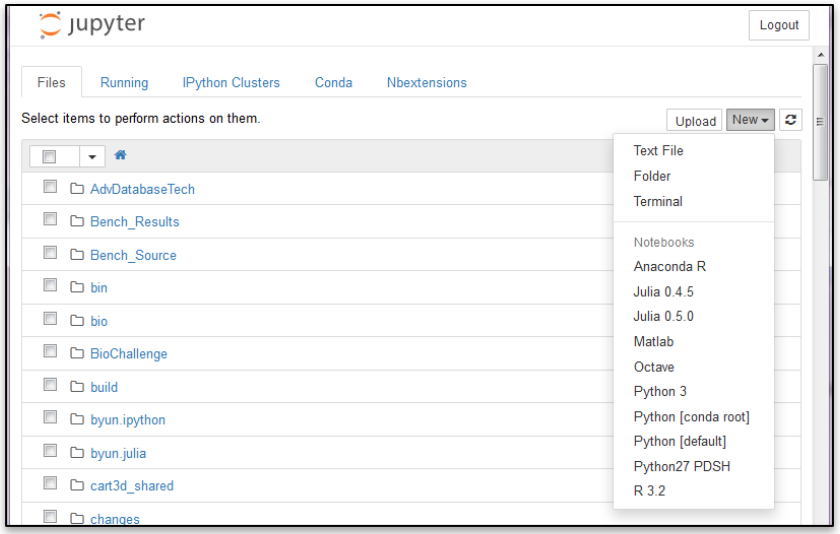
- **LLMapReduce: parallel data analysis in any language with 1 line of code**

```
>> LLMapReduce --mapper myCode --input myInDir --output myOutDir --np 256
```

- **Interactive Hardware: get a processor core or a whole node**

```
>> LLsub -i                    >> LLsub -i full
```

- **Jupyter Notebook: web-based IDE & more**
  - **Julia, Python, R, Matlab, and Octave**



- **Dynamic Web Services: start an authenticated web-service**

```
>> LLWebSvcStartHTTPD --group myGroup myGroupWebService
```

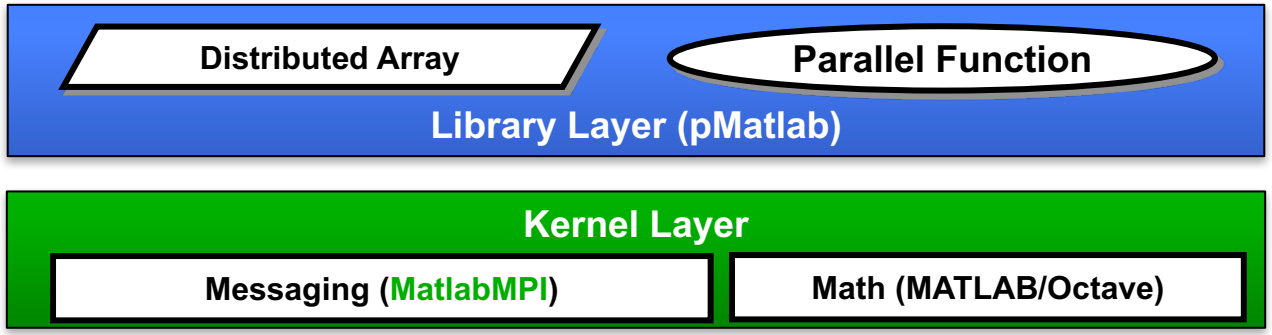
- **Dynamic Databases: manage world's most powerful databases from a GUI**

MatlabMPI & pMatlab
Bridge the Parallel Programming Gap

User

User Interface

Parallel Middleware Library provides high productivity, supporting short time horizon projects

