An Introduction to BeeGFS

Solid, fast, flexible – and easy!

www.beegfs.com

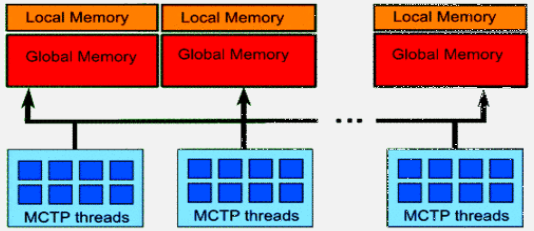
An Introduction to BeeGFS

- Introduction
- BeeGFS Architecture
- BeeOND – BeeGFS on Demand
- Buddy Mirroring
- Tools & API
- BeeGFS News
- BeeGFS in use
- Conclusion

Fraunhofer

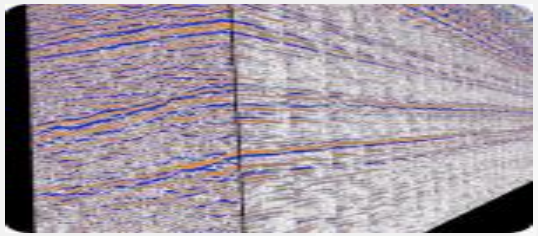
- BeeGFS was originally developed at the Fraunhofer Center for HPC
- The Fraunhofer Gesellschaft (FhG)
  - **Largest organization for applied research in Europe**
    - Special base funding by German government
  - Institutes, research units and offices around the globe
  - Staff: ~24000 employees

The Fraunhofer Center for HPC



Parallel Programming models & tools

Photo realistic real time ray tracing



Interactive seismic imaging

Parallel File System

Big Data

Smart Energy / Green by IT

thinkparQ

- A Fraunhofer spin-off
- Founded in 2014 specifically for BeeGFS
- Based in Kaiserslautern (right next to Fraunhofer HPC Center)

- **Consulting, professional services & support for BeeGFS**
- Cooperative development together with Fraunhofer
  (Fraunhofer will continue to maintain a core BeeGFS HPC team)

- First point of contact for BeeGFS

BeeGFS Architecture

History

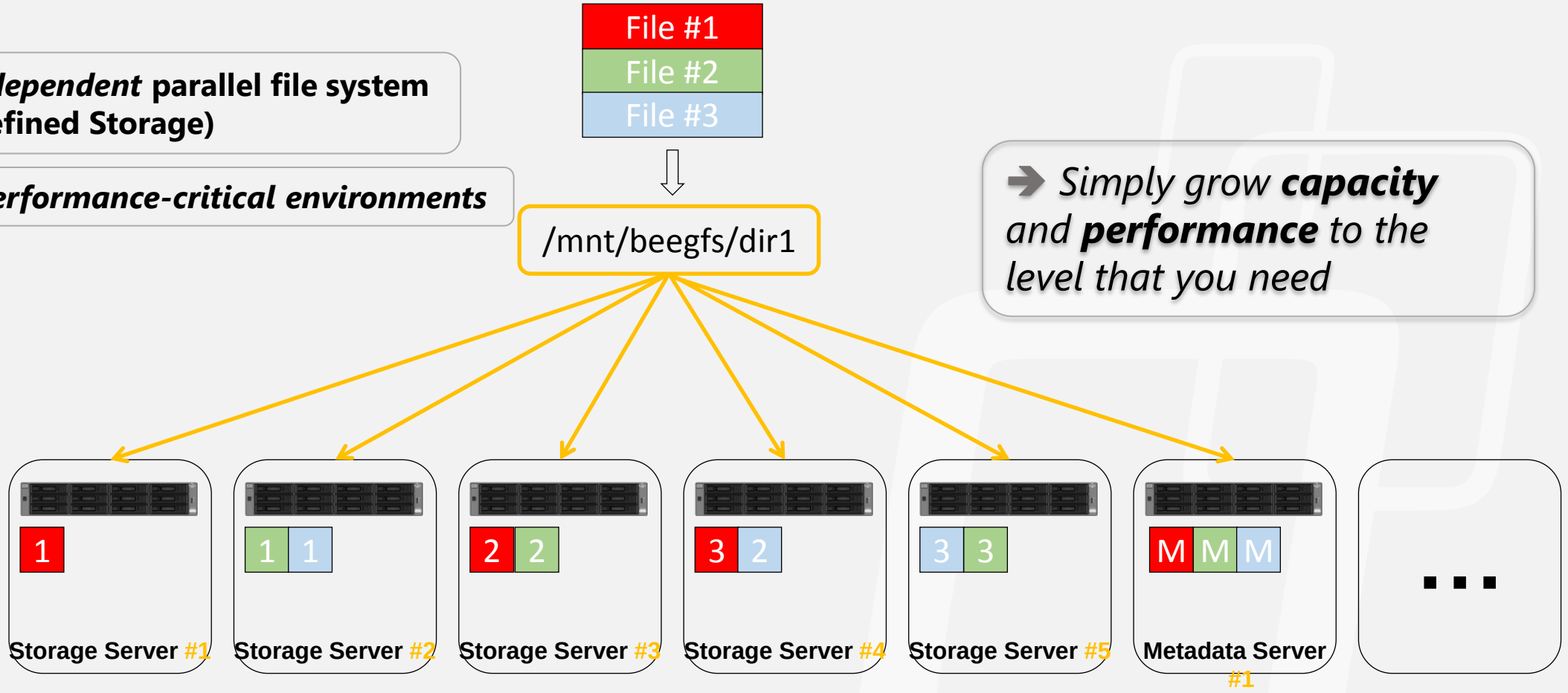
- Development started in 2005
- Why?
  - Evaluated existing solutions, not happy with what we found:
    - Very complex and limited flexibility
    - Required dedicated staff for continuous maintenance
    - Expensive
    - Scalability and performance problems for metadata access, shared file writes, single-stream I/O, …
  - We're a HPC center, so a lot of knowledge and users in-house

What is BeeGFS?

BeeGFS is…

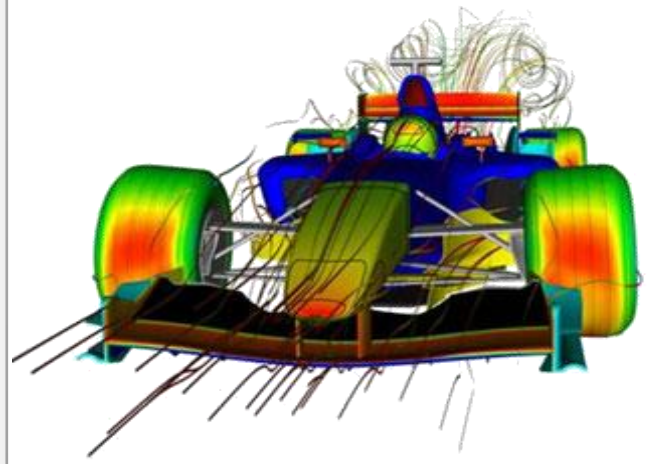
…a *hardware-independent* parallel file system (aka Software-defined Storage)

…designed for *performance-critical environments*



➔ *Simply grow **capacity** and **performance** to the level that you need*

Key Aspects



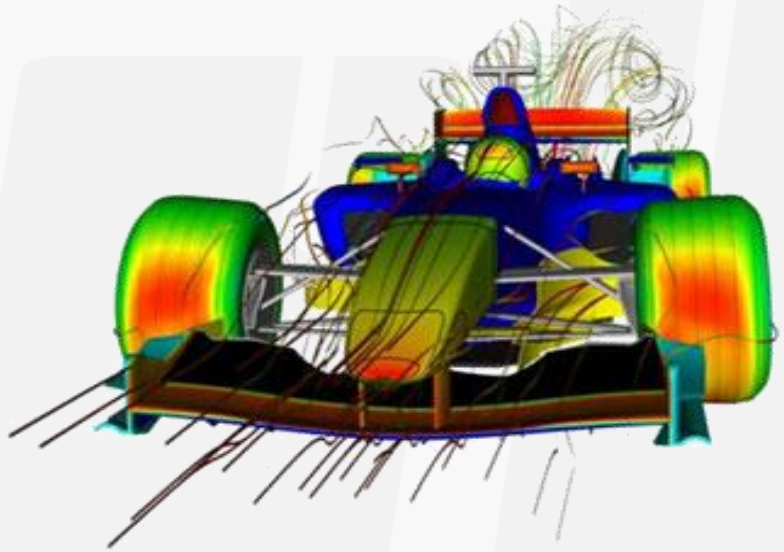
Maximum Performance & Scalability

High Flexibility

Robust & Easy to use

Key Aspects

- **Performance & Scalability**
  - Initially optimized for performance-critical workloads
  - Efficiently multi-threaded and light-weight design
    - *"Not even breaking a sweat: BeeGFS at 10GB/s on single node all-flash unit over 100Gbit network"* – ScalableInformatics
  - Supports RDMA/RoCE and TCP (Infiniband, Omni-Path, 100/40/10/1GbE, …)
  - Distributed file contents & distributed metadata
    - Aggregated IOPS and throughput of multiple servers
    - Scales to millions of metadata operations *per second*
  - High single stream performance
    - 9GB/s single-stream throughput with Mellanox EDR (Few file streams completely saturate a 100GBit link.)



