Generalized Hypercube (GHC)

A topology plugin

M. Clayer
A. Faure

September 25, 2018
HyperCube

Generalized HyperCube

Slurm configuration

Examples
A \textit{n-dimensional unit hypercube} is defined by $2^n$ point, which coordinates are composed by 0 or 1. These points represent the corners of the unit hypercube. For $n = 2$: a square, $n = 3$: a cube.
Unit Hypercube:

A \textit{n-dimensional unit hypercube} is defined by $2^n$ points, which coordinates are composed by 0 or 1. These points represent the corners of the unit hypercube. For $n = 2$: a square, $n = 3$: a cube.
Hypercube topology

Unit Hypercube:

A \textit{n-dimensional unit hypercube} is defined by \(2^n\) point, which coordinates are composed by 0 or 1. These points represent the corners of the unit hypercube. For \(n = 2\): a square, \(n = 3\): a cube.

\(\text{two cubes, } n = 3\)
Hypercube topology

Unit Hypercube:

A **n-dimensional unit hypercube** is defined by $2^n$ point, which coordinates are composed by 0 or 1. These points represent the corners of the unit hypercube. For $n = 2$: a square, $n = 3$: a cube. $a$ tesseract, $n = 4$
Hypercube topology

Hypercube Topology:

- Each corner represent a switch

A tesseract, $n = 4$
Hypercube topology

Hypercube Topology:

- Each corner represents a switch
- We connect $T$ terminals on each switch

A tesseract, $n = 4$
Hypercube topology

Hypercube Topology:

- Each corner represents a switch
- We connect $T$ terminals on each switch
- Maximum of $T \times 2^n$ terminals

A tesseract, $n = 4$
Hypercube Topology:

- Each corner represents a switch.
- We connect $T$ terminals on each switch.
  - Maximum of $T \times 2^n$ terminals.
- Number of hops between 2 terminals: $2 + n$.

A tesseract, $n = 4$. 

Hypercube Topology
Hypercube topology

Hypercube Topology:

- Each corner represents a switch
- We connect $T$ terminals on each switch
  - Maximum of $T \times 2^n$ terminals
- Number of hops between 2 terminals: $2 + n$

Limitation

*Hypercube have a strong constraint: the number of switches: $2^n$*

A tesseract, $n = 4$
Hypercube topology

Hypercube Topology:
- Each corner represents a switch
- We connect $T$ terminals on each switch
  - Maximum of $T \times 2^n$ terminals
- Number of hops between 2 terminals: $2 + n$

Limitation
Hypercube have a strong constraint: the number of switches: $2^n$

Solution
A similar topology avoids this constraint: the Generalized HyperCube topology (GHC)

a tesseract, $n = 4$
Defining a n-dimensional GHC topology by:

- a number of switches for each dimension: $S_i$
  
  $\Rightarrow$ number total of switches: $\prod_{i=1}^{n} S_i$
Generalized HyperCube

**GHC**

*Defining a n-dimensional GHC topology by:*

- a number of switches for each dimension: $S_i$

  $\Rightarrow$ *number total of switches:* $\prod_{i=1}^{n} S_i$

---

*Proposition*

2 switches are linked $\iff$ Their coordinates differ by only one coordinate.
Defining a n-dimensional GHC topology by:
- a number of switches for each dimension: \( S_i \)
- number total of switches: \( \prod_{i=1}^{n} S_i \)

\[ n = 2, \ S = (3, 3) \]
Defining a $n$-dimensional GHC topology by:

- a number of switches for each dimension: $S_i$
  
  $\Rightarrow$ number total of switches: $\prod_{i=1}^{n} S_i$

- a number of terminal $T$ per switch

$n = 2$, $S = (3, 3)$
Generalized HyperCube

**Defining a n-dimensional GHC topology by:**

- a number of switches for each dimension: $S_i$
  
  $⇒$ number total of switches: $\prod_{i=1}^{n} S_i$

- a number of terminal $T$ per switch

- in each dimension, switches are fully connected

$n = 2, S = (3, 3)$
Generalized HyperCube

Defining a $n$-dimensional GHC topology by:

- a number of switches for each dimension: $S_i$
  \( \Rightarrow \) number total of switches: \( \prod_{i=1}^{n} S_i \)
- a number of terminal $T$ per switch
- in each dimension, switches are fully connected

\( n = 2, S = (3, 3) \)
Defining a n-dimensional GHC topology by:

- a number of switches for each dimension: $S_i$
  
  \[ \Rightarrow \text{number total of switches: } \prod_{i=1}^{n} S_i \]

