Experiences with FhGFS

Frank Schlünzen
DESY-IT

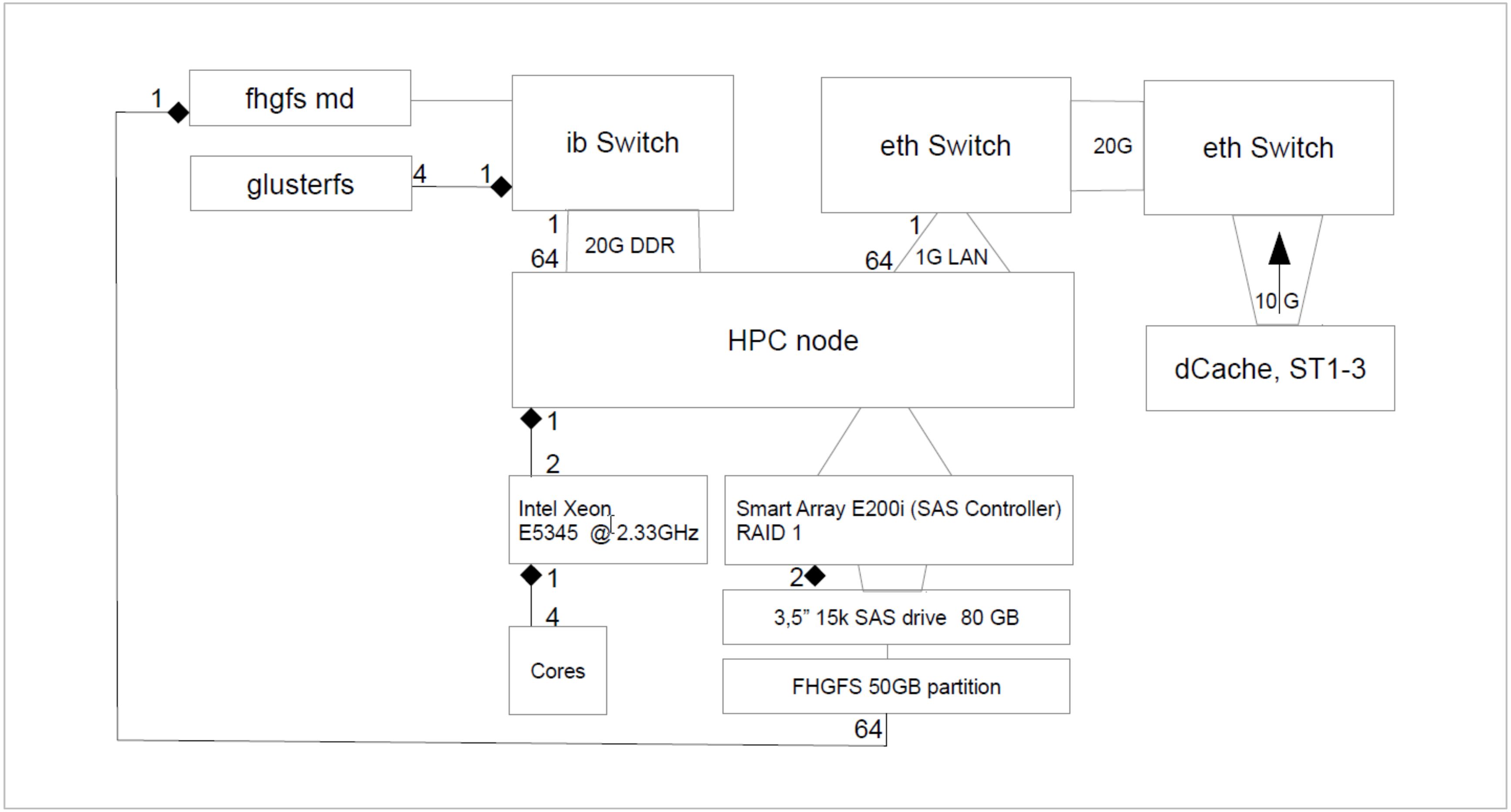
History

- Needed a reasonably fast network storage for batch farm
  - Got 4 storage boxes with a total of 72TB
  - Installed FhGFS as a testbed
  - Worked without any problems in a heterogeneous many-host environment
  - But only slow, high-latency network available.
- HPC cluster
  - Started to setup a HPC cluster (with old hardware) in 2011
  - Needed a fast filesystem for high-performance computing
  - FhGFS worked well in the batch farm, so why not on the HPC cluster
  - Problem: no dedicated storage for the old hardware, so had to be creative ….
- New HPC cluster
  - Meanwhile established a new HPC cluster, 1024 cores, 4TB ram, IB backbone
  - Use FhGFS as "work horse"
- Experimental setup
  - non-persistent storage setup