Key Aspects

✓Performance & Scalability

- **Flexibility**
  - Multiple BeeGFS services (any combination) can run together on the same machine: Converged system
  - Flexible striping per-file / per-directory
  - Add servers at runtime
  - On-the-fly creation of file system instances (BeeOND)
  - Installation & updates without even rebooting
  - NFS & Samba re-export possible

  - Runs on different Architectures, e.g.
    - …

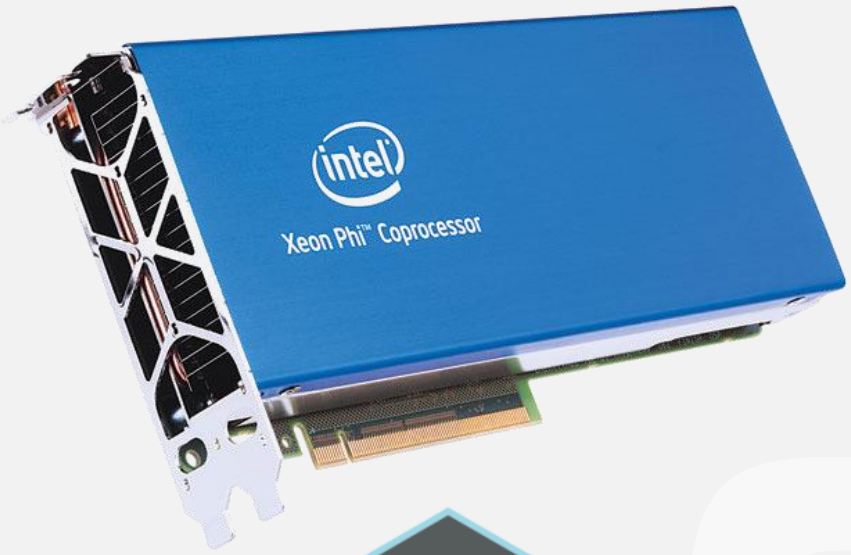
Flexibility: CPU Architectures

OpenPOWER IBM

BeeGFS®

BeeGFS is a proud new Member of the OpenPower Ecosystem thanks to Support from IBM

thinkparQ



Key Aspects

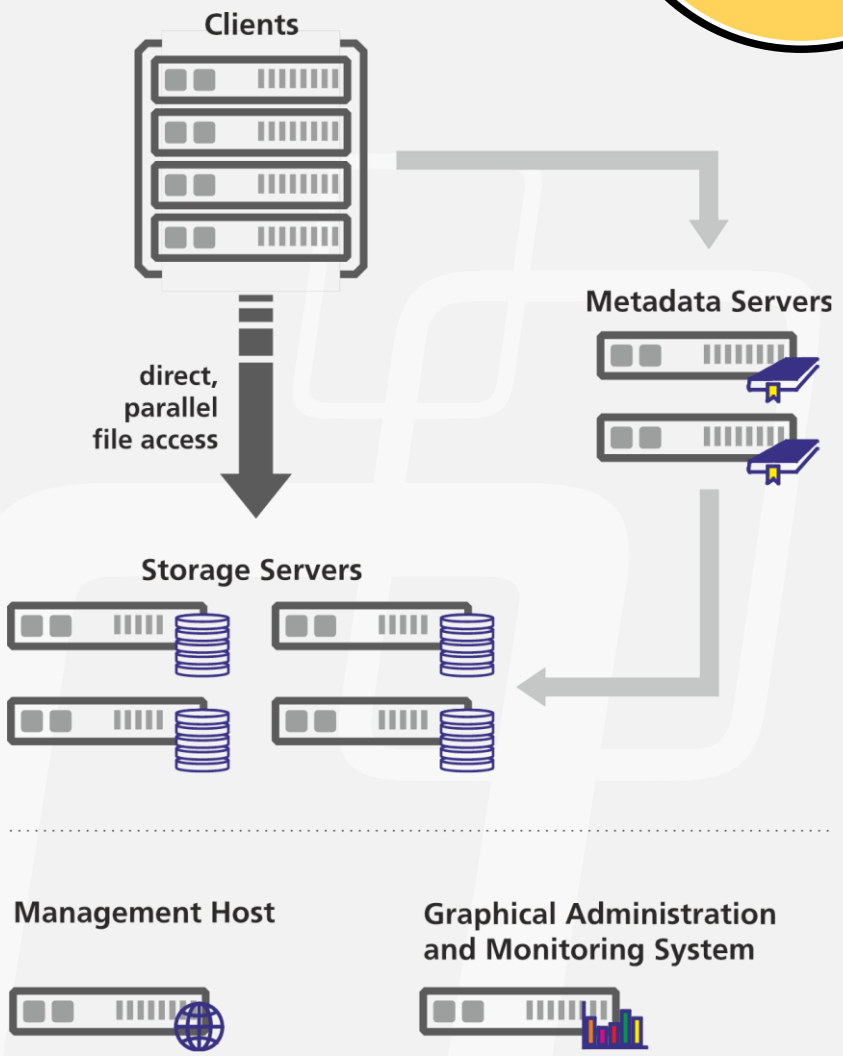
✓Performance & Scalability

✓Flexibility

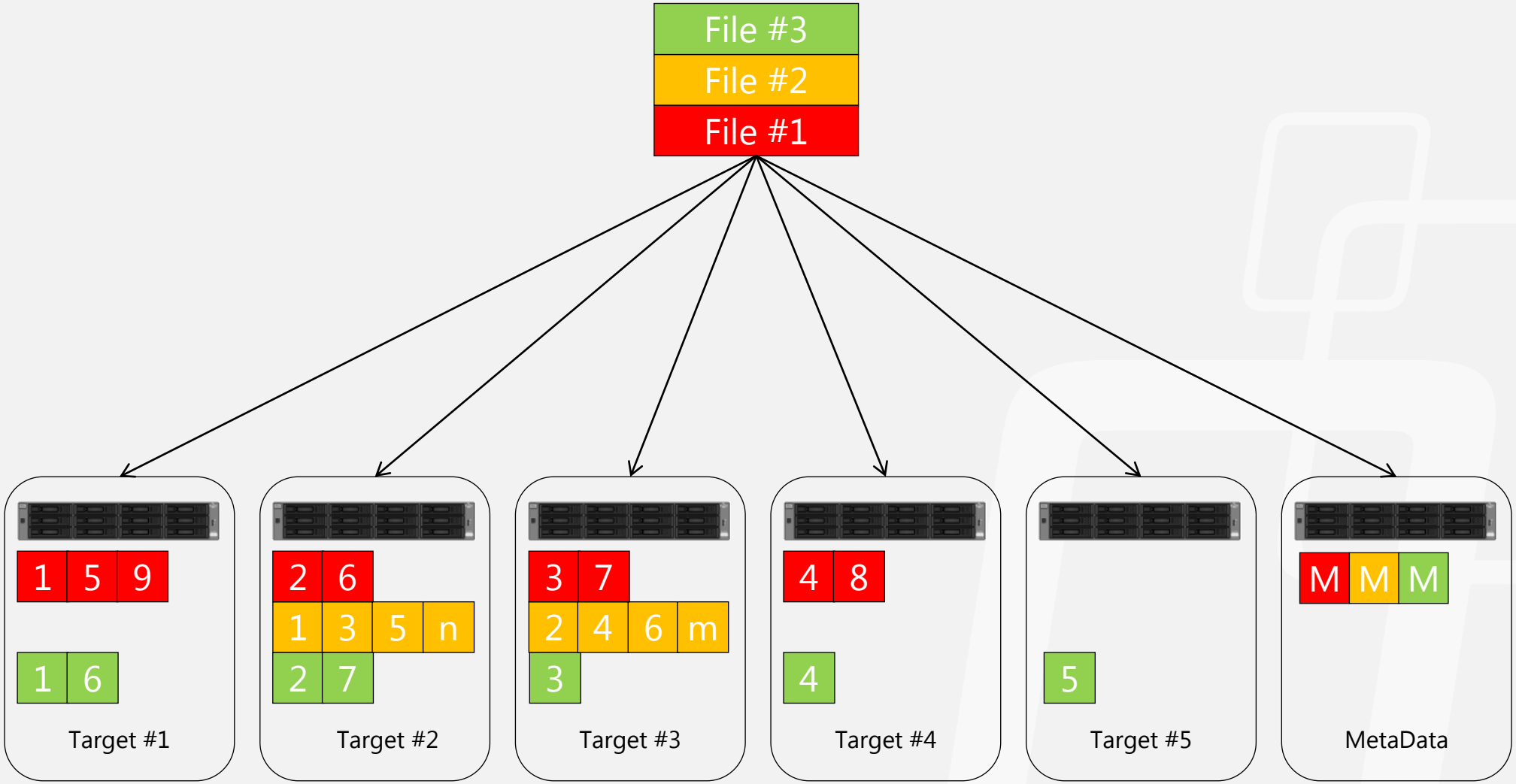
- **Robust & Easy to use**
  - Very intensive suite of release stress tests, in-house production use before public release
    - The move from a 256 nodes system to a 1000 nodes system did not result in a single hitch, similar for the move to a 2000 nodes system.
  - Applications access BeeGFS as a normal (very fast) file system mountpoint
  - Servers run on top of standard local filesystems (ext4, xfs, zfs, …)
  - No kernel patches
    - Updates of system packages, kernel and BeeGFS are trivially simple
  - Packages for Redhat, SuSE, Debian and derivatives
  - No special hardware requirements
  - Graphical monitoring tool

