BeeGFS in the DEEP/-ER Project

Cristina Manzano, Jülich Supercomputing Centre
EU-Exascale projects
20 partners
Total budget: 28,3 M€
EU-funding: 14,5 M€
Nov 2011 – Sept 2016

Visit us @ ISC’16, Frankfurt (Germany)
20.-22.06.2016
-Booth
-BoF
-Workshop
What are the projects about?

**DEEP**
- **Cluster-Booster archit.**
- Software stack
- Programming environ.
- Energy efficiency

**Applications:**
- Co-design
- Evaluation/demonstration
- Code modernisation

**DEEP-ER**
- Extend memory hierarchy
- High-performance I/O
- Scalable **resiliency**

**Applications:**
- Co-design
- Evaluation/demonstration
- Code modernisation
“Standard” heterogeneity

InfiniBand

Flat topology
Simple management of resources

Static assignment of accelerators to CPUs
Accelerators cannot act autonomously
Cluster-Booster architecture

Flexible assignment of resources (CPUs, accelerators)
Direct communication between accelerators
“Offload” of large and complex parts of applications
• Installed at JSC
• 1,5 racks
• 500 TFlop/s peak perf.
• 3.5 GFlop/s/W
• Water cooled
DEEP System

File Servers
(6 Xeon Sandy Bridge)

- /work file system
- ~2000 MB/s write/read BW*

* Measured with IOR benchmark.

JBOD 2245
(45x 2TB disks)
**DEEP Storage servers**
- 6x DELL PowerEdge R520 storage servers (deep-fs01 – deep-fs06)

**SAS switch**
- 1x LSI 6140 SAS switch connecting the storage servers with the JBOD

**JBOD**
- 1x SGI JBOD 2245 with 45x 2TB disks

**Storage space on each server**

<table>
<thead>
<tr>
<th>Server</th>
<th>RAID Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>deep-fs01</td>
<td>RAID1: 2x mirrored disks</td>
</tr>
<tr>
<td>deep-fs02</td>
<td>RAID1: 2x mirrored disks</td>
</tr>
<tr>
<td>deep-fs03</td>
<td>RAID6: 10x disks</td>
</tr>
<tr>
<td>deep-fs04</td>
<td>RAID6: 10x disks</td>
</tr>
<tr>
<td>deep-fs05</td>
<td>RAID6: 10x disks</td>
</tr>
<tr>
<td>deep-fs06</td>
<td>RAID6: 10x disks</td>
</tr>
</tbody>
</table>
## BeeGFS configuration

<table>
<thead>
<tr>
<th>Node</th>
<th>Description</th>
<th>BeeGFS roles</th>
<th>BeeGFS services</th>
</tr>
</thead>
<tbody>
<tr>
<td>deep-fs01</td>
<td>Storage node</td>
<td>Management, Metadata, Administration, Monitoring, Helper</td>
<td>beegfs-mgmd, beegfs-meta, beegfs-admon, beegfs-helperd</td>
</tr>
<tr>
<td>deep-fs02</td>
<td>Storage node</td>
<td>Metadata, Helper</td>
<td>beegfs-meta, beegfs-helperd</td>
</tr>
<tr>
<td>deep-fs0[3-6]</td>
<td>Storage nodes</td>
<td>Storage, Helper</td>
<td>beegfs-storage, beegfs-helperd</td>
</tr>
<tr>
<td>deep[1-128]</td>
<td>Compute nodes</td>
<td>Client, Helper</td>
<td>beegfs-client, beegfs-helperd</td>
</tr>
<tr>
<td>deepm</td>
<td>Administration (master) node</td>
<td>Client, Helper</td>
<td>beegfs-client, beegfs-helperd</td>
</tr>
<tr>
<td>deepl</td>
<td>Login node</td>
<td>Client, Helper</td>
<td>beegfs-client, beegfs-helperd</td>
</tr>
</tbody>
</table>
What are the projects about?

DEEP

- **Cluster-Booster archit.**
- Software stack
- Programming environ.
- Energy efficiency

- Applications:
  - Co-design
  - Evaluation/demonstration
  - Code modernisation

DEEP-ER

- Extend memory hierarchy
- High-performance I/O
- Scalable **resiliency**

- Applications:
  - Co-design
  - Evaluation/demonstration
  - Code modernisation
What are the projects about?