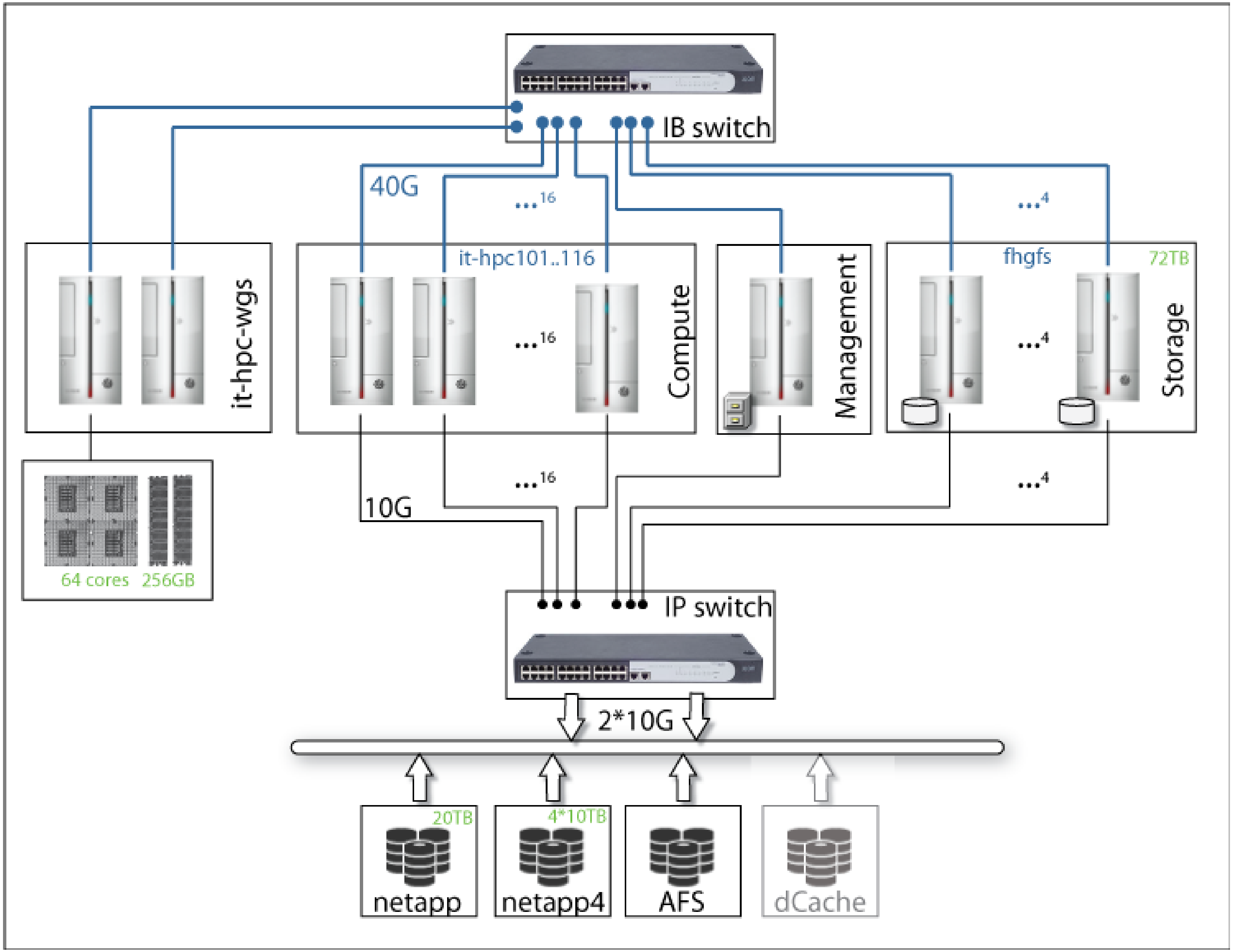
Experimental setup (3d old)

- 16 meanwhile ancient HPC nodes
  - 8 cores each, E5345 @ 2.33GHz, 16GB per core, 20G IB backbone, 1G eth
  - Created a 10GB ramdisk on each node
  - 1 node for FhGFS management (on ramdisk)
  - 1 node for FhGFS metadata (on ramdisk)
  - 14 storage nodes, each node also acting as a client mounting a 100GB file space
  - Entire system lives in memory (just a game)
- 1st time setup
  - Never set up a FhGFS system before
  - Just following the instructions on the wiki
  - Total time to success < 1h
  - Works like a charm.
  - Documentation and support are superb

Old HPC cluster



New HPC setup



Storage characterization - fhgfs

Name:	FHGFS 2011
Vendor:	Fraunhofer
Version:	2011.04.r21
Protocol:	fhgfs client/server
Storage Size:	3.2TByte / 94% free
# of head nodes:	**6**4
OS/Kernel:	SL 6.3 / 2.6.32-279.5.1.el6.x86_64
Disks per node:	2*80GB*0.5 SAS
Disk speed:	15k
Transfer speed:	6.00Gb/s
Raid Level:	1
Filesystem:	xfs
IRQ binding:	none
# Metadata server:	1
OS/Kernel:	SL 6.3 / 2.6.32-279.5.1.el6.x86_64
Disks per node:	2*80GB SAS
Disk speed:	15k
Transfer speed:	6.00Gb/s
Raid Level:	1
Filesystem:	ext4
IRQ binding:	none
Interconnect:	Infiniband DDR 20Gb/s
PingPong:	max. 1.000MB/s

Name:	FHGFS 2012
Vendor:	Fraunhofer
Version:	2011.04.r21
Protocol:	fhgfs client/server
Storage Size:	73TByte / 99% free
# of head nodes:	4
OS/Kernel:	SL 6.3 / 2.6.32-279.5.1.el6.x86_64
Disks per node:	12*2TB SATA
Disk speed:	7.2k
Transfer speed:	3.00Gb/s
Raid Level:	5
Filesystem:	xfs
IRQ binding:	none
# Metadata server:	1
OS/Kernel:	SL 6.3 / 2.6.32-279.5.1.el6.x86_64
Disks per node:	1*600GB SSD
I/O speed:	270 MB/sec (*read*) and 220 MB/sec (*write*).
Raid Level:	5
Filesystem:	ext4
IRQ binding:	none
Interconnect:	Mellanox Infiniband QDR 40Gb/s
PingPong:	max. 2.200MB/s