BeeGFS Architecture

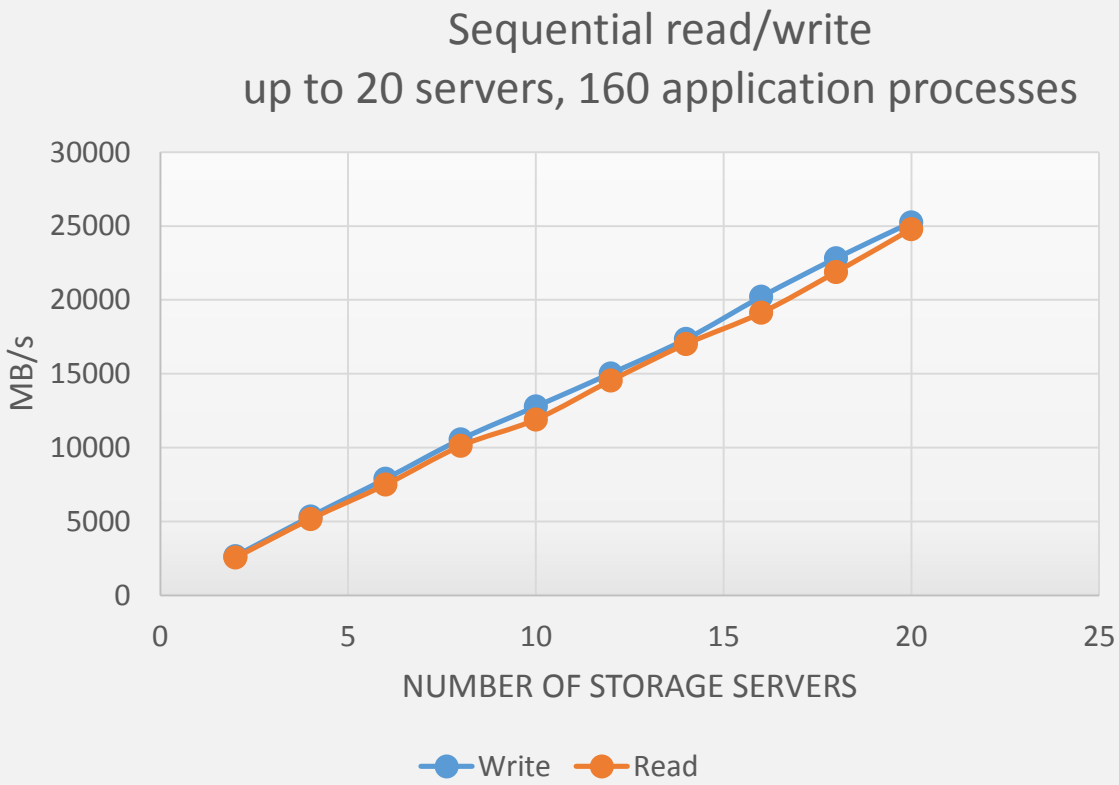
- Client
  - Native Linux module to mount the file system
- Storage Service
  - Store the (distributed) file contents
- Metadata Service
  - Maintain striping information for files
  - Not involved in data access between file open/close
- Management Service
  - Service registry and watch dog
- Graphical Administration and Monitoring System
  - GUI to perform administrative tasks and monitor system information
    - Can be used for “Windows-style installation“

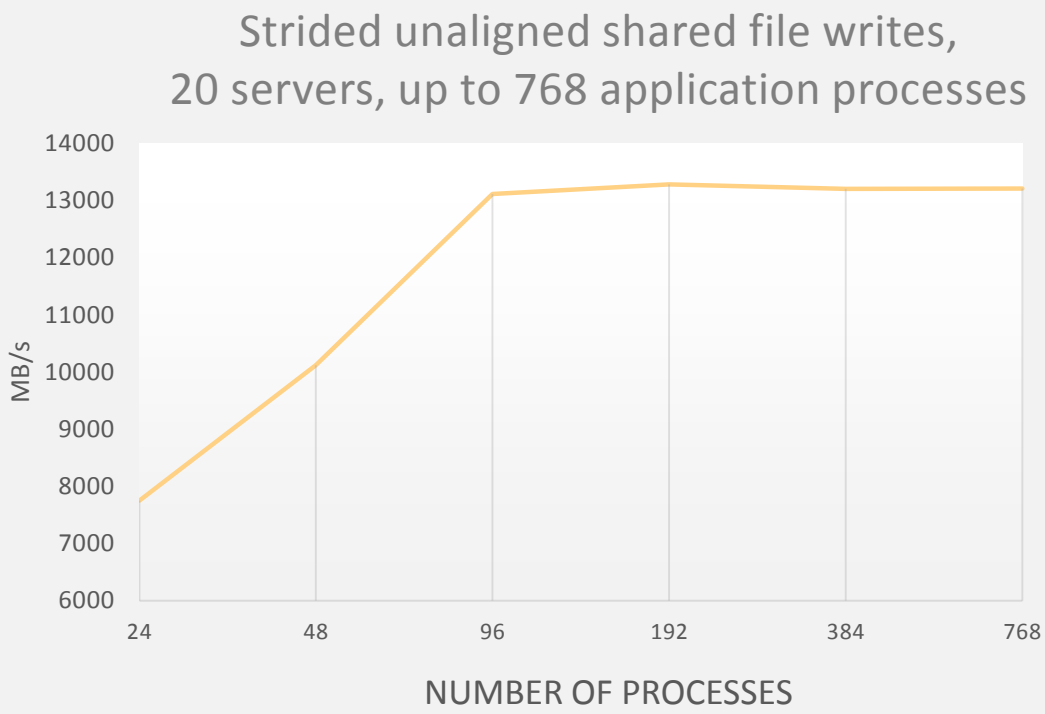


BeeGFS Striping



Throughput Scalability

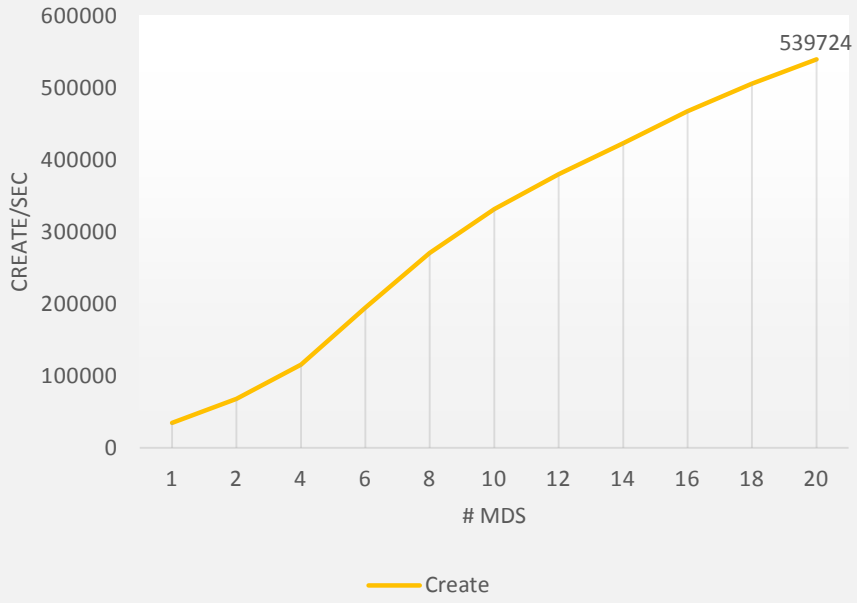




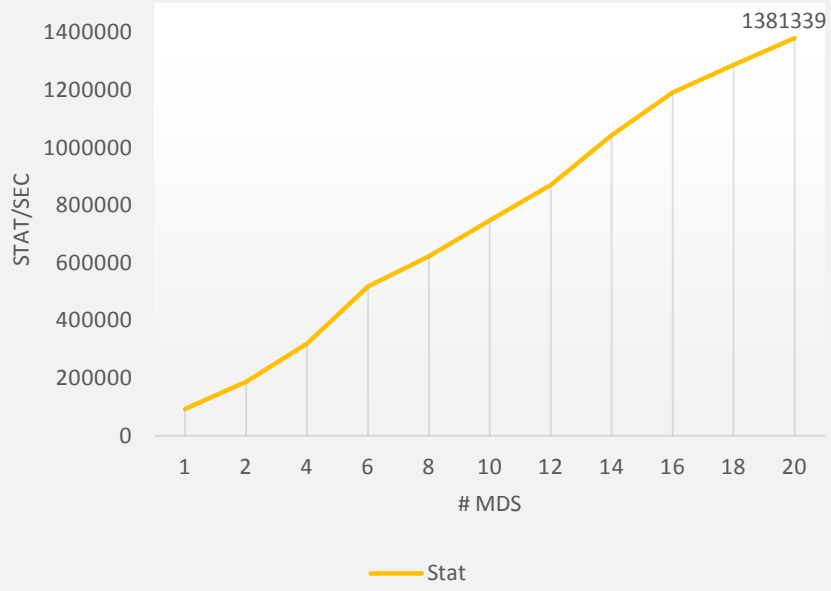
Note: Absolute numbers in these cases depend on per-server hardware performance, of course.