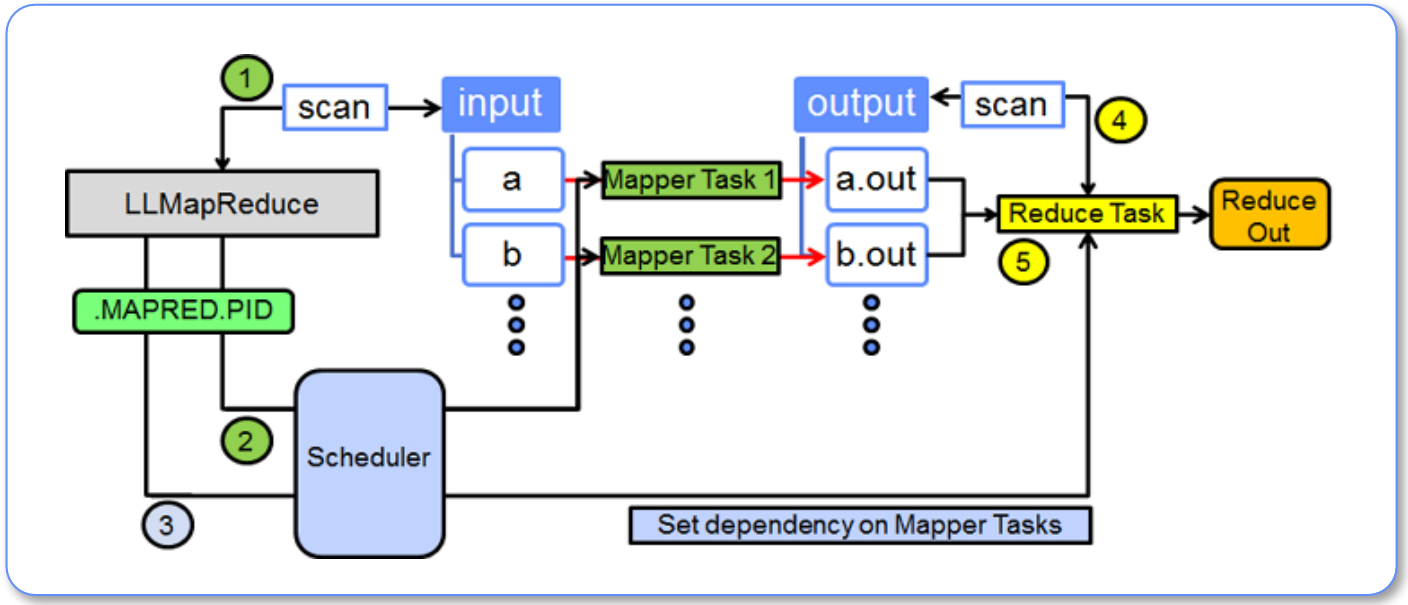


Hardware Interface

Parallel Hardware

LLMapReduce

- Create environment similar to Hadoop Map Reduce framework with a central storage filesystem
- Minimize effort required to deploy users' applications
- Support any programming language
- Prevent resource contention between jobs
- Provide options for performance optimization



Apache Mesos Cluster As a Job

- Slurm provides the compute resource where Apache Mesos cluster is dynamically constructed

```
sbatch --array=1-N --exclusive --gres=sngljob \
       /path/to/mesos-job.sh
```

  where N is the number of compute nodes
- Mesos job (mesos-job.sh) script does
  - Install and configure Apache HDFS (v. 2.7.3), ZooKeeper (v. 3.4.6), and Mesos (v. 0.25.0)
  - Start all necessary daemons
  - Deploy and configure Spark (v. 2.0.1) on Mesos
  - Wait for a signal to dismantle the Mesos cluster
- Job termination

```
scancel -b -s TERM jobid
```