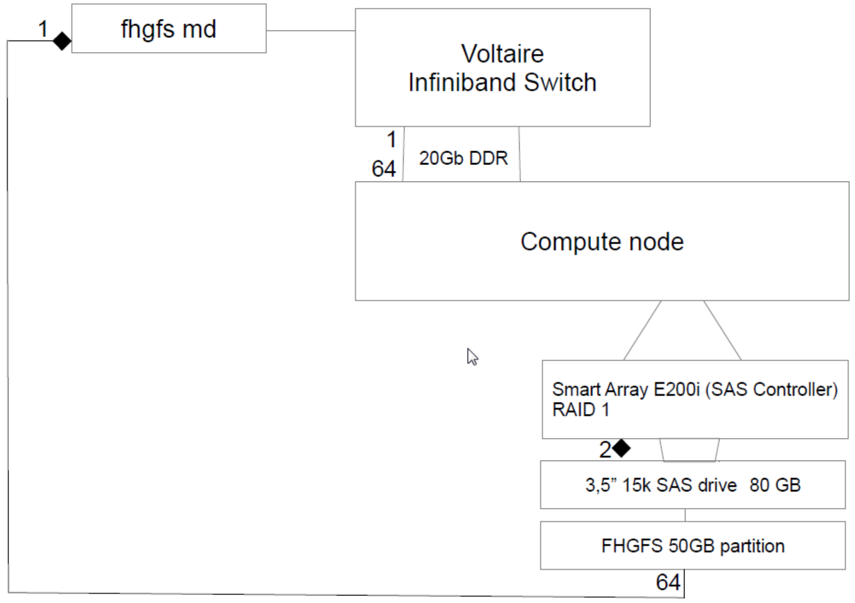
```
Client:  KB          reclen    write    rewrite      read    reread
         314572800     2048   822208          0   1233811         0
Server:  314572800     2048   919830          0    969687         0
```

Benchmarks – Pilatus 6M

- Pilatus 6M detector simulation
  - Typical example for PX detector
  - Can operate at ~25Hz (newer versions even at 100Hz)
  - Data format either raw (tiff) or compressed (cbf)
  - Data rates @20Hz: 1Gb/s for cbf, twice as much for tiff
- Execution: `pssh -t 0 -H “host1 host2” pilatus.sh`

	Description	Type	Capacity / TB	Protocol
1	fhgfs 2011	FHGFS (ipoib/rdma)	3.2	FHGFS
2	ST1	WAFL	20	NFS 3
3	ST2	WAFL	40	NFS 3
4	ST2	WAFL	4*10	NFS 4.1
5	ST3	GPFS	443	NFS 3
6	ST4	pnfs	10.000	NFS 4.1
7	fhgfs 2012	FHGFS (ipoib/rdma)	73	FHGFS
8	Glusterfs	Glusterfs (rdma)	73	3.2.6