Metadata Scalability

File creation scalability with increasing number of metadata servers



File stat (attribute query) scalability with increasing number of metadata servers



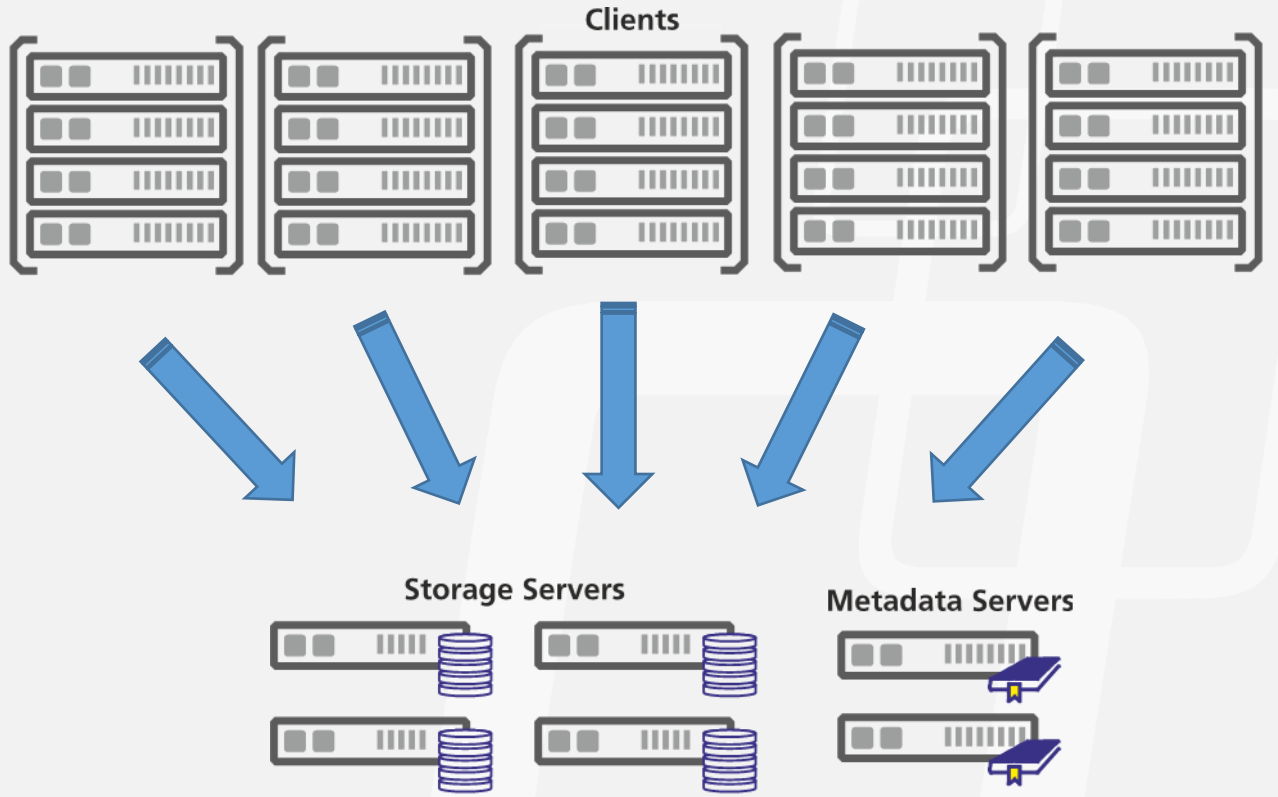
Note: Absolute numbers in these cases depend on per-server hardware performance, of course.

BeeOND – BeeGFS on Demand

Cluster File Systems

- Usually: One big file system for the whole cluster

- Despite parallelism:
  - Potential bottlenecks:
    - Network
    - Hard drives



Flexibility

- Any machine can run any BeeGFS service
- Multiple BeeGFS instances at the same time

BeeOND

- Per-job on-demand storage

The easiest way to set up a parallel FS

- Package part of BeeGFS repository

```
root@seislab-compute@node02 ~>yum search beeond
Loaded plugins: auto-update-debuginfo, dellsysid, security
============================= N/S Matched: beeond ==============
beeond.i686 : BeeOND
beeond.x86_64 : BeeOND
beeond-thirdparty-gpl.i686 : BeeOND Thirdparty GPL
beeond-thirdparty-gpl.x86_64 : BeeOND Thirdparty GPL

  Name and summary matches only, use "search all" for everything.
```

See `beeond –help` or wiki.beegfs.com/BeeOND for more information

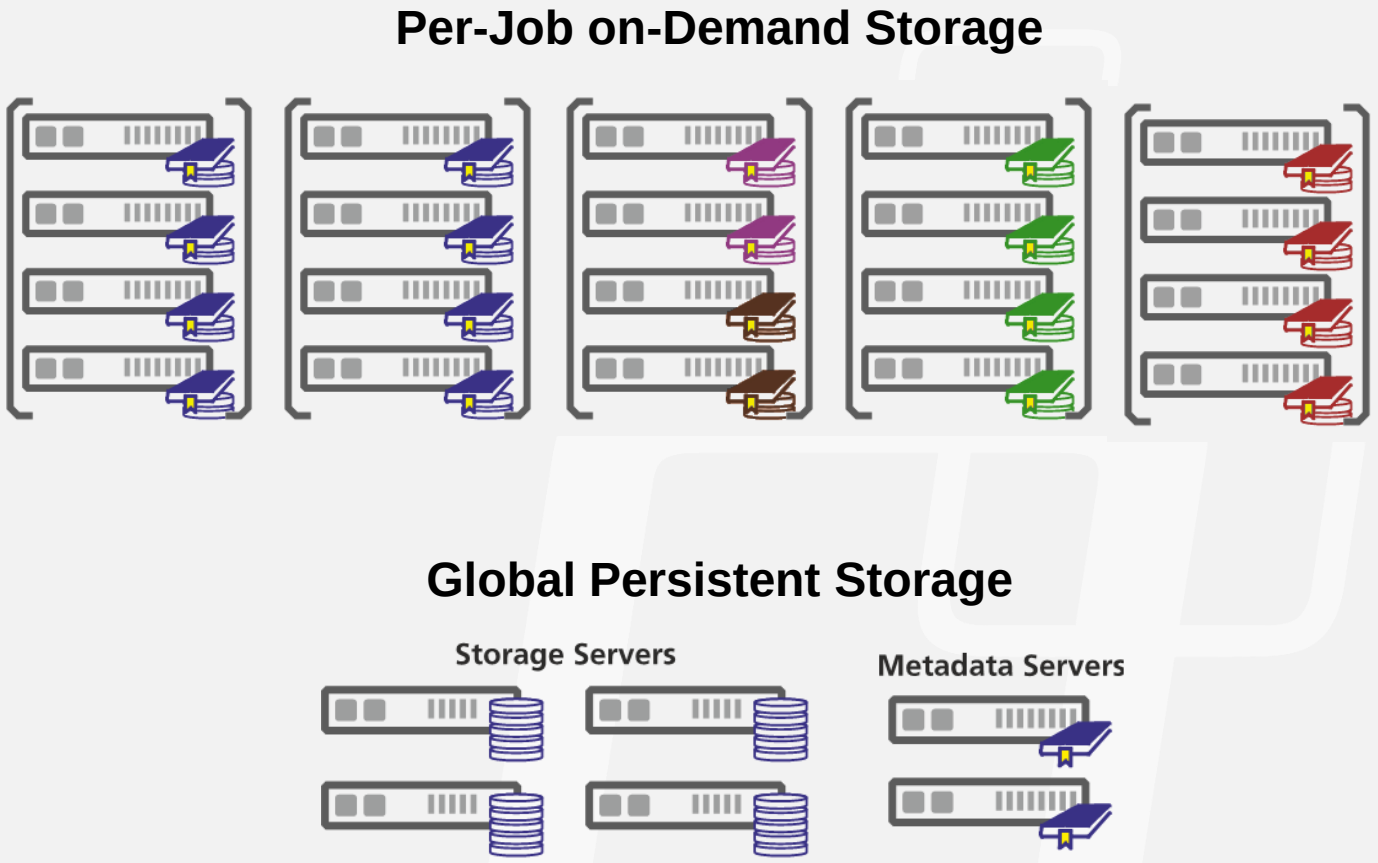
- Startup with a single command line

```
# beeond start –n nodefile –d /data/beeond –c /mnt/beeond
```

  - **nodefile**: contains all hostnames to run BeeOND on (one host per line)
  - **/data/beeond**: path, where BeeOND will save its raw data on each node
  - **/mnt/beeond**: mountpoint of the filesystem on each node