Web Portal Service

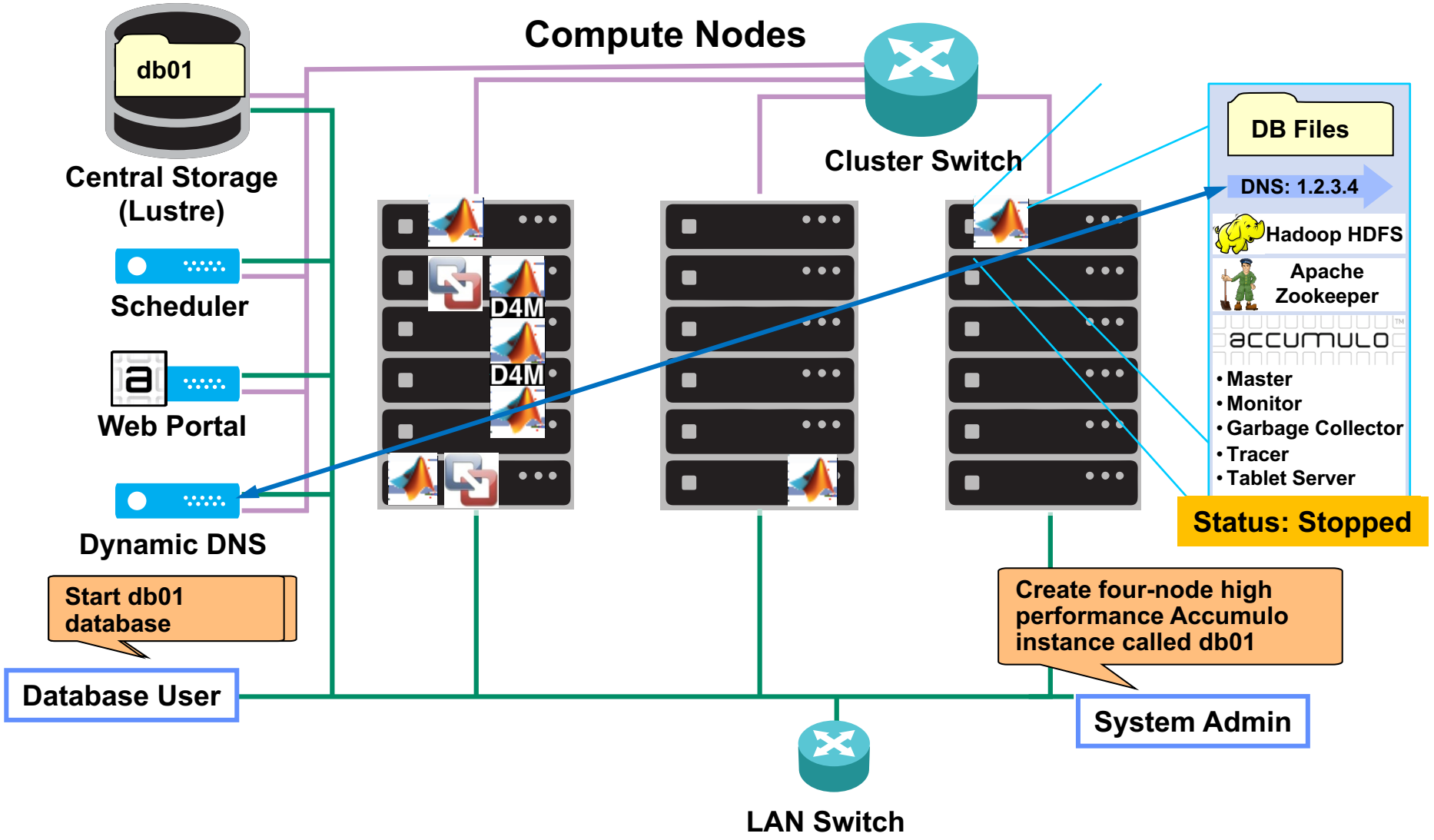
Dynamic Database: Start, Stop and Create Checkpoint

- **Start, Stop and Create Checkpoint can be executed by any member of the security group**
- **Can be requested either from the command line or web portal**
  - **Checkpoint can only be taken while the database is stopped**
- **Current status is reported in a text file on shared storage**