- a number of terminal $T$ per switch

- in each dimension, switches are fully connected

$n = 2, S = (4, 3)$
Defining a n-dimensional GHC topology by:
- a number of switches for each dimension: \( S_i \)
- ⇒ number total of switches: \( \prod_{i=1}^{n} S_i \)
- a number of terminal \( T \) per switch
- in each dimension, switches are fully connected

\[ n = 2, \ S = (4, 3) \]
Generalized HyperCube

Defining a n-dimensional GHC topology by:

- a number of switches for each dimension: $S_i$
- number total of switches: $\prod_{i=1}^{n} S_i$
- a number of terminal $T$ per switch
- in each dimension, switches are fully connected

Note

GHC topology can be represented in a n-dimensional euclidian space

$n = 2, S = (4, 3)$
Defining a $n$-dimensional GHC topology by:

- a number of switches for each dimension: $S_i$
- number total of switches: $\prod_{i=1}^{n} S_i$
- a number of terminal $T$ per switch
- in each dimension, switches are fully connected

Note

GHC topology can be represented in a $n$-dimensional euclidian space
⇒ switches have coordinates!

$n = 2$, $S = (4, 3)$
**GHC**

Defining a \( n \)-dimensional GHC topology by:

- a number of switches for each dimension: \( S_i \)
  - ⇒ number total of switches: \( \prod_{i=1}^{n} S_i \)
- a number of terminal \( T \) per switch
- in each dimension, switches are fully connected

**Note**

GHC topology can be represented in a \( n \)-dimensional euclidian space
  ⇒ switches have coordinates!

**Proposition**

2 switches are linked ⇔ Their coordinates differ by only one coordinate
Framework

**topology.conf**

SwitchName=sw1 Nodes=n0 Switches=sw2,sw4  
SwitchName=sw2 Nodes=n1 Switches=sw1,sw3  
SwitchName=sw3 Nodes=n2 Switches=sw2,sw4  
SwitchName=sw4 Nodes=n3 Switches=sw1,sw3

Initialisation

topology.conf permit to compute:

▶ topology parameters (dimension \( n \) and \( S \))

▶ set up coordinates on switches

slurm.conf

TopologyPlugin=topology/ghc

SelectType=select/linear

▶ use of Slurm best fit algorithm

⇒ linear path across the GHC topology
### Framework

#### topology.conf

- `SwitchName=sw1  Nodes=n0  Switches=sw2,sw4`
- `SwitchName=sw2  Nodes=n1  Switches=sw1,sw3`
- `SwitchName=sw3  Nodes=n2  Switches=sw2,sw4`
- `SwitchName=sw4  Nodes=n3  Switches=sw1,sw3`

#### Initialisation

- `topology.conf` permit to compute:
  - topology parameters (dimension n and S)
  - set up coordinates on switches
Framework

**topology.conf**

`SwitchName=sw1 Nodes=n0 Switches=sw2,sw4
SwitchName=sw2 Nodes=n1 Switches=sw1,sw3
SwitchName=sw3 Nodes=n2 Switches=sw2,sw4
SwitchName=sw4 Nodes=n3 Switches=sw1,sw3`

**slurm.conf**

`TopologyPlugin=topology/ghc
SelectType=select/linear`

**Initialisation**

topology.conf permit to compute:
- topology parameters (dimension $n$ and $S$)
- set up coordinates on switches
topology.conf

SwitchName=sw1 Nodes=n0 Switches=sw2,sw4
SwitchName=sw2 Nodes=n1 Switches=sw1,sw3
SwitchName=sw3 Nodes=n2 Switches=sw2,sw4
SwitchName=sw4 Nodes=n3 Switches=sw1,sw3

Initialisation

topology.conf permit to compute:
▶ topology parameters (dimension $n$ and $S$)
▶ set up coordinates on switches

slurm.conf

TopologyPlugin=topology/ghc
SelectType=select/linear

select linear

▶ use of Slurm best fit algorithm
⇒ linear path across the GHC topology
How to get a linear path

- Hilbert’s curve
  - map the switches to n-dimensional space into a 1-dimensional space
  - achieve a high degree of locality for jobs

2D Hilbert’s curve
linear Path & switches selection

How to get a linear path

- Hilbert’s curve
  - map the switches to $n$-dimensional space into a 1-dimensional space
  - achieve a high degree of locality for jobs

switches selection

- loop through the Hilbert curve
  - create a cluster: of neighboring nodes
  - compute the variance for this cluster: based on the distance set, between each cluster’s nodes.
- choose the cluster with the lowest variance