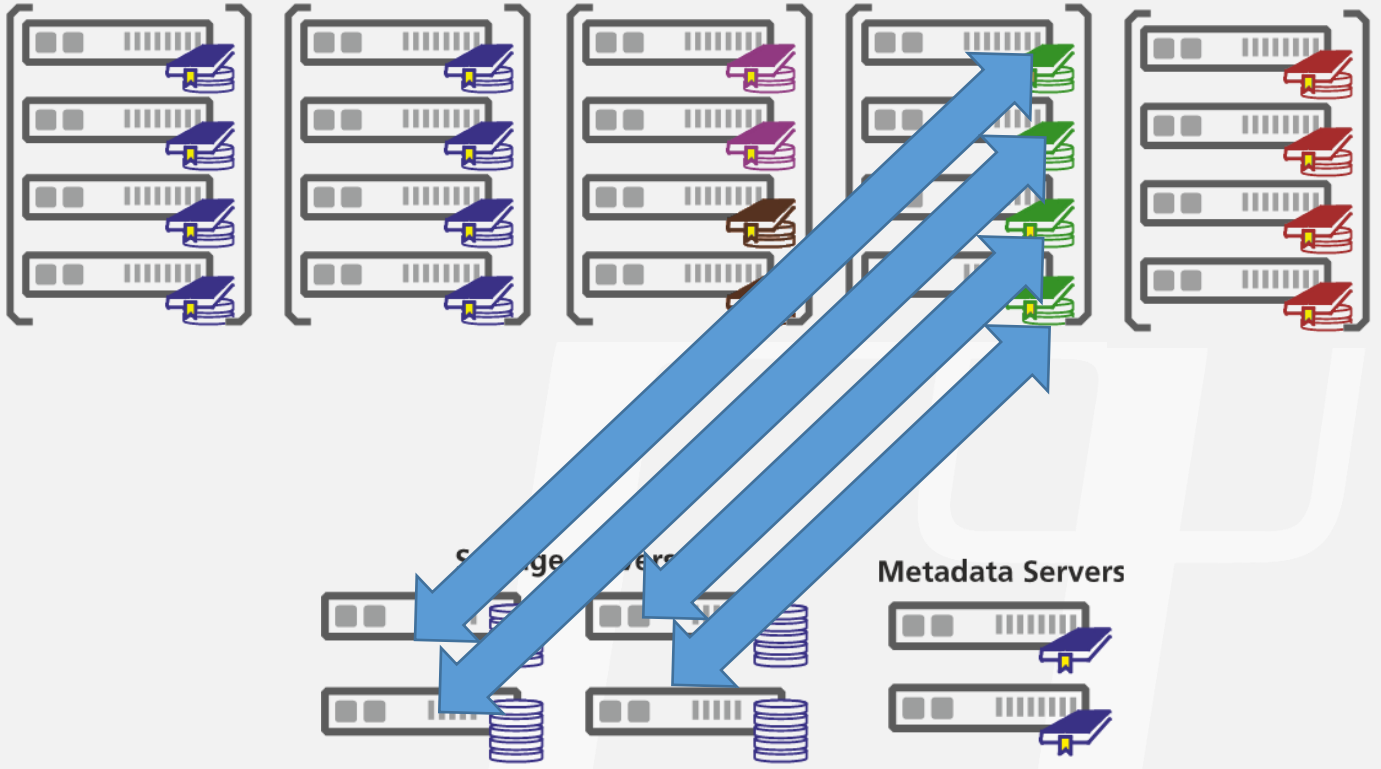
BeeOND – BeeGFS on Demand

- BeeGFS instance is started when job starts
- Provides temporary storage during job runtime
- Is shut down after compute job is finished
- Stage in/out process



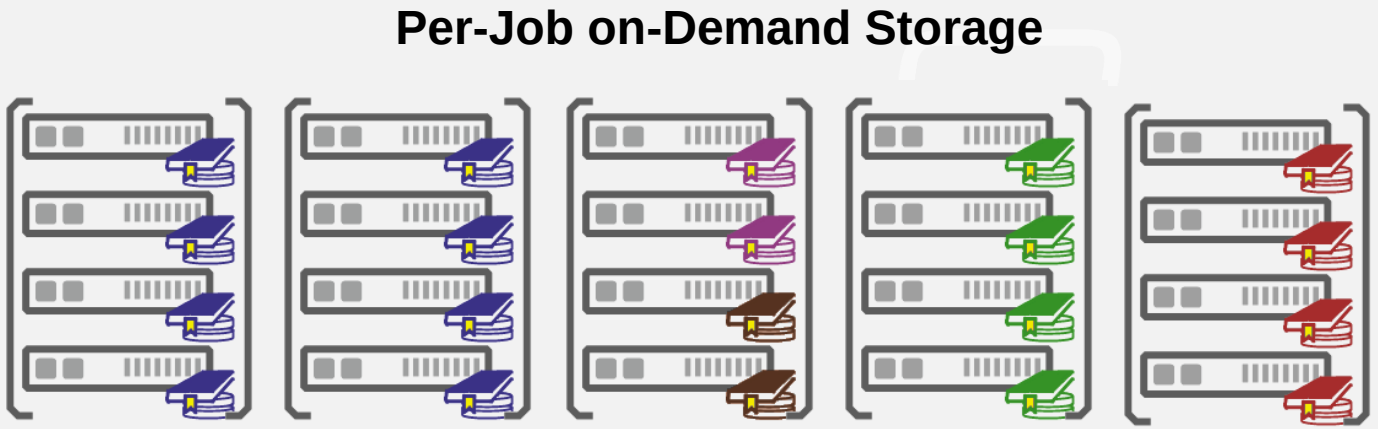
Stage-in / stage-out

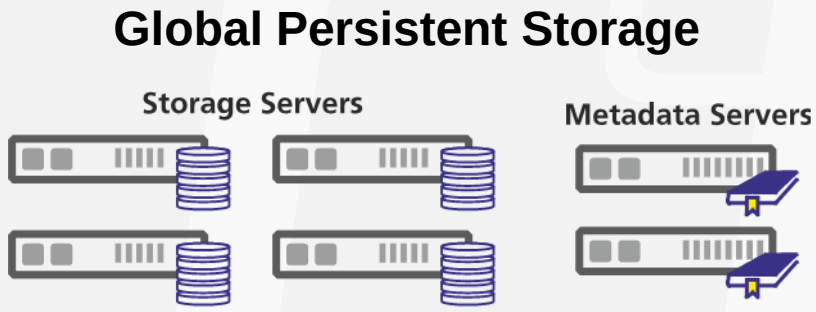
- Parallel copy from and to global store
- Optional
  - Depending on IO-pattern of job



Move storage closer to computation

- Scales with job size
- Closer to the computation
  - Less network traffic
- Reduces traffic on global storage servers
- Speeds up "dirty" IO patterns





Features

- Selectable number of storage + metadata servers
  - Diskless nodes can be client-only
- Multiple instances at the same time
- Access to all BeeGFS tuning parameters
- Use tmpfs
- Works with any kind of global storage layer (NFS, ...)

Use-cases

- Read input once
- Produce lots of temporary data
- Write output all at once

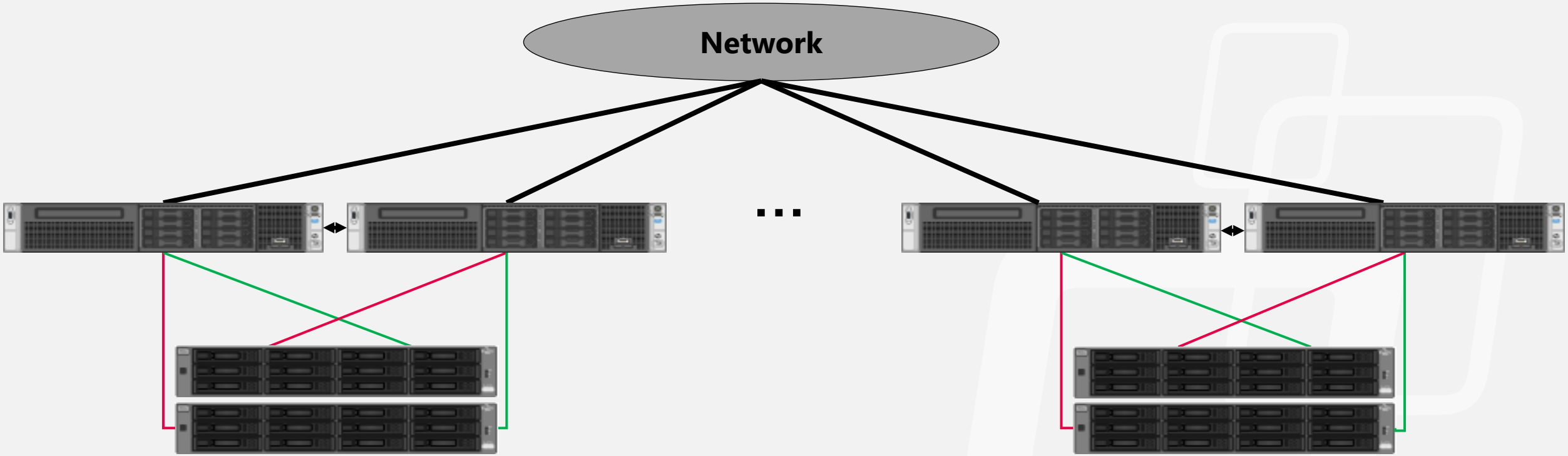
- Read input x-times
- No temporary data
- Write output in random pattern

- Read and modify small chunks of data in-place

- Some software can even be simplified when used with BeeOND:
  - Take advantage of one shared "temp" filesystem instead of one temp folder per node

Buddy Mirroring

High Availability - Shared Storage



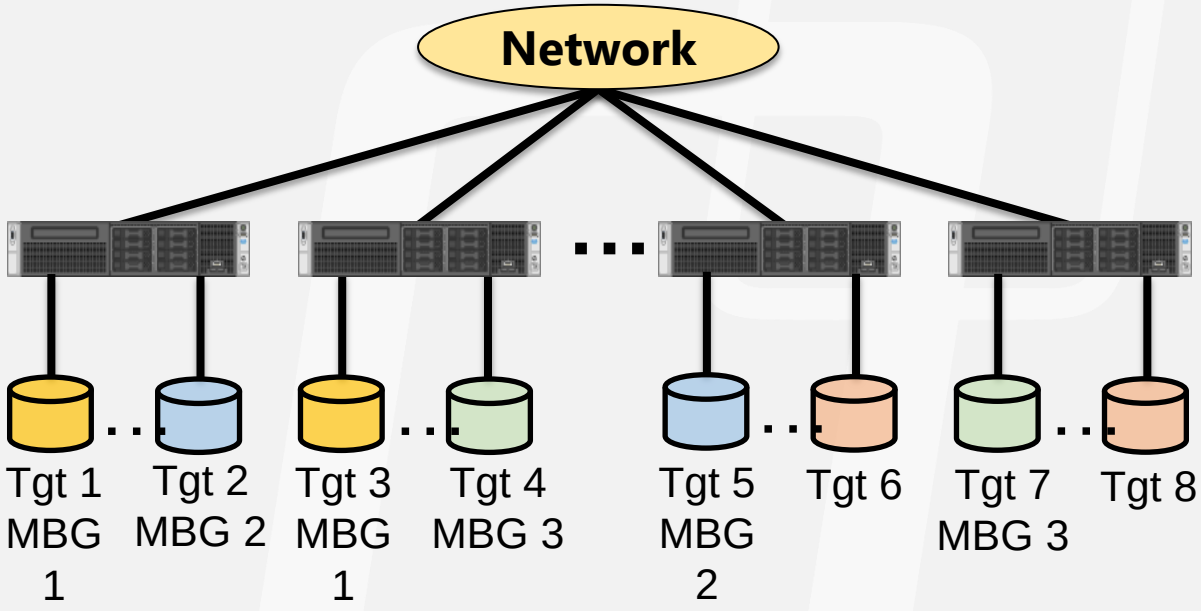
High Availability - Shared Storage

- No additional storage capacity needed

- Expensive storage components needed
- 3rd party software components needed
- Complex to set up and maintain
- Failover Risk
- No increased data safety