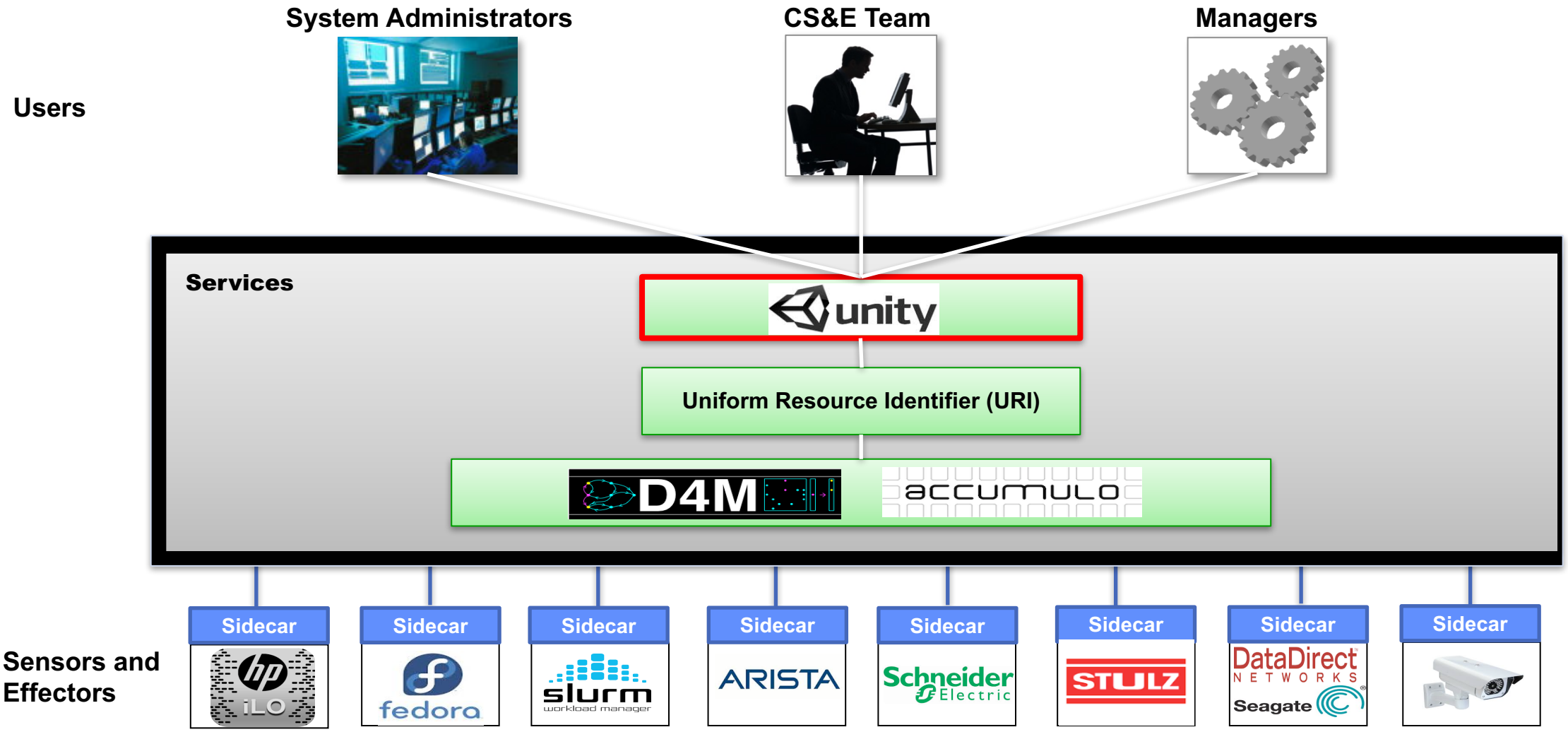
Refresh Status

Folder Name	Type	Status	Actions		
classdb01	Accumulo v1.4.1	starting	View Info		
classdb02	Accumulo v1.5.0	started	View Info	Stop	
classdb03	Accumulo v1.6.0	stopped	View Info	Start	Checkpoint
scidb01	SciDB 14.3	stopped		Start	Checkpoint
scidb02	SciDB 14.3	started		Stop	