2D Hilbert’s curve
Example - scontrol show topology

\[ n = 2, \ S = (4, 4), \ \text{with 1 node per switch} \]

```
slurm$ scontrol show topology
HGC NbSwitches: 16 Dimensions: 2
Dimension 1: 4 Dimension 2: 4
SwitchName=sw1 NodeCount=2 Nodes=node[0-1]
Switches=sw2,sw3,sw4,sw5,sw9,sw13
```

\[ 2D, \ S = (4, 4) \]
Example - jobs allocation

Example - launch of 4 tasks

\( n = 2, \ S = (4, 4), \) with 1 node per switch

launch multiple `srun -n4`:

```
slurm$ srun -n4 sleep 50 &
slurm$ squeue

<table>
<thead>
<tr>
<th>JOBID</th>
<th>NAME</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES</th>
<th>NODELIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>153</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:02</td>
<td>4</td>
<td>node[1-4]</td>
</tr>
</tbody>
</table>
```

\( 2D, \ S = (4, 4) \)
Example - jobs allocation

Example - launch of 4 tasks

$n = 2$, $S = (4, 4)$, with 1 node per switch

launch multiple `srun -n4`:

```
slurm$ srun -n4 sleep 50 &
slurm$ squeue
```

<table>
<thead>
<tr>
<th>JOBID</th>
<th>NAME</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES</th>
<th>NODELIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>153</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:02</td>
<td>4</td>
<td>node[1-4]</td>
</tr>
</tbody>
</table>
Example - jobs allocation

Example - launch of 4 tasks

\( n = 2, \ S = (4, 4) \), with 1 node per switch

launch multiple srun -n4:

```
slurm$ srun -n4 sleep 50 &
slurm$ squeue

<table>
<thead>
<tr>
<th>JOBID</th>
<th>NAME</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES</th>
<th>NODELIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>154</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:02</td>
<td>4</td>
<td>node[5-8]</td>
</tr>
<tr>
<td>153</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:07</td>
<td>4</td>
<td>node[1-4]</td>
</tr>
</tbody>
</table>
```

\( 2D, \ S = (4, 4) \)
Example - jobs allocation

Example - launch of 4 tasks

$n = 2$, $S = (4, 4)$, with 1 node per switch

launch multiple `srun -n4`:

```
slurm$ srun -n4 sleep 50 &
slurm$ squeue

<table>
<thead>
<tr>
<th>JOBID</th>
<th>NAME</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES</th>
<th>Nodelist</th>
</tr>
</thead>
<tbody>
<tr>
<td>154</td>
<td>sleep slurm R</td>
<td>0:02</td>
<td>4</td>
<td>node[5-8]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>153</td>
<td>sleep slurm R</td>
<td>0:07</td>
<td>4</td>
<td>node[1-4]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

$2D$, $S = (4, 4)$
Example - jobs allocation

Example - launch of 4 tasks

\( n = 2, \ S = (4, 4), \) with 1 node per switch
launch multiple `srun -n4`:

```
slurm$ srun -n4 sleep 50 &
slurm$ squeue
```

<table>
<thead>
<tr>
<th>JOBID</th>
<th>NAME</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES</th>
<th>NODELIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>155</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:02</td>
<td>4</td>
<td>node[9-10,13-14]</td>
</tr>
<tr>
<td>154</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:07</td>
<td>4</td>
<td>node[5-8]</td>
</tr>
<tr>
<td>153</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:12</td>
<td>4</td>
<td>node[1-4]</td>
</tr>
</tbody>
</table>

\( 2D, \ S = (4, 4) \)
Example - jobs allocation

**Example - launch of 4 tasks**

\( n = 2, S = (4, 4) \), with 1 node per switch
launch multiple srun -n4:

```
slurm$ srun -n4 sleep 50 &
slurm$ squeue
```

<table>
<thead>
<tr>
<th>JOBID</th>
<th>NAME</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES</th>
<th>NODELIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>155</td>
<td>sleep</td>
<td>slurm</td>
<td>0:02</td>
<td>4</td>
<td>node[9-10,13-14]</td>
<td></td>
</tr>
<tr>
<td>154</td>
<td>sleep</td>
<td>slurm</td>
<td>0:07</td>
<td>4</td>
<td>node[5-8]</td>
<td></td>
</tr>
<tr>
<td>153</td>
<td>sleep</td>
<td>slurm</td>
<td>0:12</td>
<td>4</td>
<td>node[1-4]</td>
<td></td>
</tr>
</tbody>
</table>

\( 2D, S = (4, 4) \)
Example - jobs allocation

Example - launch of 4 tasks

\( n = 2, \ S = (4, 4), \) with 1 node per switch

launch multiple `srun -n4`:

```
srun -n4 sleep 50 &
```

```
squeue
```

<table>
<thead>
<tr>
<th>JOBID</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES</th>
<th>Nodelist</th>
</tr>
</thead>
<tbody>
<tr>
<td>156</td>
<td>slurm</td>
<td>0:02</td>
<td>4</td>
<td>node[11-12,15-16]</td>
<td></td>
</tr>
<tr>
<td>155</td>
<td>slurm</td>
<td>0:07</td>
<td>4</td>
<td>node[9-10,13-14]</td>
<td></td>
</tr>
<tr>
<td>154</td>
<td>slurm</td>
<td>0:12</td>
<td>4</td>
<td>node[5-8]</td>
<td></td>
</tr>
<tr>
<td>153</td>
<td>slurm</td>
<td>0:17</td>
<td>4</td>
<td>node[1-4]</td>
<td></td>
</tr>
</tbody>
</table>

\( 2D, \ S = (4, 4) \)
Example - launch of 4 tasks

\( n = 2, S = (4, 4) \), with 1 node per switch

launch multiple slurm -n4:

\[
\begin{align*}
\text{slurm}\$ \text{srun -n4 sleep 50} & \quad & \text{slurm}\$ \text{squeue} \\
\end{align*}
\]

<table>
<thead>
<tr>
<th>JOBID</th>
<th>NAME</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES</th>
<th>NODELIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>156</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:02</td>
<td>4</td>
<td>node[11-12,15-16]</td>
</tr>
<tr>
<td>155</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:07</td>
<td>4</td>
<td>node[9-10,13-14]</td>
</tr>
<tr>
<td>154</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:12</td>
<td>4</td>
<td>node[5-8]</td>
</tr>
<tr>
<td>153</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:17</td>
<td>4</td>
<td>node[1-4]</td>
</tr>
</tbody>
</table>

\( 2D, S = (4, 4) \)

Example - jobs allocation
Example - launch of 3 tasks

\( n = 2, \ S = (4, 4), \) with 1 node per switch

launch multiple `srun -n3`:

```bash
slurm$ srun -n3 sleep 50 &
slurm$ squeue
```

<table>
<thead>
<tr>
<th>JOBID</th>
<th>NAME</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES</th>
<th>NODELIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>197</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:01</td>
<td>3</td>
<td>node[1,3-4]</td>
</tr>
<tr>
<td>196</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:05</td>
<td>3</td>
<td>node[7,11,15]</td>
</tr>
<tr>
<td>195</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:09</td>
<td>3</td>
<td>node[8,12,16]</td>
</tr>
<tr>
<td>194</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:11</td>
<td>3</td>
<td>node[2,6,14]</td>
</tr>
<tr>
<td>193</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:15</td>
<td>3</td>
<td>node[5,9,13]</td>
</tr>
</tbody>
</table>
```

\( 2D, \ S = (4, 4) \)
Example - jobs allocation

Example - launch of 3 tasks

\( n = 2, S = (4, 4), \) with 1 node per switch

launch multiple srun -n3:

```
slurm$ srun -n3 sleep 50 &
slurm$ squeue
```

<table>
<thead>
<tr>
<th>JOBID</th>
<th>NAME</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES NODELIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>197</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:01</td>
<td>3 node[1,3-4]</td>
</tr>
<tr>
<td>196</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:05</td>
<td>3 node[7,11,15]</td>
</tr>
<tr>
<td>195</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:09</td>
<td>3 node[8,12,16]</td>
</tr>
<tr>
<td>194</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:11</td>
<td>3 node[2,6,14]</td>
</tr>
<tr>
<td>193</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:15</td>
<td>3 node[5,9,13]</td>
</tr>
</tbody>
</table>

\( 2D, S = (4, 4) \)
Example - jobs allocation

Example - launch of 3 tasks

\[ n = 2, \quad S = (4, 4), \quad \text{with 1 node per switch} \]

launch multiple srun -n3:

```
slurm$ srun -n3 sleep 50 &
slurm$ squeue

<table>
<thead>
<tr>
<th>JOBID</th>
<th>NAME</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES</th>
<th>NODELIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>197</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:01</td>
<td>3</td>
<td>node[1,3-4]</td>
</tr>
<tr>
<td>196</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:05</td>
<td>3</td>
<td>node[7,11,15]</td>
</tr>
<tr>
<td>195</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:09</td>
<td>3</td>
<td>node[8,12,16]</td>
</tr>
<tr>
<td>194</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:11</td>
<td>3</td>
<td>node[2,6,14]</td>
</tr>
<tr>
<td>193</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:15</td>
<td>3</td>
<td>node[5,9,13]</td>
</tr>
</tbody>
</table>
```

