More than just Performance

A scale out file system that solves I/O problems

HP-CAST 24   Frankhurt

11th July 2015   Dr. Franz-Josef Pfreundt

Who makes BeeGFS ?

Fraunhofer Spin Off

Sales & Support
Industry adaptations

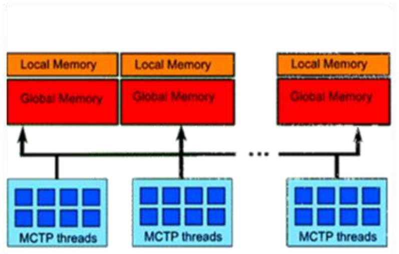
Fraunhofer
ITWM

Research & Development

Fraunhofer Center for High Performance Computing

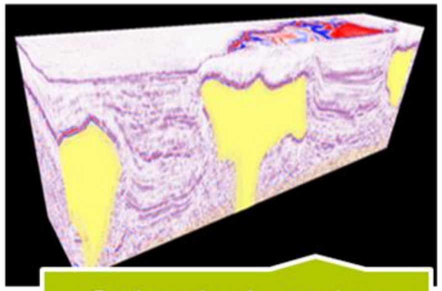
40 Scientists from Computer Science, Geophysics, Mathematics, Physics

BeeGFS®
developed by Fraunhofer



Parallel Programming GPI

Big Data GPI-Space



Seismic Imaging RTM – GRT – ISIM

thinkparQ

Fraunhofer ITWM

BeeGFS Design Philosophy

- ➢ Designed for Performance and Scalability
- ➢ Distributed Metadata
- ➢ No Linux patches, on top of EXT, XFS, ZFS, BTRFS, ..
- ➢ Scalable multithreaded architecture
- ➢ Native IB and Ethernet with dynamic failover (TCP, RDMA)
- ➢ Easy to install and maintain
- ➢ High Software Quality

→ Free software, 1000´s of users, 150 supported installations

BeeGFS Key Features

- Performance & Scalability
- Flexibility
  - Multiple daemons (any combination) can run on the same machine
  - Flexible striping per file/per directory
  - Add servers without downtime
  - On demand filesystem „per job“ possible
  - Client runs on any kernel >2.6.16
  - Client runs on Xeon PHI
  - ARM port available
  - NFS & SMB/CIFS re-export possible

BeeGFS Key Features

- Performance & scalability
- Flexibility
- Easy to use
  - Servers run in user space
  - No kernel patches
  - Servers use existing local filesystems (ext4, xfs, zfs, ...)
  - Packages for RHEL/SL/CentOS/SLES/Debian/Ubuntu
  - Hardware independent
  - Graphical monitoring tool

Flexibility

Bursting I/O

- Checkpointing
- Open Foam
- Shared File I/O
- HDF 5
- CFD output
- Ugly Life Sciences Codes
- ....

Wolfgang Nagel: Flex I/O System with different I/O targets
monitoring system steers I/O
DARPA: High Speed Burst Buffer, transparent caching

Complex Solutions - Let us make it easy

BeeGFS ON Demand

Fraunhofer Seislab

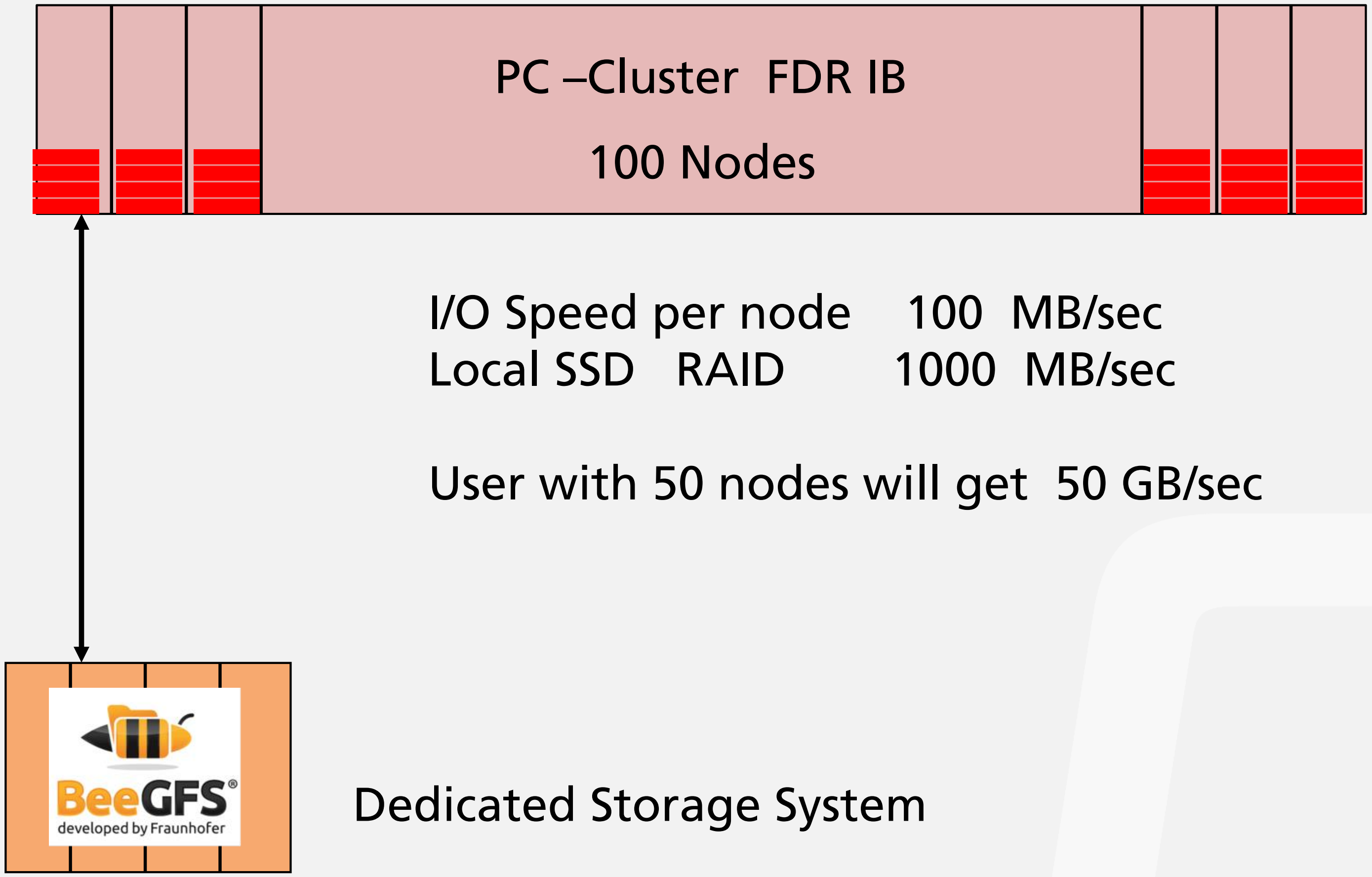
In-house cluster of CC-HPC at Fraunhofer ITWM