Benchmarks – Pilatus 6M



fhgfs md

Voltaire
Infiniband Switch

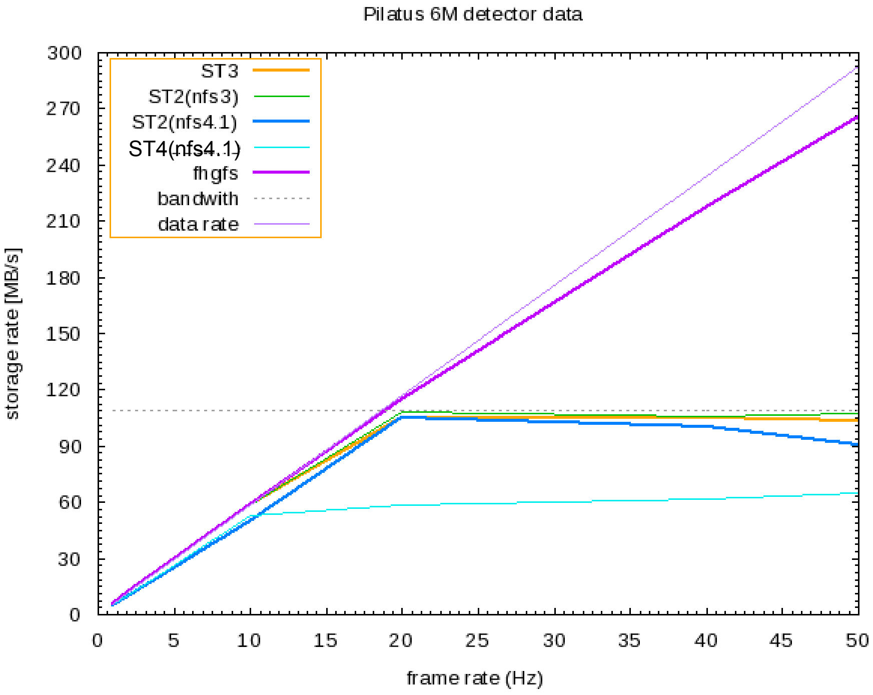
20Gb DDR

Compute node

Smart Array E200i (SAS Controller)
RAID 1

3,5” 15k SAS drive 80 GB

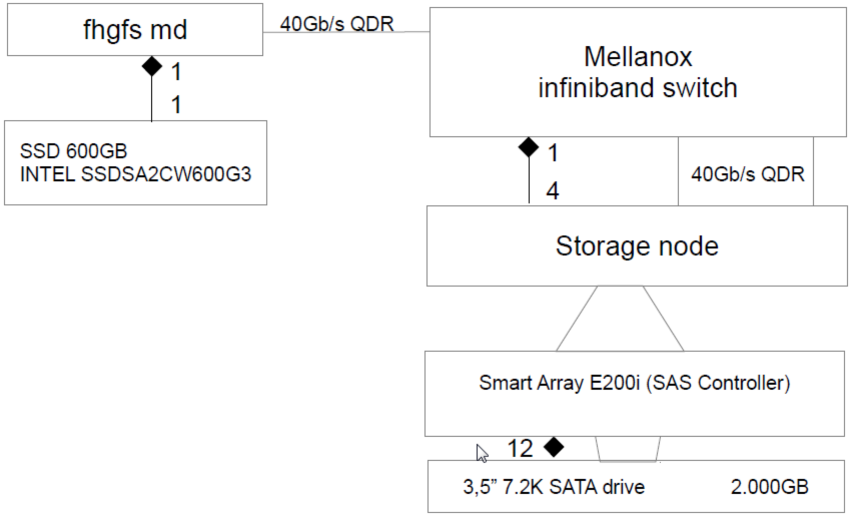
FHGFS 50GB partition

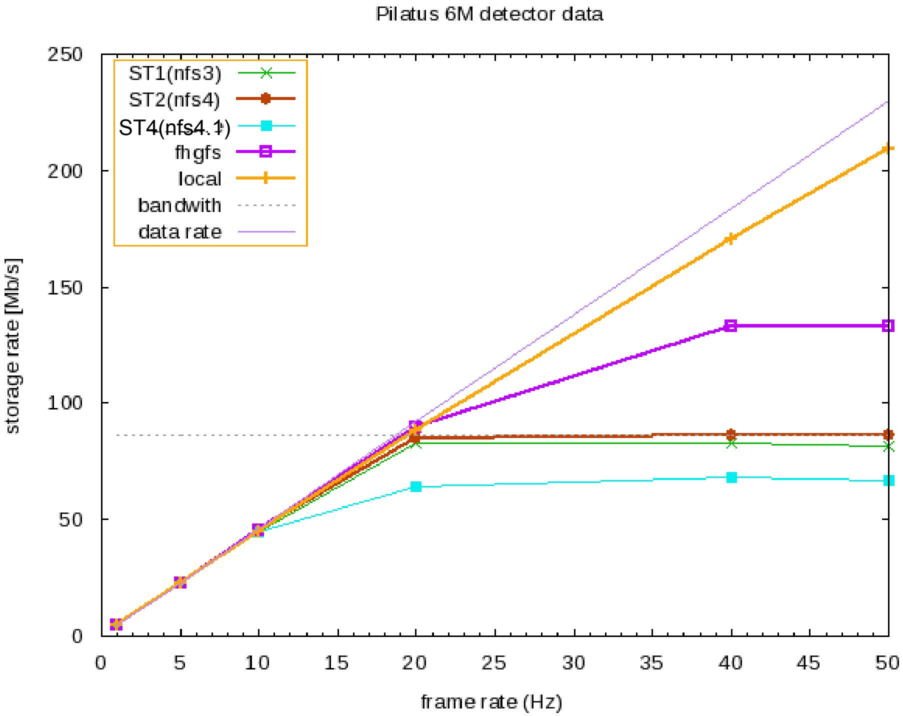


Single stream:

- 10Hz no problem
- 20Hz no problem
- 50Hz no problem for fhgfs

Benchmarks – Pilatus 6M

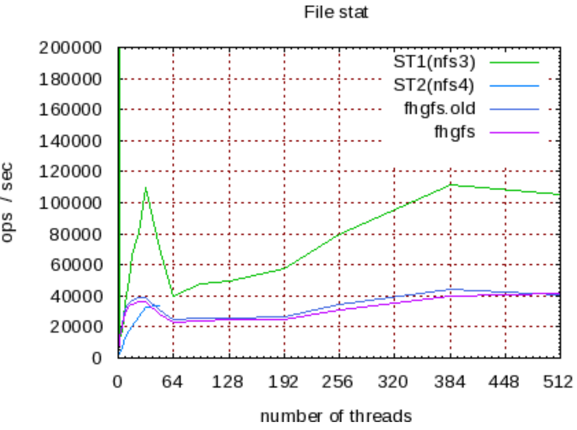


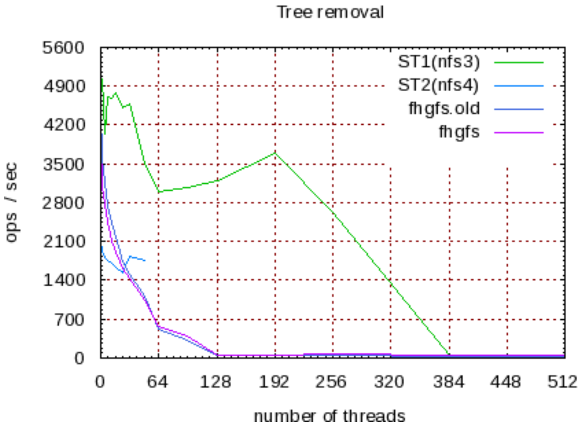


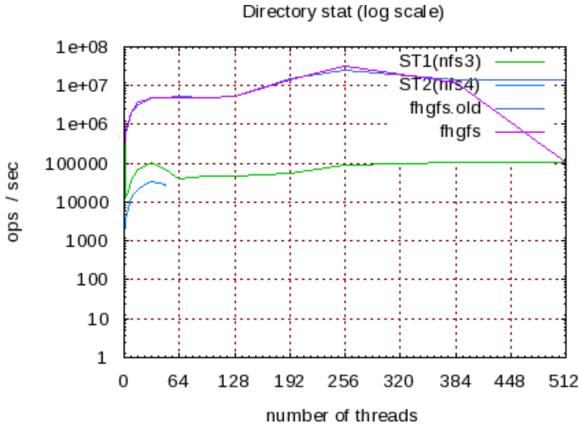
Single stream:

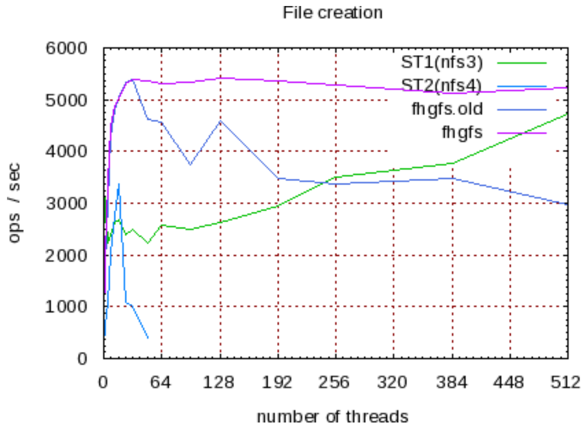
- 10Hz no problem
- 20Hz no major problem
- 50Hz might be a problem