Built-in Data Mirroring

- Based on „mirror buddy groups“ of storage targets
  - Primary/secondary target in a buddy group internally replicate chunks
  - But: Targets can still also store non-mirrored chunks
  - Write operations are forwarded for high throughput
  - Read possible from both targets
- Internal failover mechanisms
  - In case primary is unreachable or fails, an automatic switch is performed
  - Self-healing (differential rebuild) when buddy comes back
- Flexible: Can be enabled globally or on a per-directory basis

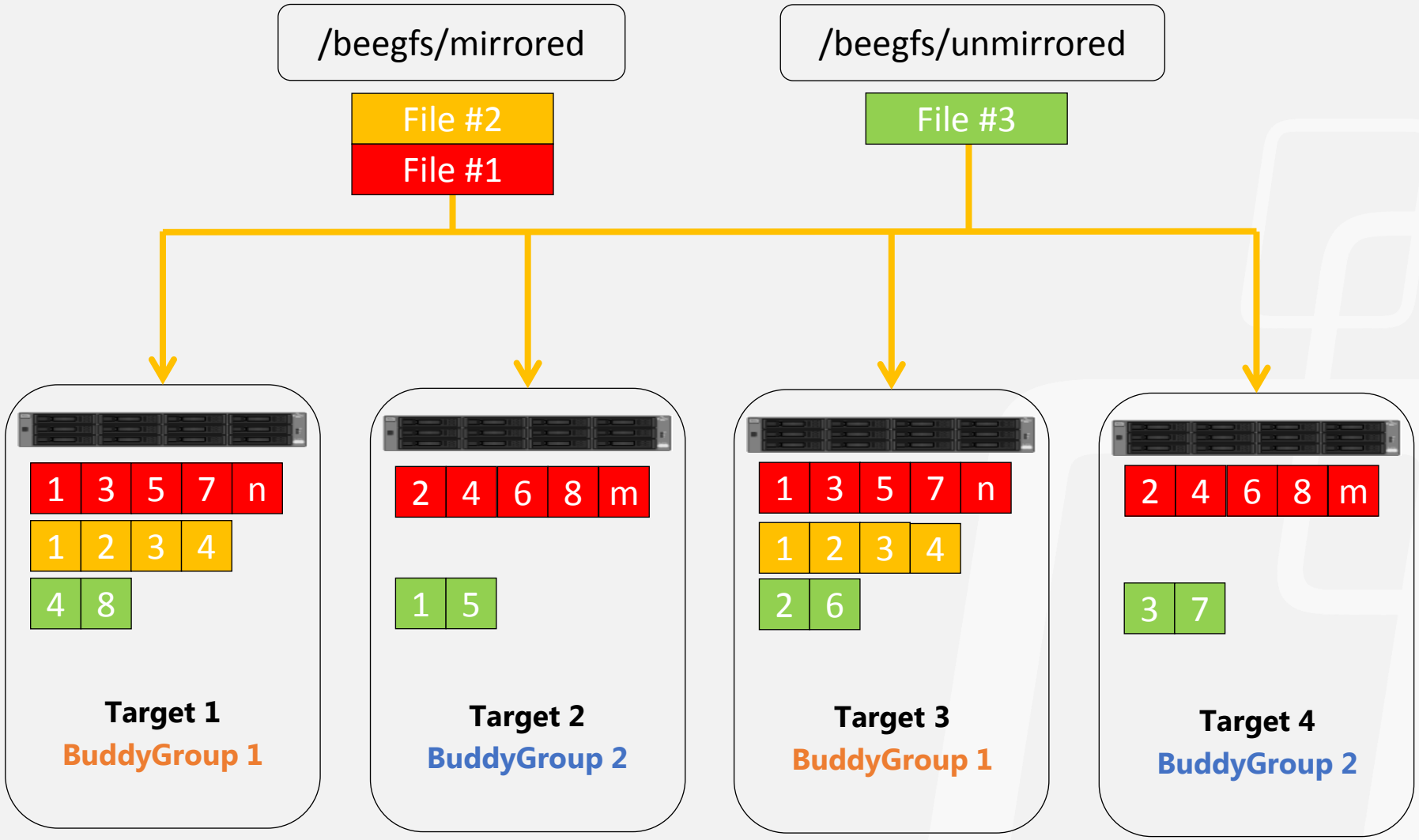


High Availability – Built-in Replication

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

- Overhead in storage capacity
- Write penalty for replicated data

Buddy Mirroring per Directory



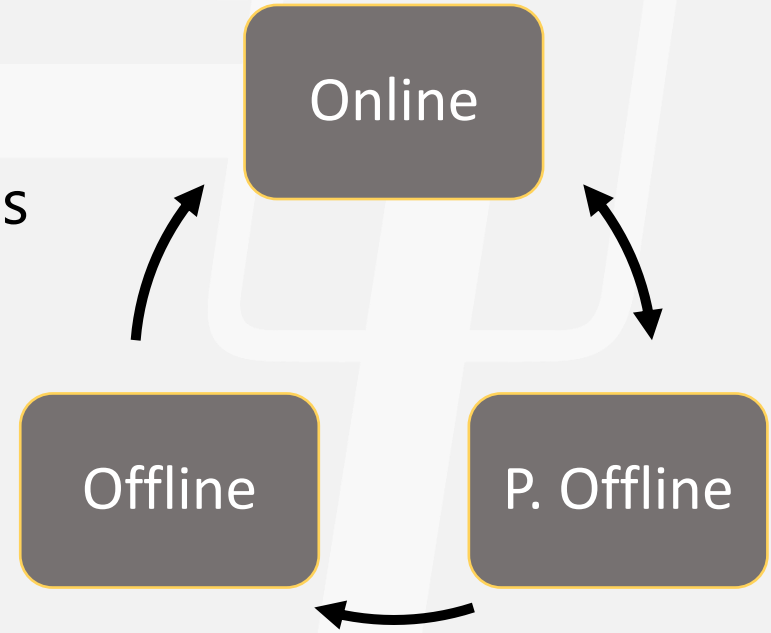
Target States

- Introduced in BeeGFS 2015.03
- Two different states for targets
  - Consistency state
  - Reachability state
- Internally used for several optimizations
- Important for HA
- Display with beegfs-ctl

```
mohrbach@seislab-master1:~$ beegfs-ctl --listtargets --state
TargetID     Reachability  Consistency   NodeID
========     ============  ===========   ======
       1           Online         Good    65055
       2           Online         Good    48360
       3           Online         Good    45532
       4           Online         Good    38454
       5           Online         Good    34731
       6           Online         Good    37553
       7           Online         Good    14879
       8           Online         Good    62821
       9           Online         Good     8360
      10           Online         Good    17740
      11           Online         Good    37553
      12           Online         Good    14879
      13           Online         Good    62821
      14           Online         Good     8360
      15           Online         Good    17740
      16           Online         Good    37553
      17           Online         Good    14879
      18           Online         Good    62821
      19           Online         Good     8360
```