- 92 compute nodes with 1 TB of SSDs each
- Global BeeGFS storage on 3,5“ SATA drives
- 1500 Avoton nodes , single SSD per node

How does BeeOND work

- Create a temporary PFS per Job across SSD´s
- Done using PBS prolog
- Stage-in input data, work on BeeOND, stage-out results

Why ?



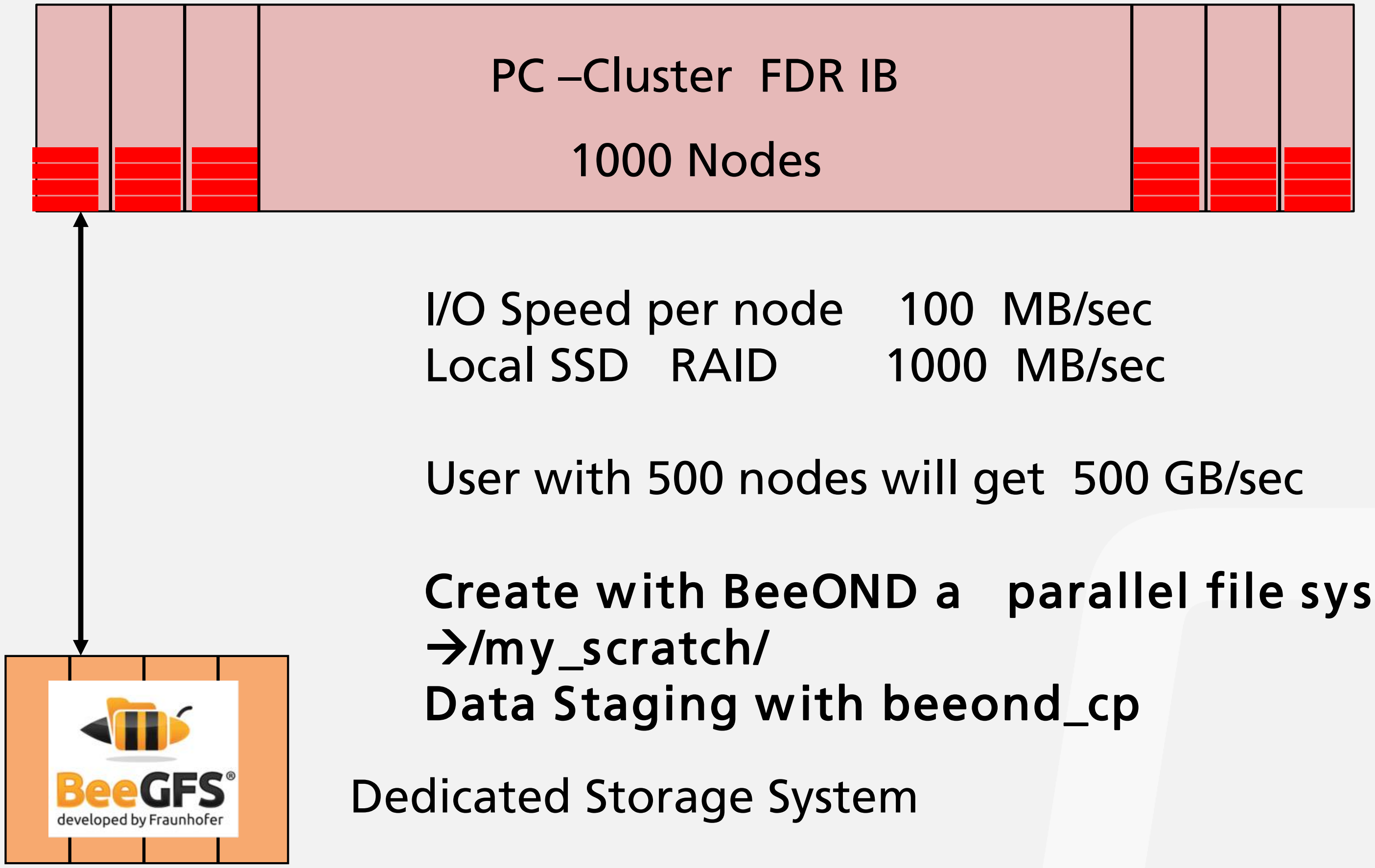
I/O Speed per node   100 MB/sec
Local SSD RAID   1000 MB/sec