DEEP
- **Cluster-Booster archit.**
- Software stack
- Programming environ.
- Energy efficiency
- Applications:
  - Co-design
  - Evaluation/demonstration
  - Code modernisation

DEEP-ER
- Extend memory hierarchy
- High-performance I/O
- Scalable resiliency
- Applications:
  - Co-design
  - Evaluation/demonstration
  - Code modernisation
Enhance DEEP architecture

Legend:
CN: Cluster Node
BN: Booster Node
NIC: Network Interface Card
NAM: Network Attached Memory
NVM: Non Volatile Memory
Software Development Vehicle (SDV)

- /sdv-work file system
- ~1500 MB/s write/read BW*

*Measured with BeeGFS benchmark. IOR benchmarking ongoing work.
DEEP-ER Storage servers

- 3x DELL PowerEdge R530 storage servers (deeper-fs01 – deeper-fs03)

Metadata

- 2x internal SSD disks

RAID System

- 1x EUROstor ES-6600 with 4 x 8Gbit FC connector
- 24x 6 TB SAS Nearline (RAID6)
- 4x 31500.0GB Volumes (2 unused for future expansion of the storage system)

Storage space on each server

<table>
<thead>
<tr>
<th>Server</th>
<th>Storage Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>deeper-fs01</td>
<td>RAID1: 2x internal SSD disks (mirrored)</td>
</tr>
<tr>
<td>deeper-fs02</td>
<td>1x 31500.0GB Volume</td>
</tr>
<tr>
<td>deeper-fs03</td>
<td>1x 31500.0GB Volume</td>
</tr>
</tbody>
</table>
## BeeGFS configuration

<table>
<thead>
<tr>
<th>Node</th>
<th>Description</th>
<th>BeeGFS roles</th>
<th>BeeGFS services</th>
</tr>
</thead>
<tbody>
<tr>
<td>deeper-fs01</td>
<td>Storage node</td>
<td>Management, Metadata, Administration, Monitoring, Helper</td>
<td>beegfs-mgmtd, beegfs-meta, beegfs-admon, beegfs-helperd</td>
</tr>
<tr>
<td>deeper-fs0[2-3]</td>
<td>Storage node</td>
<td>Storage, Helper</td>
<td>beegfs-storage, beegfs-helperd</td>
</tr>
<tr>
<td>deeper-sdv[01-16]</td>
<td>Compute nodes</td>
<td>Client, Helper</td>
<td>beegfs-client, beegfs-helperd</td>
</tr>
<tr>
<td>deepm</td>
<td>Administration (master) node</td>
<td>Client, Helper</td>
<td>beegfs-client, beegfs-helperd</td>
</tr>
<tr>
<td>deepl</td>
<td>Login node</td>
<td>Client, Helper</td>
<td>beegfs-client, beegfs-helperd</td>
</tr>
</tbody>
</table>
I/O and Resiliency in DEEP-ER

Optimized I/O

- Hierarchical global FS
  - Fast caches (NVMe)
- SIONlib & E10
  - Address the “small I/O” problem

Enhanced resiliency

- Enhanced SCR
  - Built on top of the optimized I/O
- Task-based resiliency
NVMe SSD devices

- NVM component: Intel DC P3700
  - 20nm MLC NAND Flash technology
  - PCI Express generation 3 × 4
- 1 NVMe with 400 GB attached to each node in Cluster and Booster
- 1 BeeOND instance running on each NVMe device
- BeeGFS cache layer
  - Local tier in a multi-tier storage environment
  - Burst buffer for temporary storage (like checkpointing)
- More about this in Frank’s talk later today!
Performance NVMe ext4 vs. BeeOND running on NVMe:
<table>
<thead>
<tr>
<th>testdir</th>
<th>itemspertask</th>
<th>filesperdir</th>
<th>FCreateMax [ops/sec]</th>
<th>FRemoveMax [ops/sec]</th>
</tr>
</thead>
<tbody>
<tr>
<td>/nvme/tmp/</td>
<td>41666</td>
<td>651</td>
<td>127.947,63</td>
<td>58.818,83</td>
</tr>
<tr>
<td>/mnt/beeond/</td>
<td>41666</td>
<td>651</td>
<td>12.158,12</td>
<td>16.653,14</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------</td>
<td>-----------------</td>
<td>-----------</td>
<td>--------------</td>
</tr>
<tr>
<td>nvme/tmp/</td>
<td>POSIX</td>
<td>file-per-process</td>
<td>10GB</td>
<td>16MB</td>
</tr>
<tr>
<td>mnt/beeond/</td>
<td>POSIX</td>
<td>file-per-process</td>
<td>10GB</td>
<td>16MB</td>
</tr>
<tr>
<td>nvme/tmp/</td>
<td>MPIIO</td>
<td>file-per-process</td>
<td>10GB</td>
<td>16MB</td>
</tr>
<tr>
<td>mnt/beeond/</td>
<td>MPIIO</td>
<td>file-per-process</td>
<td>10GB</td>
<td>16MB</td>
</tr>
<tr>
<td>nvme/tmp/</td>
<td>POSIX</td>
<td>single-shared-file</td>
<td>10GB</td>
<td>16MB</td>
</tr>
<tr>
<td>mnt/beeond/</td>
<td>POSIX</td>
<td>single-shared-file</td>
<td>10GB</td>
<td>16MB</td>
</tr>
<tr>
<td>nvme/tmp/</td>
<td>MPIIO</td>
<td>single-shared-file</td>
<td>10GB</td>
<td>16MB</td>
</tr>
<tr>
<td>mnt/beeond/</td>
<td>MPIIO</td>
<td>single-shared-file</td>
<td>10GB</td>
<td>16MB</td>
</tr>
</tbody>
</table>
Some lessons learned