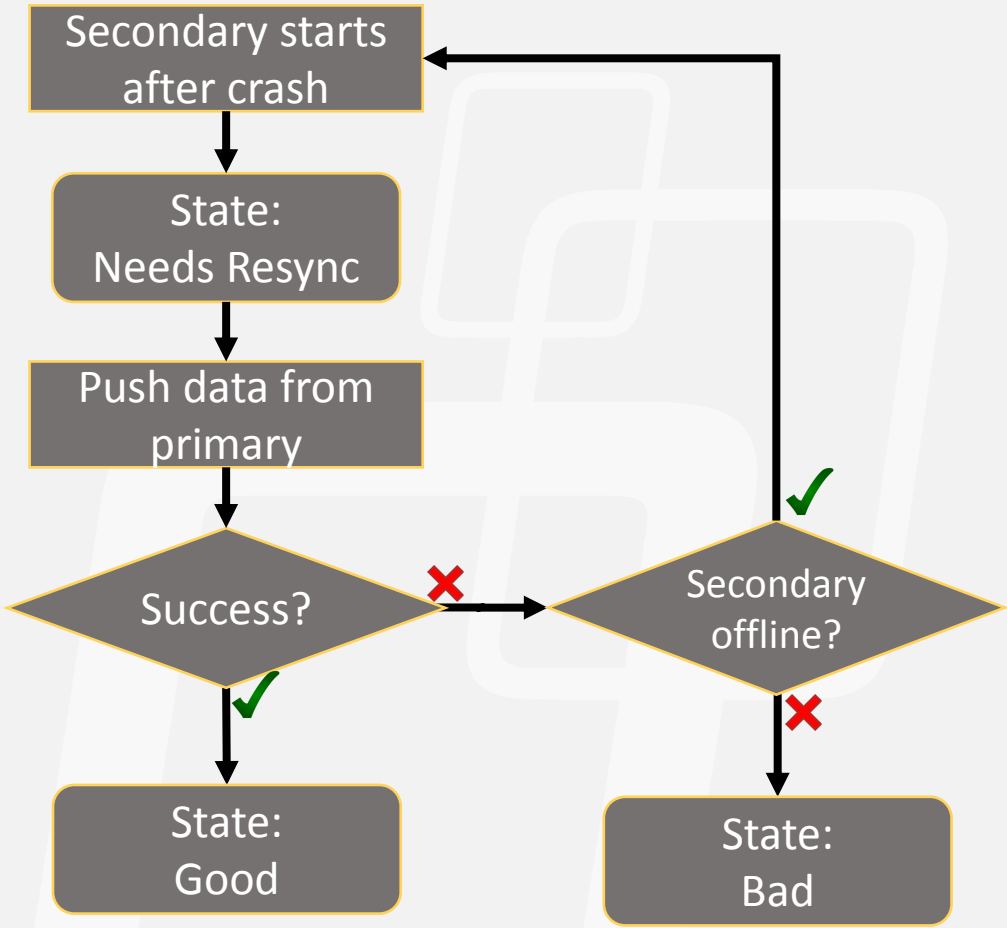
Reachability State

- Online
  - The target is reachable and fully usable by clients
- Probably Offline
  - The target might be offline
  - Mirrored files on this target may not be accessed
  - Non-mirrored files can be attempted (but may fail)
  - intermediate state avoid races and split-brain situations
- Offline
  - If target is part of a buddy group, try a switchover



Consistency State

- Good
  - Target may be used without limitations
- Needs Resync / Resyncing
  - Target needs a resync (or resync in progress)
  - Only valid for secondary targets
  - Clients may still access non-mirrored files
- Bad
  - A resync failed
  - Needs manual intervention

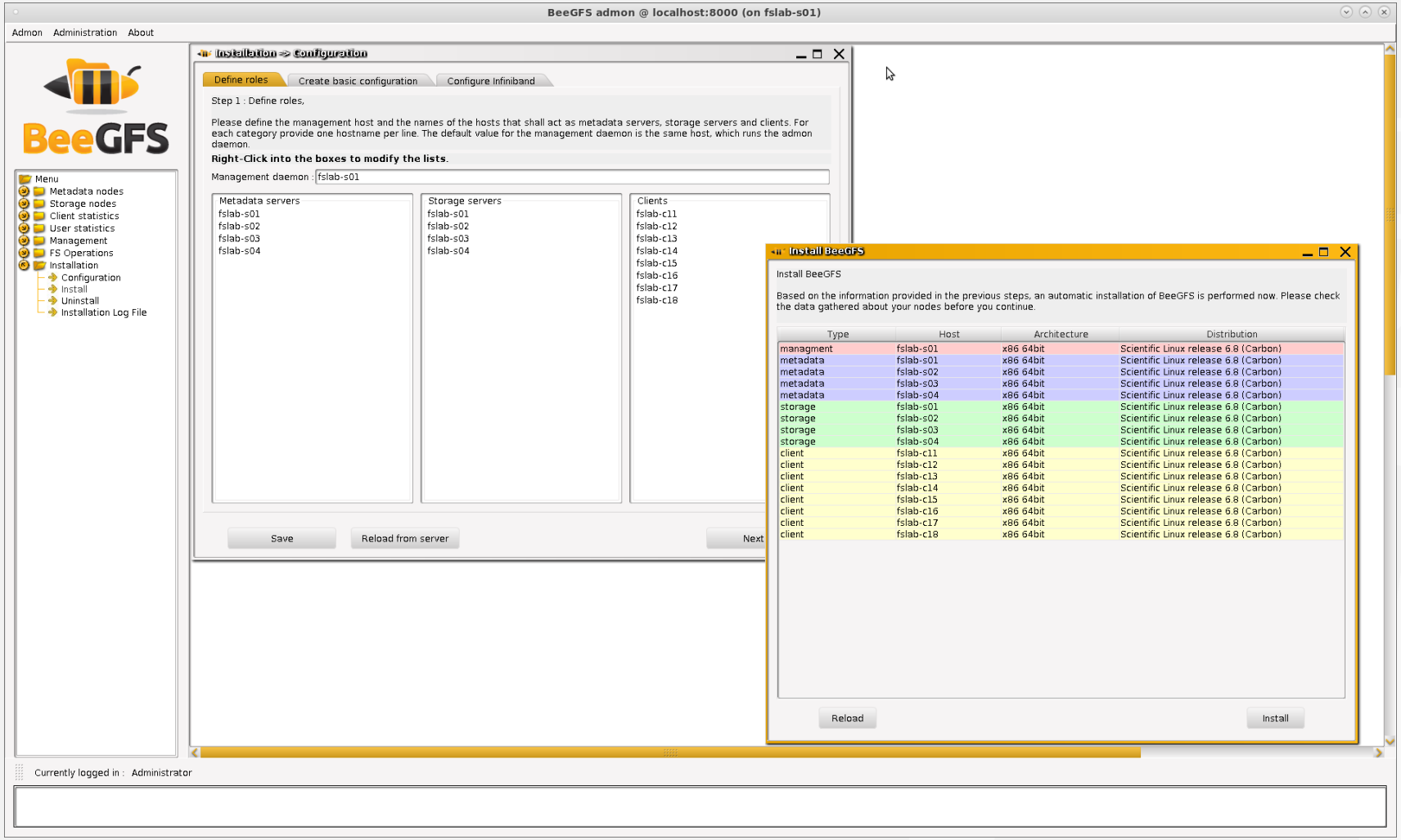


Resync

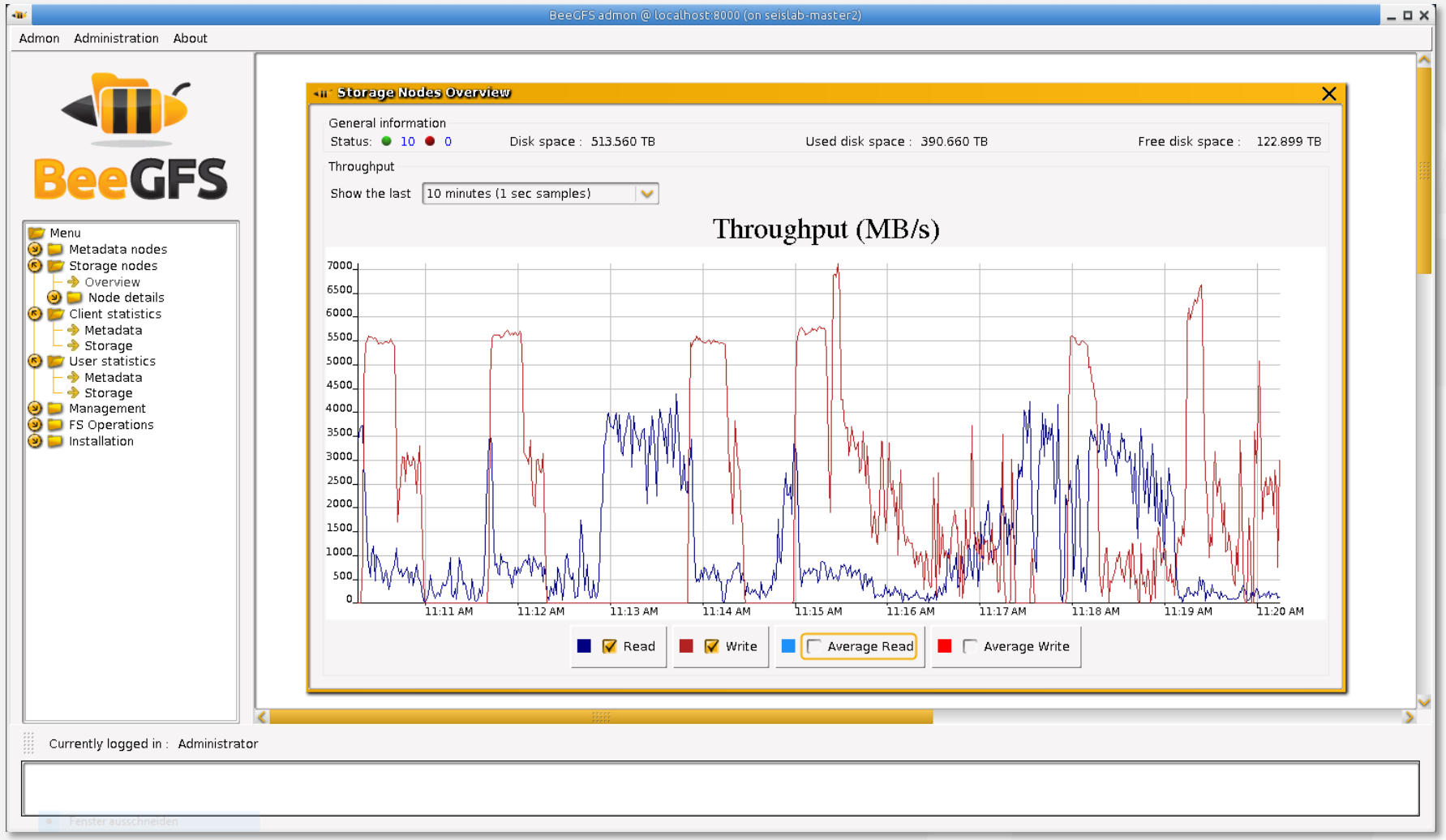
- Completely transparent & automatic in case a node comes back after a failure
- Pushes all chunks modified since last successful communication from the Primary to the Secondary
  - Optionally (e.g. for rebuild of completely lost storage target): All files
- Can also be triggered manually
- Statistics through beegfs-ctl