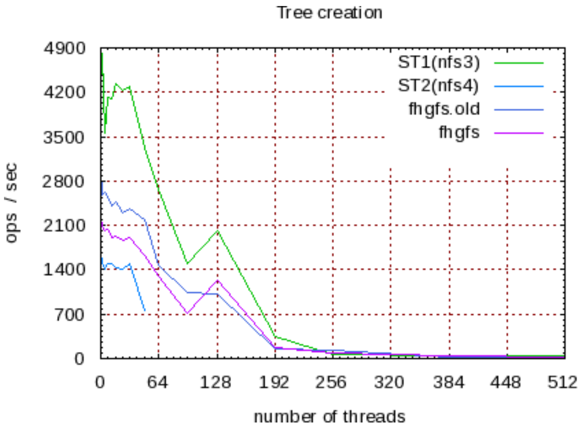
mdtests

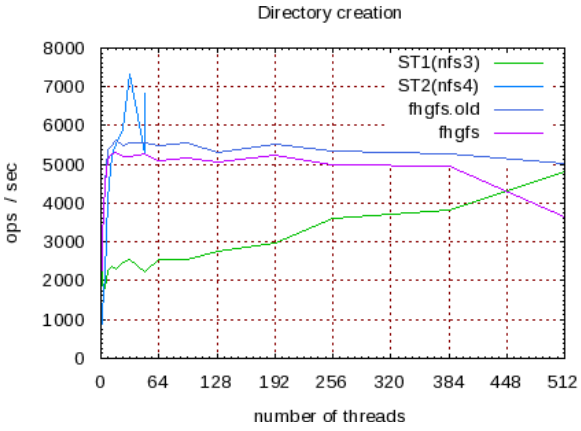

File stat
ops / sec
200000
180000
160000
140000
120000
100000
80000
60000
40000
20000
0
0
64
128
192
256
320
384
448
512
number of threads
ST1(nfs3)
ST2(nfs4)
fhgfs.old
fhgfs

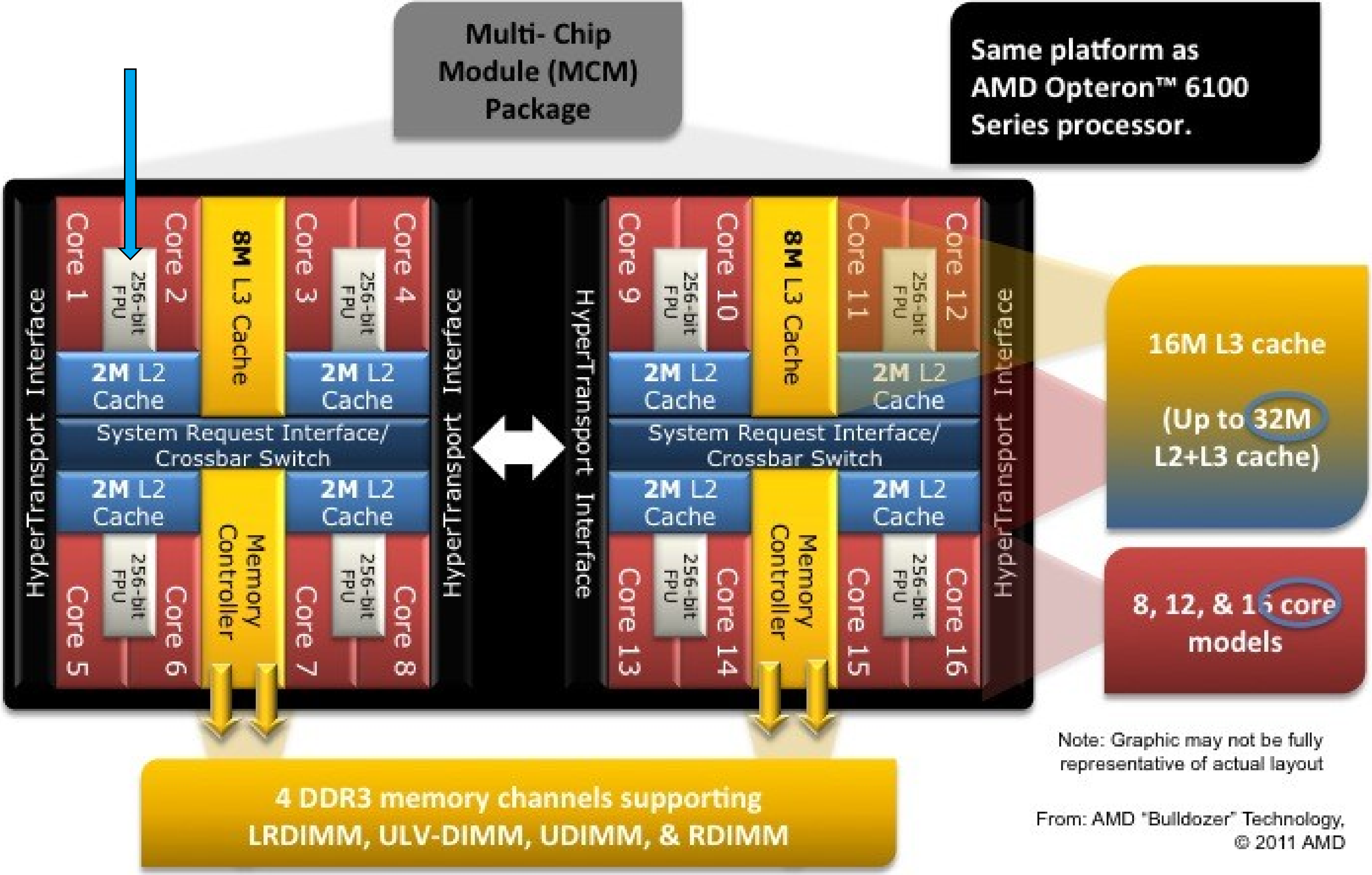

Tree removal
ops / sec
5600
4900
4200
3500
2800
2100
1400
700
0
0
64
128
192
256
320
384
448
512
number of threads
ST1(nfs3)
ST2(nfs4)
fhgfs.old
fhgfs


Directory stat (log scale)
ops / sec
1e+08
1e+07
1e+06
100000
10000
1000
100
10
1
0
64
128
192
256
320
384
448
512
number of threads
ST1(nfs3)
ST2(nfs4)
fhgfs.old
fhgfs


File creation
ops / sec
6000
5000
4000
3000
2000
1000
0
0
64
128
192
256
320
384
448
512
number of threads
ST1(nfs3)
ST2(nfs4)
fhgfs.old
fhgfs


Tree creation
ops / sec
4900
4200
3500
2800
2100
1400
700
0
0
64
128
192
256
320
384
448
512
number of threads
ST1(nfs3)
ST2(nfs4)
fhgfs.old
fhgfs


Directory creation
ops / sec
8000
7000
6000
5000
4000
3000
2000
1000
0
0
64
128
192
256
320
384
448
512
number of threads
ST1(nfs3)
ST2(nfs4)
fhgfs.old
fhgfs



Same platform as AMD Opteron™ 6100 Series processor.