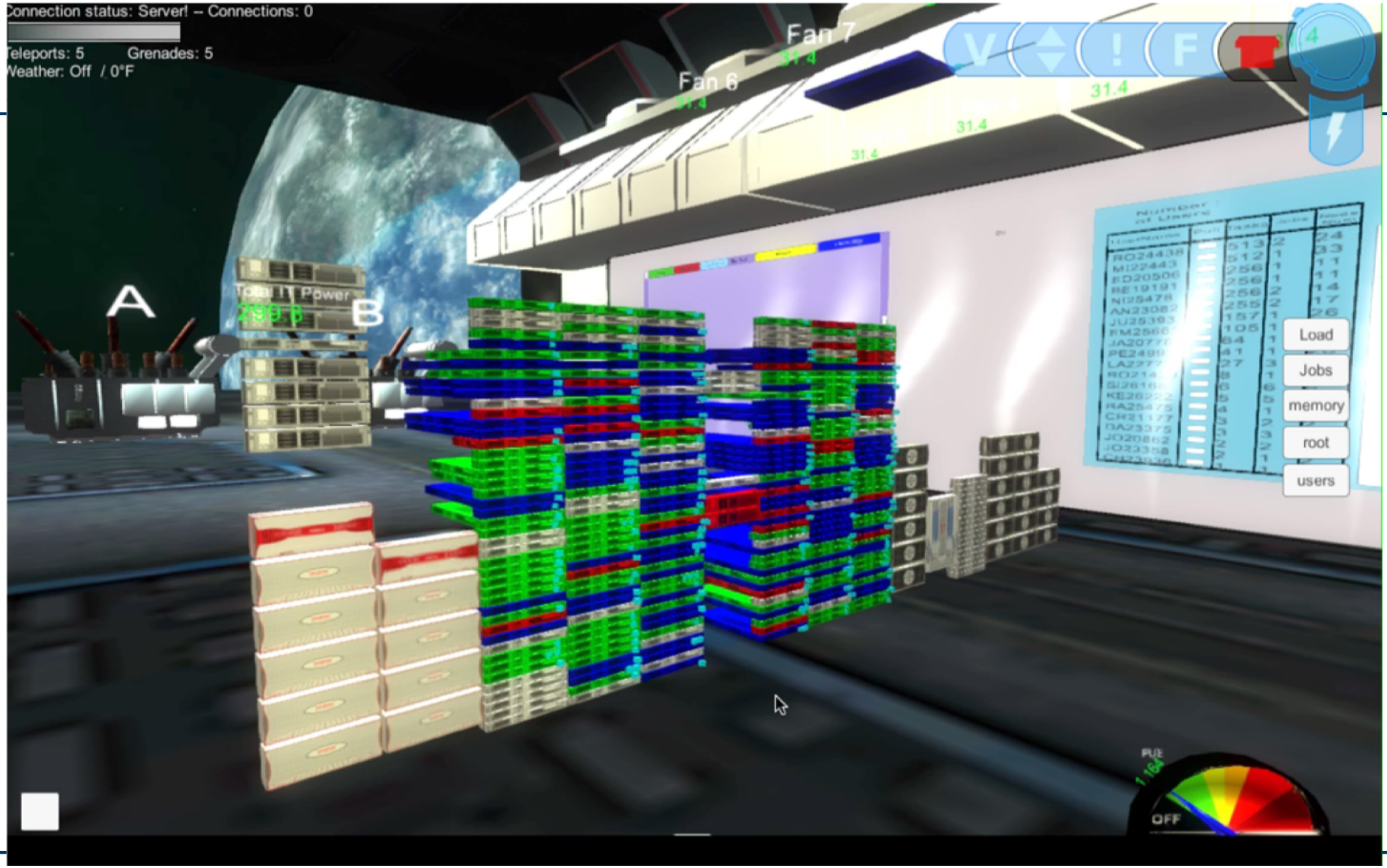
Dynamic High Performance Accumulo Database Lifecycle