User with 50 nodes will get 50 GB/sec

Dedicated Storage System

10 GB/sec streaming I/O

BeeOND



I/O Speed per node 100 MB/sec
Local SSD RAID 1000 MB/sec

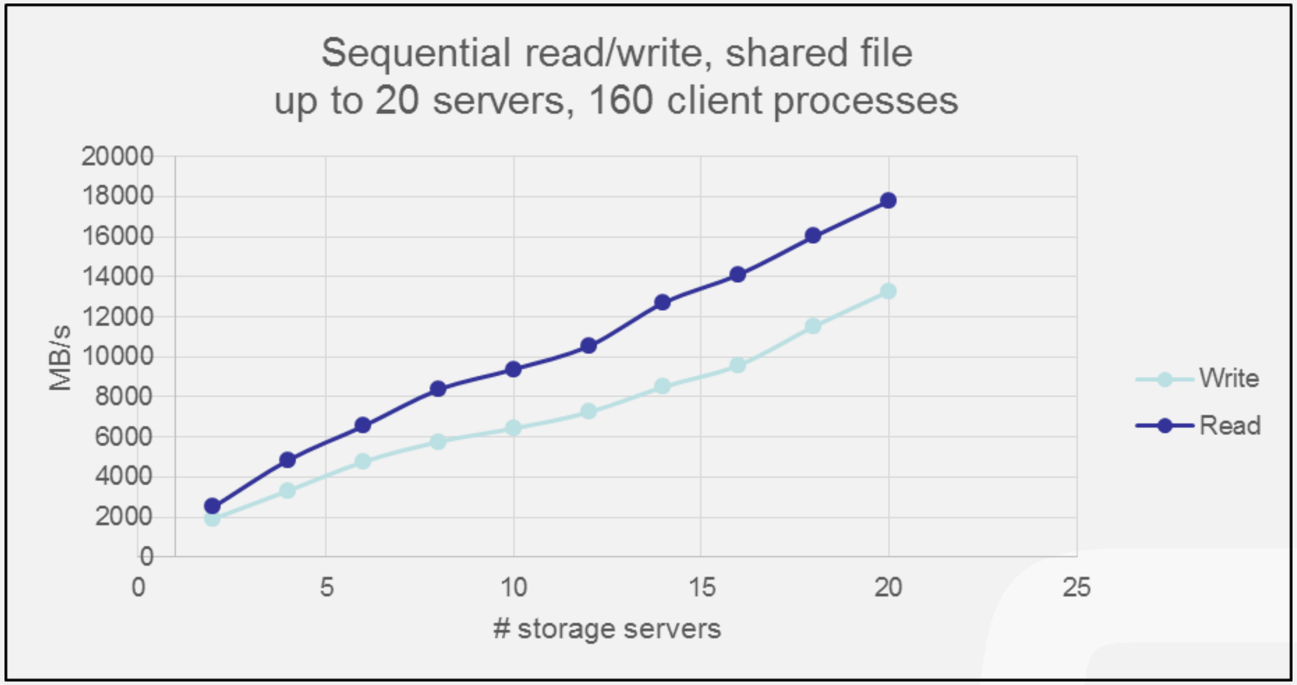
User with 500 nodes will get 500 GB/sec

Create with BeeOND a parallel file system (BeeGFS) per
→/my_scratch/
Data Staging with beeond_cp