16M L3 cache

(Up to 32M L2+L3 cache)

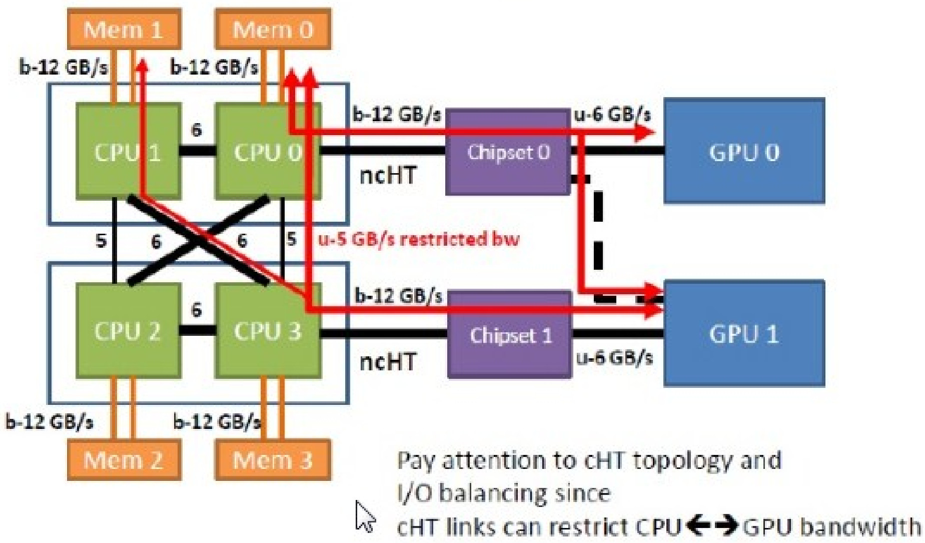
8, 12, & 16 core models

4 DDR3 memory channels supporting LRDIMM, ULV-DIMM, UDIMM, & RDIMM

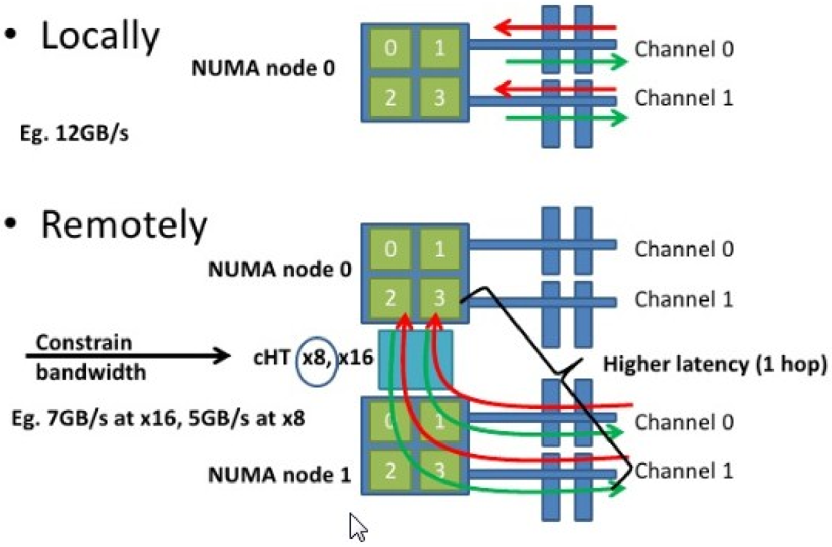
Note: Graphic may not be fully representative of actual layout

From: AMD "Bulldozer" Technology, © 2011 AMD

Numa bindings

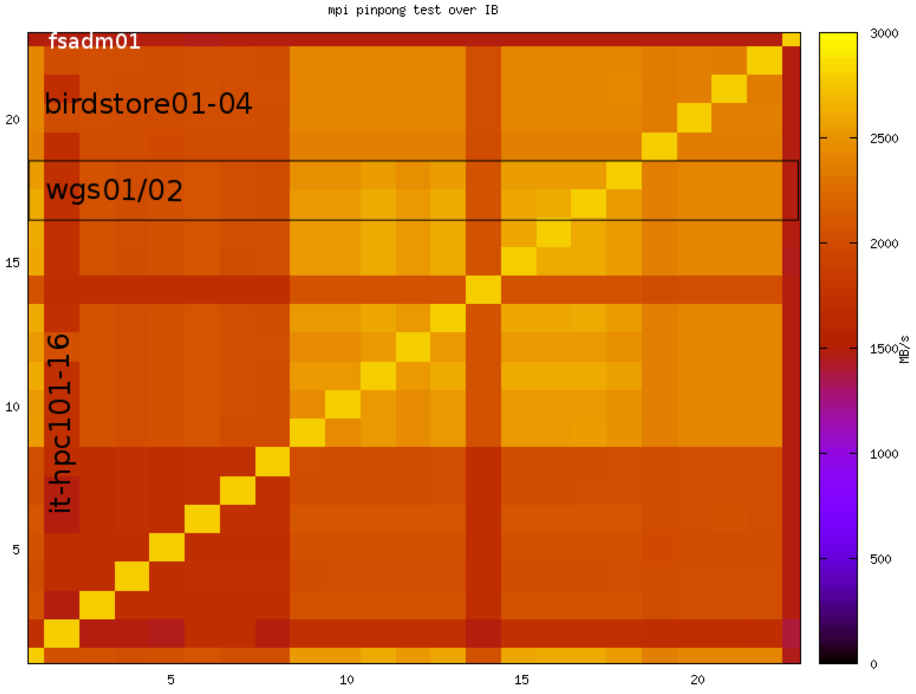


Pay attention to cHT topology and I/O balancing since cHT links can restrict CPU←→GPU bandwidth

