SLURM Roadmap

Versions 15.xx and beyond
Moe Jette
jette@schedmd.com

Jacob Jenson
jacob@schedmd.com

Yiannis Georgiou
yiannis.georgiou@bull.net
V15.xx - Highlights

- Heterogeneous Environment
  - Asymmetric Resources and MPMD model
  - GPU Affinity

- Scalability
  - Support of PMI-x project
  - Messages Aggregation
  - HDF5 Profiling Framework
• Power Management and Energy Efficiency
  – Extension of Energy Accounting and Power Profiling Framework
  – Power-Capping logic in Job Scheduling
  – Energetic Fairsharing
Heterogeneous Environments
Asymmetric Resources

- Slurm, in its current stable versions provides a limited MPMD (Multiple Program Multiple Data) support.
  - Users can specify different binaries to be used within an parallel job but all the tasks are currently associated with the same resources requirements.

- We can call this Symmetric Resources Requirements Model (SRRM)
  - `srun -n4 -c4 --mem-per-core 2048 -C SSD ./myapp`
  - `srun -n4 -c2 --multi-prog myapps_descfile`

- SRRM **not very well suited to manage complex jobs**, like jobs with part of the code running on GPUs while another is running on standard CPUs with 2GB of RAM per core and a last part on CPUs with 8GB per core.
Asymmetric Resources

- Hence there is a need to extend the SRRM logic and move to what we could describe by the term "Asymmetric Resources Requirement Model" (ARRM)

- With ARRM, the idea is to describe a job by a set of tasks group, each tasks group having the same resources requirements.

- Examples of executions illustrating the targeted capability:
  
  ```
  srun -n 2 -c2 ./app1 : -n 4 --mem-per-core 256 --gres=gpu:2 ./app2
  ```

  - Or similarly
  ```
  sbatch -n 2 -c 2 : -n 4 --mem-per-core 256 --gres=gpu:2
  srun --task-group 0 ./app1 : --task-group 1 ./app2
  ```
GPU Affinity

- GPUDirect RDMA is a technology introduced in Kepler-class GPUs and CUDA 5.0
  - Strong affinity effect for GPU direct RDMA applications for both bandwidth and latency

- SLURM handles CPU affinity it should be extended to support affinity for both GPUs' and IB cards [1]

- Example of usage:
  - Two MPI tasks on two nodes
  - Each task wants to use GPU Direct RDMA

```
$ srun --gres=gpu:2 -N2 -n2 ./MPI_bandwidth_rdma
```

No explicit choice for IB and GPU
→ bad affinity (no luck...)

**GPU Affinity**

- Goal: Bind on GPU(s) closest to the CPU cores and Bind on IB cards closest to the GPU
- Introduced `--accel-bind=0|1|2`
- For each MPI task we set two environment variables

```
CUDA_VISIBLE_DEVICES=1
OMPI_MCA_btl_openib_if_include='ib1'
```

```
srun --gres=gpu:2 -N2 -n2 --accel-bind=2 ./MPI_bandwidth_rdma
```

Explicit choice for IB and GPU → good affinity is guaranteed
Scalability
Support of PMI-x

- PMI-2 has shown important scalability improvements when compared to PMI-1 but both standards are not suitable for exascale.

- **PMI-x** (exascale) aims to resolve these issues and tends to become the new standard to deal with Process Management in MPI for the exascale.

- Support of PMI-x is planned for the following SLURM versions.
Messages Aggregation

- Extensions in RPC messages exchanges to diminish the traffic between compute nodes and controller by aggregating them on particular compute nodes (collectors)
  - Higher scalability in terms of number of nodes

- Extensions in the processing logic of those new composite messages to improve the duration of the processing and the management of bigger number of messages
  - Higher scalability in terms of commands management (sinfo, squeue, etc)
HDF5 Profiling Framework

- Issues of the current Implementation
  - Plugins Architecture and Code
    - Not optimal usage of HDF5 API
    - Redundant code
    - HDF5 files: Space and time overhead
  - Structure of the HDF5 files
    - unclear and often inconvenient
    - Contain redundant data
HDF5 Profiling Framework

- Need for a new more scalable architecture
  - AcctGatherProfile should operate as a service
    - New Interface for profiling
    - Gathering plugins proceed in steps
  - Update AcctGatherProfileHDF5
    - Usage of high-level HDF5 API (H5 Packet Table)
    - Added possibilities for data compression
  - Update sh5util
    - Calculate statistics during merge and not during processing

Diagram:
- AcctGatherEnergy
- AcctGatherInfiniband
- AcctGatherFilesystem
- JobAcctGather
- AcctGatherProfileHDF5
- sh5util
- hdf5_api
HDF5 Profiling Framework

- Results: Profiling of a medium instance of HPLinpack upon 2 nodes (24min)

- Size of the profiling files:

<table>
<thead>
<tr>
<th></th>
<th>OLD (MiB)</th>
<th>NEW (MiB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node 1</td>
<td>17</td>
<td>0.64</td>
</tr>
<tr>
<td>Node 2</td>
<td>9.8</td>
<td>0.37</td>
</tr>
<tr>
<td>Total</td>
<td>26.8</td>
<td><strong>1.01</strong></td>
</tr>
</tbody>
</table>

- Time to merge per-node profiling files in one:

<table>
<thead>
<tr>
<th></th>
<th>OLD (sec)</th>
<th>NEW (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Merge-Time</td>
<td>6.477</td>
<td><strong>0.077</strong></td>
</tr>
</tbody>
</table>

Work done by
Yoann Blein
Internship Summer 2014
Power Management
Extensions in Energy Accounting and Power Profiling

- Support of finer-grained energy accounting and power profiling
  - Extending AcctGatherEnergy
    - Possible to record a variable number of fields
    - New configuration format to describe sensors:
      EnergyIPMIPowerSensors =Node=1,2,3;CPU=3;RAM=1,2
  - Extending AcctGatherProfileHDF5

Work done in BULL by
Yoann Blein
Internship Summer 2014
Extensions in Energy Accounting and Power Profiling

- New plugin (ipmi-raw) to support particular BMC functionality and support of FPGA that enables high-resolution monitoring of sensors' energy consumption
  - Project HDEEM (in collaboration with TU Dresden)
Power Capping Logic in Job Scheduling

- Version based upon layouts
  - Option to take into account the theoretical values as given statically in the layouts
  - Or integration with IPMI and dynamic updates of power consumption of nodes
Energetic Fairsharing

- Energy consumption can be accounted and charged independently,
  - Real need for fairness in terms of energy
- New parameter in multi-factor plugin to deal with fair-share scheduling based on past energy usage.
  - Feature to motivate users for more energy efficient codes / usage of resources
Current Works

- Multi-parametric scheduling
  - MOEBUS Project ([http://moebus.gforge.inria.fr/](http://moebus.gforge.inria.fr/))
    - 4 years ANR (French funded) project started October 2013
Current Works

• Job placement based on communications patterns
  – Support of treematch (http://treematch.gforge.inria.fr/) algorithm directly in the resources selection plugin of SLURM
Current Works

- Support of new fair scheduling algorithm in SLURM [1]

Unfinished Works

- Support of Licenses Manager (FlexLM)
- Slurm – Hadoop Integration
- Support of PAM with cgroups

Any Volunteers!!
Other Features?

- A lot of ideas and interesting features
- Sometimes overlapping contexts and concurrent proposals
  - Prior communication and exchanges would help to concentrate efforts for common interests
- **Proposal:** Create a new web page summarizing current developments and providing contact information to promote collaboration and sharing of ideas … or another mailing list