Dedicated Storage System

100 GB/sec streaming I/O

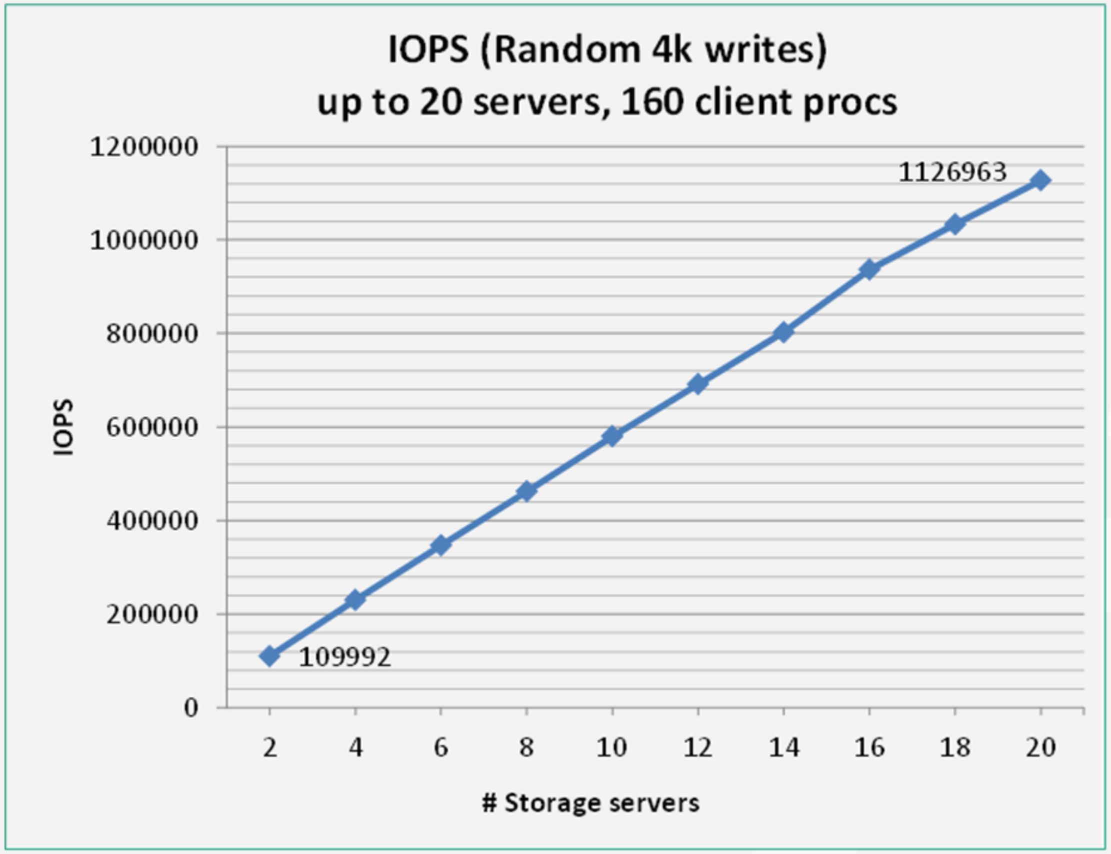
Shared File I/O with BeeGFS



Chunk size : 600 kB 20 server with 4x250GB SSD

Write : 52 % of maximal performance
Read : 67 % of maximal perforamnce

IOPS low latency I/O



Mesured on a 4 year old quad SSD system

Why does this work?

- ➢ We do not patch the kernel or a underlying file system!
- ➢ All server software is user space -> up and running in a second
- ➢ Runs on any underlying file system (ZFS, EXT, XFS, BTTRFS,..)
- ➢ BeeGFS server software is multithreaded and performance optimized
  - ➢ **Low CPU overhead**
- ➢ BeeGFS scales to high server numbers : Data & Metadata