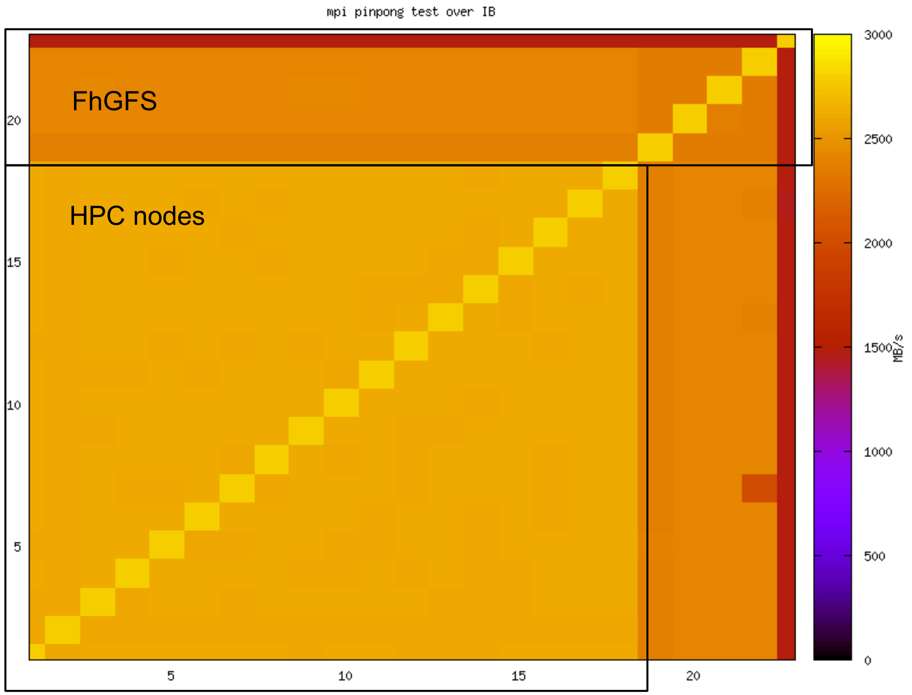


Make sure all IB traffic is bound to proper the numa node

Simple example - binding sockets and IB adapter



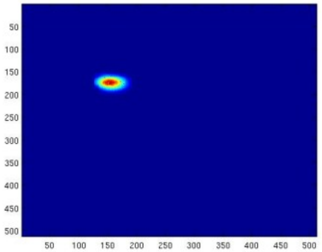
host-host ib bandwidth without socket binding