- BeeGFS is really easy to update also between major releases
  -> Script provided for updating between 2014 (FhGFS) and 2015 (BeeGFS)
- BeeGFS runs really stable
- Don’t underestimate the use of extended attributes!
  -> Gain factor of 50 (from 130 to 6300 files/second with an mdtest)
- Some users want to be able to change the stripping settings
  -> New feature in a future BeeGFS release?
- Managing BeeOND instances: clean cache after each job, start/stop services, …
  -> Developing scripts and integrating them in the ParaStation cluster management tools
- BeeGFS Client on Xeon Phi
  -> We need to provide access to the work file system also on the Booster nodes
Future challenges

- Improve performance of BeeGFS over EXTOLL
  -> Our colleagues in Fraunhofer already working in developing native EXTOLL support
- BeeOND on the NVMeS in the Booster
  -> Besides the client, other services need to be installed and configured on the Xeon Phi: beegfs-mgmtd, beegfs-meta, beegfs-storage, …
BACKUP
Interested?

Contact us!

DEEP
pmt@deep-project.eu

LinkedIn
http://linkd.in/1KiBe3y

DEEP-ER
pmt@deep-er.eu

Twitter
@DEEPprojects

The research leading to these results has received funding from the European Community’s Seventh Framework Programme (FP7/2007-2013) under Grant Agreement n° 287530 and n° 610476

BeeGFS User Meeting – May 18-19, 2016
Alternative Booster implementation
- Interconnect EXTOLL ASIC “Tourmalet”
- 32 KNC-node system
- Implement 4×4×2 topology, with Z dimension open

Experiment 2-phase immersion cooling
- NOVEC liquid from 3M
- Evaporates at about 50 degrees
- Condensates again in a water cooling pipe
- Allows very high-density integration
## Interconnect

<table>
<thead>
<tr>
<th></th>
<th>EXTOLL</th>
<th>Intel True Scale</th>
<th>Mellanox IBAN</th>
<th>PLX Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Availability</strong></td>
<td>Q3/2015</td>
<td>Now</td>
<td>2015</td>
<td>Now</td>
</tr>
<tr>
<td><strong>Switches</strong></td>
<td>None</td>
<td>IBAN</td>
<td>IBAN</td>
<td>IBAN</td>
</tr>
<tr>
<td><strong>Topologies</strong></td>
<td>≤7 direct connections</td>
<td>Switched, any, 1 rail</td>
<td>Switched, any, 1-2 rails</td>
<td>Switched, any, 1-2 rails</td>
</tr>
<tr>
<td><strong># Links per NIC</strong></td>
<td>7</td>
<td>1 or 2</td>
<td>1 or 2</td>
<td>1 or 2</td>
</tr>
<tr>
<td><strong>Link BW</strong></td>
<td>120 Gbit/s</td>
<td>40 Gbit/s</td>
<td>80 Gbit/s</td>
<td>103 Gbit/s</td>
</tr>
<tr>
<td><strong># contexts</strong></td>
<td>256</td>
<td>64</td>
<td>2*64</td>
<td>64</td>
</tr>
<tr>
<td><strong>SR-IOV support</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Drivers &amp; Firmware</strong></td>
<td>Adaptable</td>
<td>Available</td>
<td>Available</td>
<td>Available, KNL?</td>
</tr>
<tr>
<td><strong>Driver I/F</strong></td>
<td>VELO, SMFU, OFED</td>
<td>OFED, PSM</td>
<td>OFED, OFED PSM</td>
<td>OFED</td>
</tr>
</tbody>
</table>
Main EXTOLL characteristics
- Direct network: no switches required
- Integrates network interface controller
- Supports 6+1 links
- Capable of tunneling PCIe (allows remote-booting KNC from the network)

Current version of EXTOLL ASIC
- 270 million transistors
- Link bandwidth: 100 G
- MPI latency: 850 ns
- MPI bandwidth: 8.5 GB/s
- Message rate: 70 million mgs/sec
- PCIe Gen3 x16