LLSC System Monitoring Framework



CS&E: Computational Science and Engineering


Connection status: Server! -- Connections: 0
Teleports: 5 Grenades: 5
Weather: Off / 0°F
Fan 7
31.4
Fan 6
31.4
31.4
31.4
31.4
A
Total IT Power
299.6
B
Load
Jobs
memory
root
users
PUE
1.164
OFF

Outline

- Introduction
- LLSC Environment
- Slurm Migration
  - LLSC software stack
  - Slurm unique features
    - Lua job_submit script
    - SPANK plug-in module
- Summary

Consideration for Scheduler Migration

- Provide the same user experience
  - Seamless transition from the previous scheduler
    - Partitions based on jobs (historical reason)
      - pmatlab: parallel Matlab jobs and serial Matlab jobs launched with LaunchFunctionOnGrid
      - normal: all other jobs
- Support for modern hardware
  - Intel Knights Landing processor
  - Nvidia GPU co-processor
- Support for resource management
  - User limit on resources
    - CPU: allows to set different limits based on hardware
    - GPU
    - Memory: support Linux OS CGROUPS kernel feature

Current Slurm Usage

- SLURM 15.08.8 -> SLURM 16.05.10
  - Job array task dependency, “aftercorr”
- Separate partitions depending on mix of node types/work loads
  - Normal: General compute jobs (including DB services)
  - KNL: Knights Landing systems
  - GPU: GPU systems
- Several QoS’s
  - normal, pmatlab, high, db, and gpu
- LUA job_submit Plugin
  - Enforce various requirements for jobs
- Multi-factor priority scheduling
- SPANK Plugin - X11 forwarding for interactive jobs

LLSC Software Stack

- Scheduler specifics are abstracted away from users
  - pMatlab/gridMatlab: generate a sbatch command

    >> eval(pRUN('pMatlab_code', 256, 'grid'))

    - 'grid-opteron', 'grid-xeon-e5', 'grid-knl', and 'grid-gpu'

    >> setenv('GRIDMATL_CPU_TYPE','xeon-e5')

    >> LaunchFunctionOnGrid('matlab_func',var1,var2,var3)
  - LLMapReduce : generate a sbatch command

    $ LLMapReduce . . . --cpuType=CPUTYPE --gpuNameCount=GPUNAMECOUNT \

    --changeDepMode=true --options=SCHEDOPTIONS
  - LLsub: generate a sbatch or a salloc/srun command

    $ LLsub . . . -c CPU_type -q queue_name -g tesla:N
  - Web Portal Service: generate a sbatch command
    - Dynamic Database, Web services and Jupyter Notebook service

LLSC Software Stack (2)

- **LLGrid_status:**
  - **Provides the grid status information for a specific cpu nodes / partition**

```
$ LLGrid_status
LLGrid: txgreen (running slurm 16.05.10)
============================================
Online AMD opteron nodes: 205
   Unclaimed nodes: 170
   Claimed slots: 752
   Claimed slots for exclusive jobs: 456
--------------------------------------------
   Available slots: 4168
```