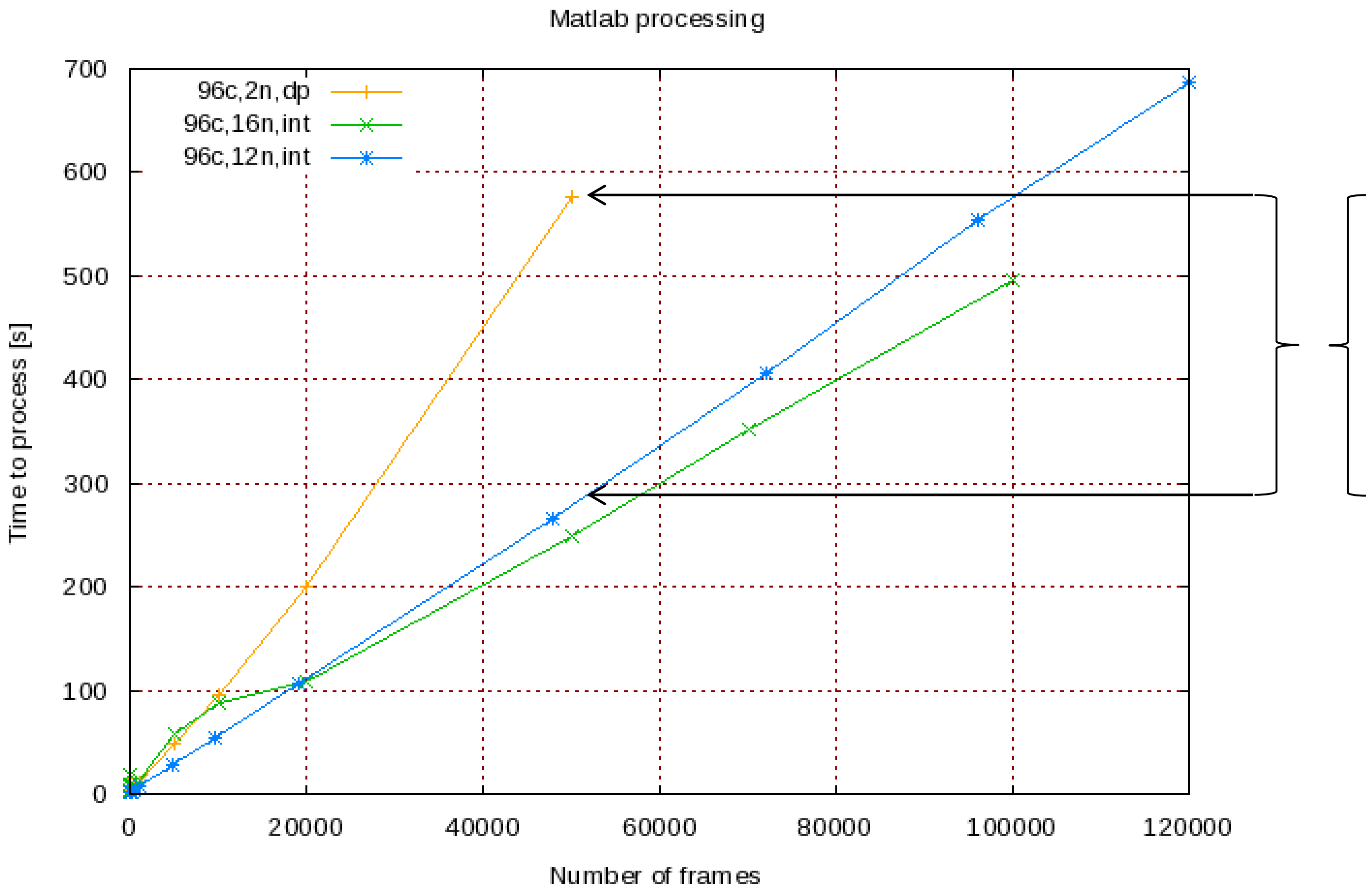


host-host ib bandwidth with socket binding

Matlab DCS Image processing

- Matlab processing 170.000 images a 512x512 px 16-bit
- Images contained in one HDF5 file
- Problem scales with #frames
- 96 concurrent threads without any problems



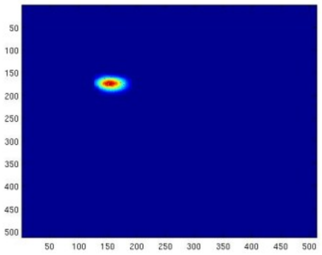


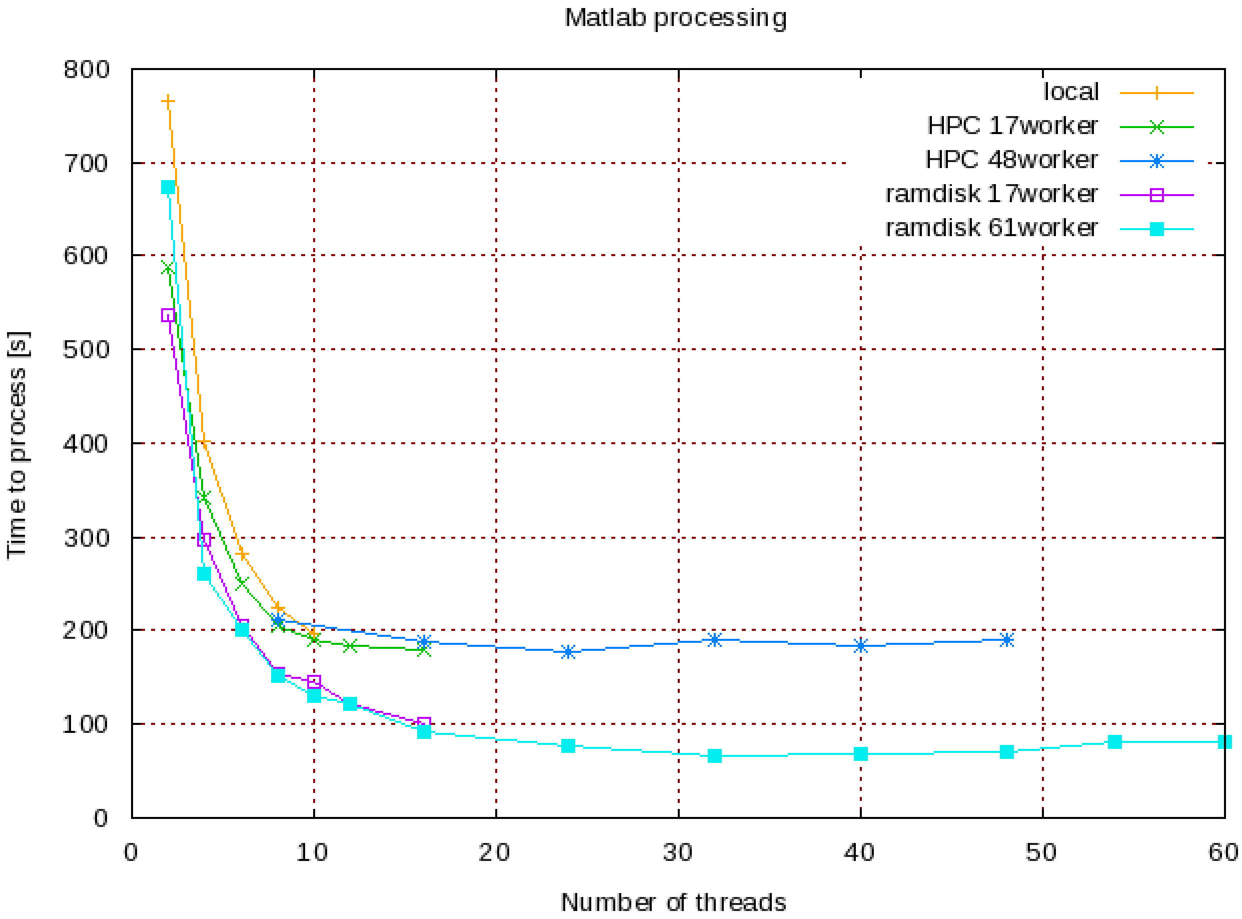
Floating point arrays

Integer arrays

Matlab DCS Image processing

- No benefit though ….
- … saturates at ~24 threads (worker)
- Same behavior for FhGFS and ramdisk
  - MP overhead? Affinity binding?





Summary

- Currently use FhGFS only as HPC scratch space
- No experience with performance for multiple mgt/meta servers
- Installation, maintenance, migration work very well
- Performance obviously depends on the number of heads, disk and controller speed
- Even with our limited setup shows very good performance
- Stability: no crashes or hick-ups at all
- No Windows client (and haven’t tried smb mounts yet)