\[ 2D, \quad S = (4, 4) \]
Example - launch of 3 tasks

\( n = 2, \ S = (4, 4) \), with 1 node per switch

launch multiple `srun -n3`:

```
slurm$ srun -n3 sleep 50 &
slurm$ squeue
```

<table>
<thead>
<tr>
<th>JOBID</th>
<th>NAME</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES</th>
<th>NODELIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>197</td>
<td>sleep slurm</td>
<td>R 0:01</td>
<td>3</td>
<td>node[1,3-4]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>196</td>
<td>sleep slurm</td>
<td>R 0:05</td>
<td>3</td>
<td>node[7,11,15]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>195</td>
<td>sleep slurm</td>
<td>R 0:09</td>
<td>3</td>
<td>node[8,12,16]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>194</td>
<td>sleep slurm</td>
<td>R 0:11</td>
<td>3</td>
<td>node[2,6,14]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>193</td>
<td>sleep slurm</td>
<td>R 0:15</td>
<td>3</td>
<td>node[5,9,13]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( 2D, \ S = (4, 4) \)
Example - jobs allocation

Example - launch of 3 tasks

\[ n = 2, \ S = (4, 4), \] with 1 node per switch

launch multiple `srun -n3`:

```bash
slurm$ srun -n3 sleep 50 &
slurm$ squeue
```

<table>
<thead>
<tr>
<th>JOBID</th>
<th>NAME</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES</th>
<th>NODELIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>197</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:01</td>
<td>3</td>
<td>node[1,3-4]</td>
</tr>
<tr>
<td>196</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:05</td>
<td>3</td>
<td>node[7,11,15]</td>
</tr>
<tr>
<td>195</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:09</td>
<td>3</td>
<td>node[8,12,16]</td>
</tr>
<tr>
<td>194</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:11</td>
<td>3</td>
<td>node[2,6,14]</td>
</tr>
<tr>
<td>193</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:15</td>
<td>3</td>
<td>node[5,9,13]</td>
</tr>
</tbody>
</table>
```

\[ 2D, \ S = (4, 4) \]
Example - jobs allocation

Example - launch of 3 tasks

\( n = 2, S = (4, 4) \), with 1 node per switch

launch multiple srun -n3:

```
slurm$ srun -n3 sleep 50 &
slurm$ squeue
```

<table>
<thead>
<tr>
<th>JOBID</th>
<th>NAME</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES</th>
<th>NODELIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>197</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:01</td>
<td>3</td>
<td>node[1,3-4]</td>
</tr>
<tr>
<td>196</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:05</td>
<td>3</td>
<td>node[7,11,15]</td>
</tr>
<tr>
<td>195</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:09</td>
<td>3</td>
<td>node[8,12,16]</td>
</tr>
<tr>
<td>194</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:11</td>
<td>3</td>
<td>node[2,6,14]</td>
</tr>
<tr>
<td>193</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:15</td>
<td>3</td>
<td>node[5,9,13]</td>
</tr>
</tbody>
</table>

\[ 2D, S = (4, 4) \]
Example - launch of 800 tasks

\( n = 6, \ S = (2, 3, 3, 5, 5) \) (1350 switches), with 1 node per switch
launch `srun -n800 sleep 120`:

```
slurm$ srun -n800 sleep 120 &
slurm$ squeue
```

<table>
<thead>
<tr>
<th>JOBID</th>
<th>NAME</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>sleep</td>
<td>slurm</td>
<td>R</td>
<td>0:50</td>
<td>node[0-11,18-29,36-39,42-45,54-65,72-83,90-93,96-99,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>108-119,126-137,144-147,150-153,162-173,180-191,198-201,204-207,216-227,234-245,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>522-525,528-531,540-551,558-569,576-579,582-585,594-605,612-623,630-633,636-639,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>648-659,666-677,684-687,690-693,702-713,720-731,738-741,744-747,756-767,774-785,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>792-795,798-801,810-821,828-839,846-849,852-855,864-875,882-893,900-903,906-909,918-929,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>936-947,954-957,960-963,972-983,990-1001,1008-1011,1014-1017,1026-1037,1044-1055,1062-1065,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1068-1071,1080-1091,1098-1109,1116-1119,1122-1125,1134-1145,1152-1163,1170-1173,1176-1179</td>
</tr>
</tbody>
</table>
```
Future work

- GHC with select cons_res
- Scalability and Efficiency evaluation
- Validate on a physical cluster
- Push to the community
Thanks for your attention!

Any questions?

Contact: mathis.clayer@atos.net