→client & server & compute on the same node

BeeOND® simply works

Data Availability

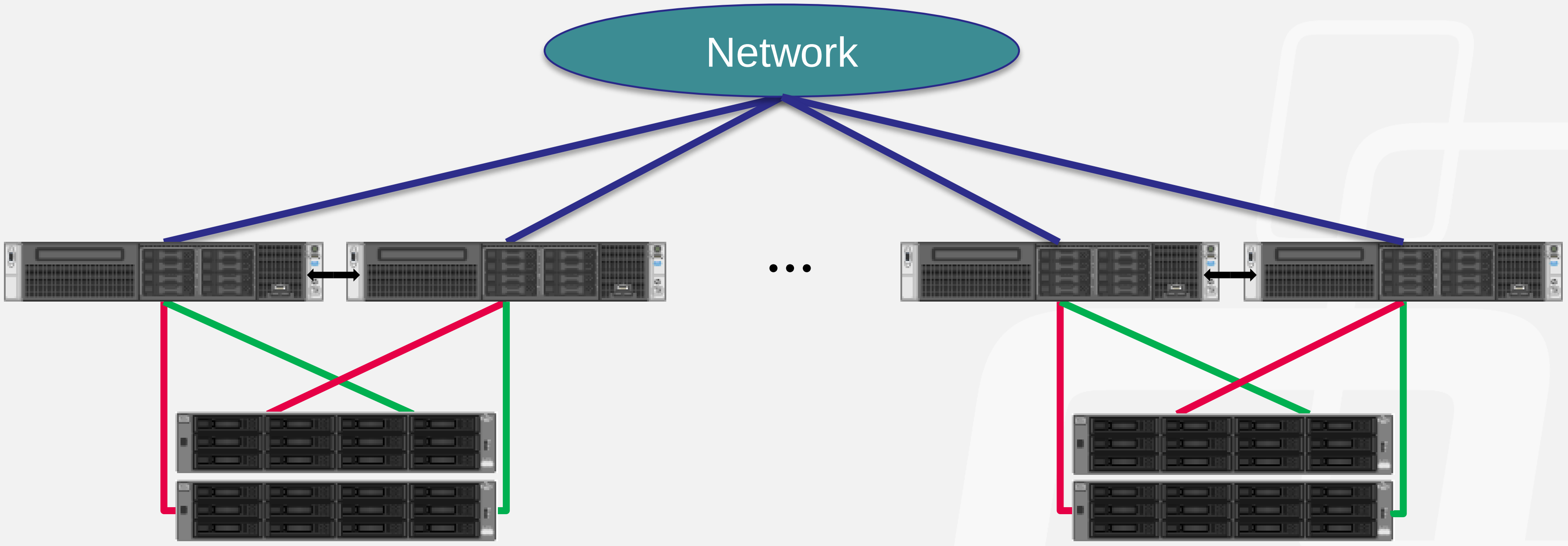
Data Availabilty Aspects of BeeGFS

- All server software is user space
    - -> Software problem would not crash a server
- Dynamic network fail over and fail back
- Uses the availability features of the underlying file systems

But

- Server Hardware breaks
- Raid Systems die
- In compute & store solutions applications may crash the system

High availability – failover with shared storage



High availability – failover with shared storage

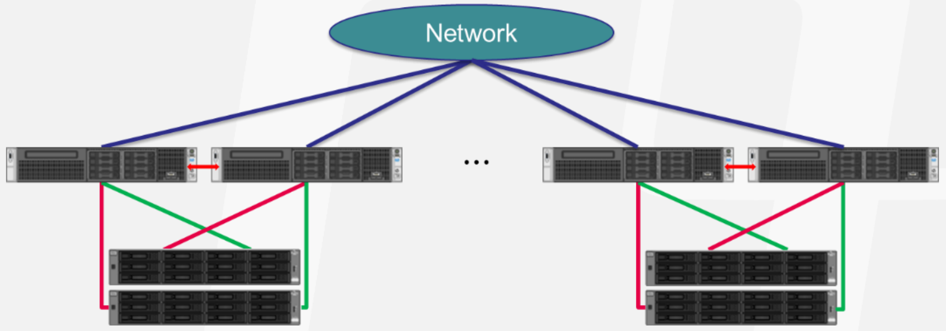
Pros:

- No system downtime in case of server failure
- No additional storage capacity needed

Cons:

- Expensive storage components needed
- 3rd party software components needed
- Complex to set up and maintain
- Failover Risk
- Does not cover RAID