- **LLGrid_myjobs : Legacy command, calls LLstat**
  - **Provides my job status from the Matlab command prompt**
- **LLstat : scontrol & squeue**

```
$ LLstat
LLGrid: txgreen (running slurm 16.05.10)
JOBID              ARRAY_J    NAME           USER    START_TIME          PARTITION  CPUS  FEATURES  MIN_MEMORY  ST  NODELIST(REASON)
21758637           21758637   srun           CH21778 2017-08-24T08:10:50 gpu        28    xeon-e5   5G          R   b-1-18
21618994           21618994   srun           CH21778 2017-08-23T08:13:02 knl        64    xeon-phi  3000M       R   b-5-18-2
```

LUA job_submit Plug-in

- Enforce the default feature (used for CPU type) request if not specified (Slurm source modification)
- Enforce the high QoS for the interactive jobs
  - With the high QoS and the multi-factor priority scheduling, the interactive jobs are immediately scheduled

```
$ salloc --immediate --constraint=opteron srun --pty bash -i
```

salloc: error: Unable to allocate resources: Immediate execution impossible, insufficient priority

```
$ salloc --immediate --constraint=opteron --qos=high \
        srun --pty bash -i
```

salloc: Granted job allocation 4109683

- Enforce the GPU resource count to be 2 or 4
  - Slurm recognizes one K80 as two K40s
  - The GPU memory is cleared at the end of job
  - The epilog script clears the entire K80 memory

SPANK Plug-in Module

- SLURM SPANK X11 plug-in
  - Used for applications requires a graphical user interface.
  - Limited to interactive jobs only.

    https://github.com/hautreux/slurm-spank-x11
  - Needed to apply a patch to resolve the following issue:

    https://bugs.schedmd.com/show_bug.cgi?id=3503

- Redirecting TMP/TMPDIR
  - A per-job temporary directory plugin creates a directory on a local filesystem and exports it in the TMPDIR environment variable.
  - This provides similar behavior of the previous scheduler, open-source Grid Engine.

LLSC Resource Management

- Partition specific user association limits enforced
  - normal partition: GrpTRES=cpu=256
  - knl partition: GrpTRES=cpu=1024
  - gpu partition: GrpTRES=cpu=56,gres/gpu:tesla=16
- This allows to increase the per-user limit if needed
- Issue
  - User account for each partition needs to be created to enforce the partition-specific user association limit.
- Desired to handle a single user account to enforce the partition-specific user association limit

Immediate Job Support

- The immediate jobs (the --immediate flag) are important to LLSC users
  - Interactive, on-demand supercomputing resources for interactive jobs
  - pMatlab jobs
  - Database jobs
  - Jupyter notebooks jobs
- Immediate jobs were not scheduled immediately when there are jobs pending with some other reasons than the actual shortage of resources
  - At first it was not clear why this was happening

- Resolution: Multi-factor priority scheduling with a custom QoS
  - Configure slurm.conf for multi-factor priority scheduling
  - Create a high QoS
  - Attach the high Qos to any immediate jobs

Issues with Slurm Migration

- No queue (partition) based prolog/epilog available
  - Key feature required for database hosting
    - To switch accounts/privilege levels
    - To ensure uninterrupted execution (unprivileged job deletion by a user does not affect prolog/epilog, and the user switch prevents the prolog/epilog process from being sent signals
    - Easy ability to flag the job into an error state if something goes wrong
  - Overcome with a few scripts & setuid (user login - > db job started with griddb user)
- Supplementary group handling issue
  - No validation on the GID field of jobs at submit time
    - Any value can be submitted and added to the pending job
  - Validates the GID against only static group memberships when job executes
- No DRMAA JAVA binding available
  - gridMathematica

Summary

- LLSC completed the scheduler migration to Slurm successfully
- LLSC has been learning Slurm and exploited a number of features available to Slurm
  - LUA job_submit plug-in
  - SPANK plug-in module
  - Association limit enforcement
  - Multi-factor priority scheduling
  - QoS
  - Prolog/Epilog
  - Advance reservation
- Future work to exploit Slurm support for the second-generation Intel Xeon-Phi, Knights Landing, processor servers.
  - Machine learning algorithms
  - long-running, large-scale MPI jobs