  System failures

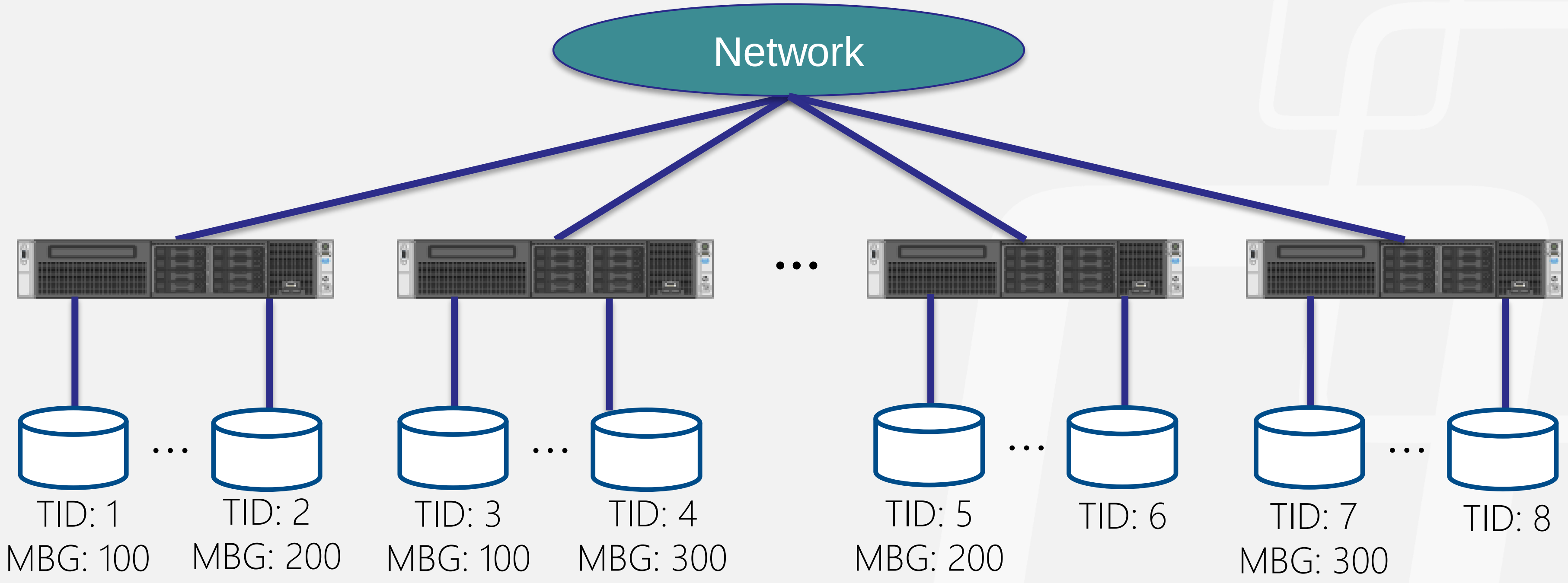


We introduce at ISC 2015

BeeGFS Enterprise Edition

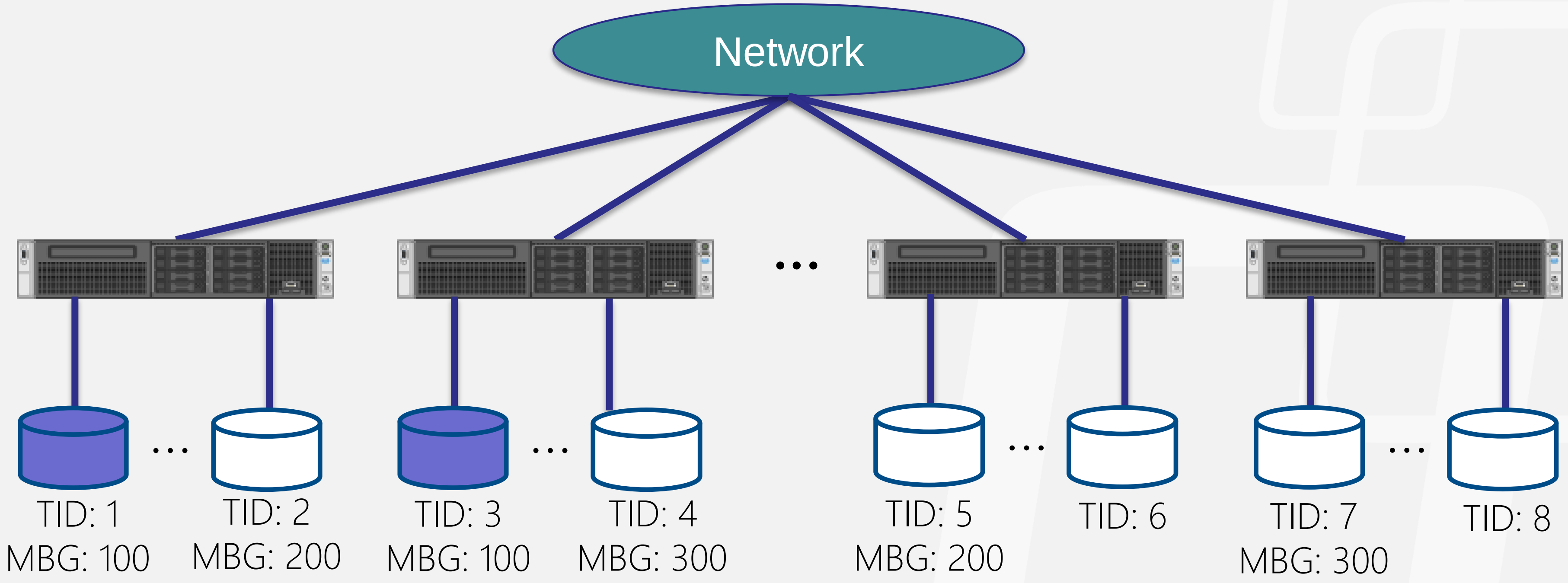
BeeGFS with Built-in replication

- Assign targets to „mirror buddy groups"
- MBGs replicate chunks (but can also store non-replicated data)
- Internal HA/failover and restore mechanisms



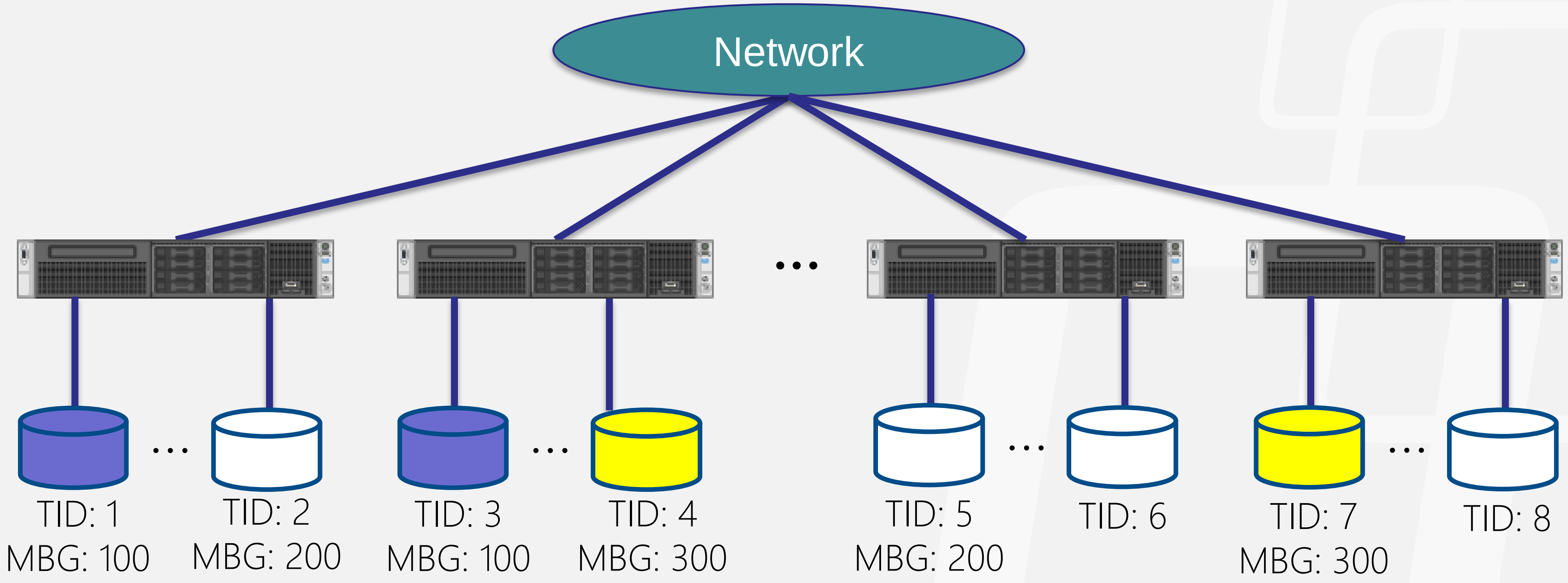
BeeGFS with Built-in replication

- Assign targets to „mirror buddy groups“
- MBGs replicate chunks (but can also store non-replicated data)
- Internal HA/failover and restore mechanisms

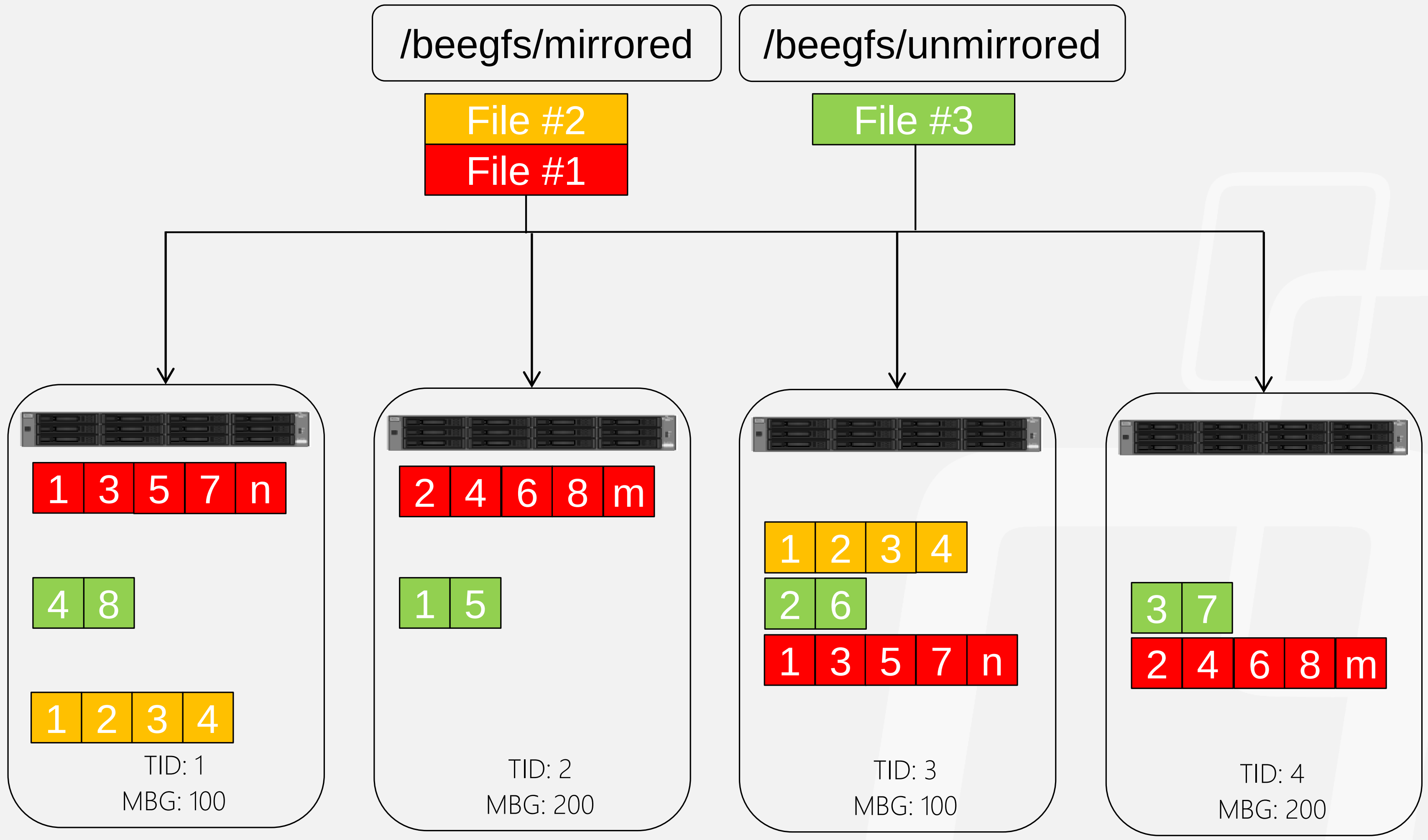


BeeGFS with Built-in replication

- Assign targets to „mirror buddy groups“
- MBGs replicate chunks (but can also store non-replicated data)
- Internal HA/failover and restore mechanisms

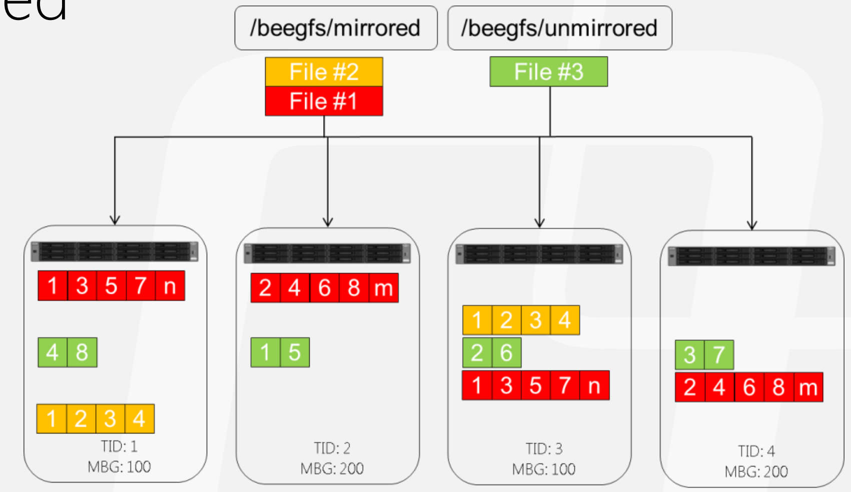


High availability – Built-in replication



High availability – Built-in replication

- Flexible (replication configurable per-directory)
- Easy to scale/extend
- No 3rd party tools for monitoring and failover functionality
- Covers server and raid failures
- Any storage backend can be used
- Additional data safety



HA at lower cost

BeeGFS EP Edition

- Built-in High Availability
- Quota
- ACL

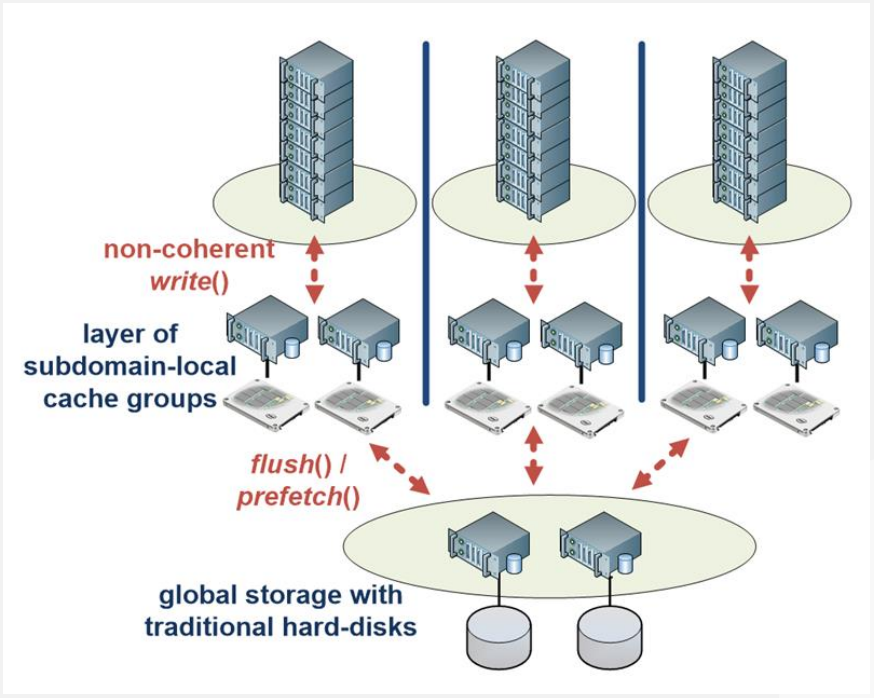
BeeGFS Hadoop Adaptor

Roadmap:

- HA for OSTs within the next weeks
- in Q4 Built-in HA for MDS

Research Topics - BeeGFS API

Create non coherency zones using BeeOND and add a data cache



BeeGFS cache layer managed by an API (for the very few exascale apps)
BeeGFS cache API will be available in 2016
-> allows application-controlled data movements to/from global storage

Research Topics - Erasure Coding

- Add k „parity“ blocks to n stripe sets
- Reduces the amount of additional storage and covers k server failures
- K=2, n=4, 50 % more storage

- Basic library is implemented - patent free
- Performance tuning

Possible future feature?

Looking for Development Sponors

BeeGFS®
developed by Fraunhofer

BeeOND®

A parallel file system that solves I/O problems

Questions?