Tools & API

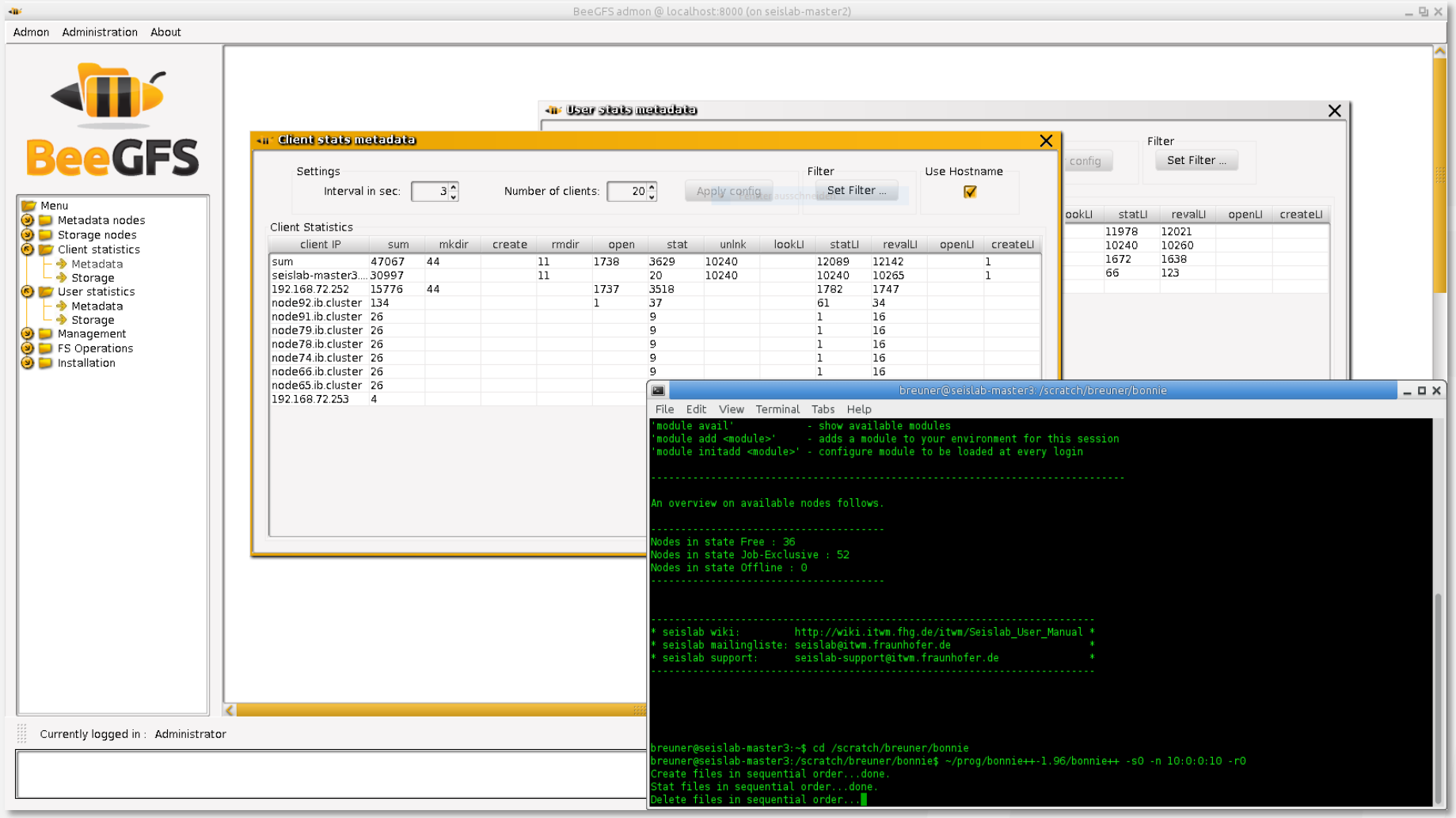
GUI for Windows-style Installation



Live Throughput Overview

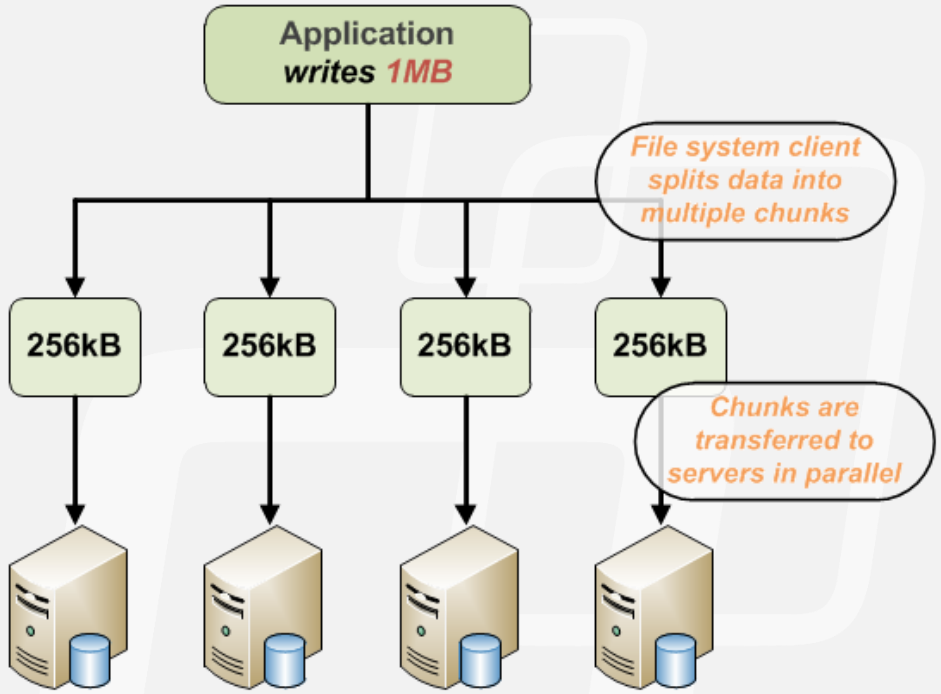


Live per-Client and per-User Statistics



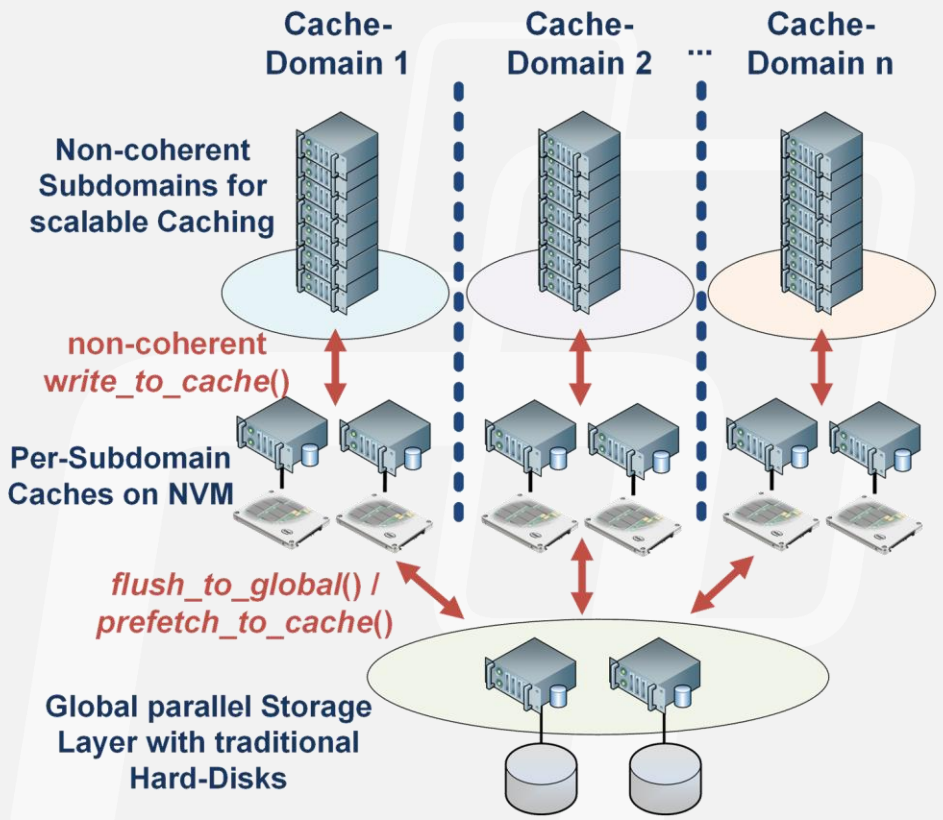
Striping API

- Include file beegfs.h
- Creates new files with stripe-settings
  - Block size
  - Number of strip targets
  - Stripe pattern
- Gets the striping information of a file
- Checks a given path if the file-system is a BeeGFS



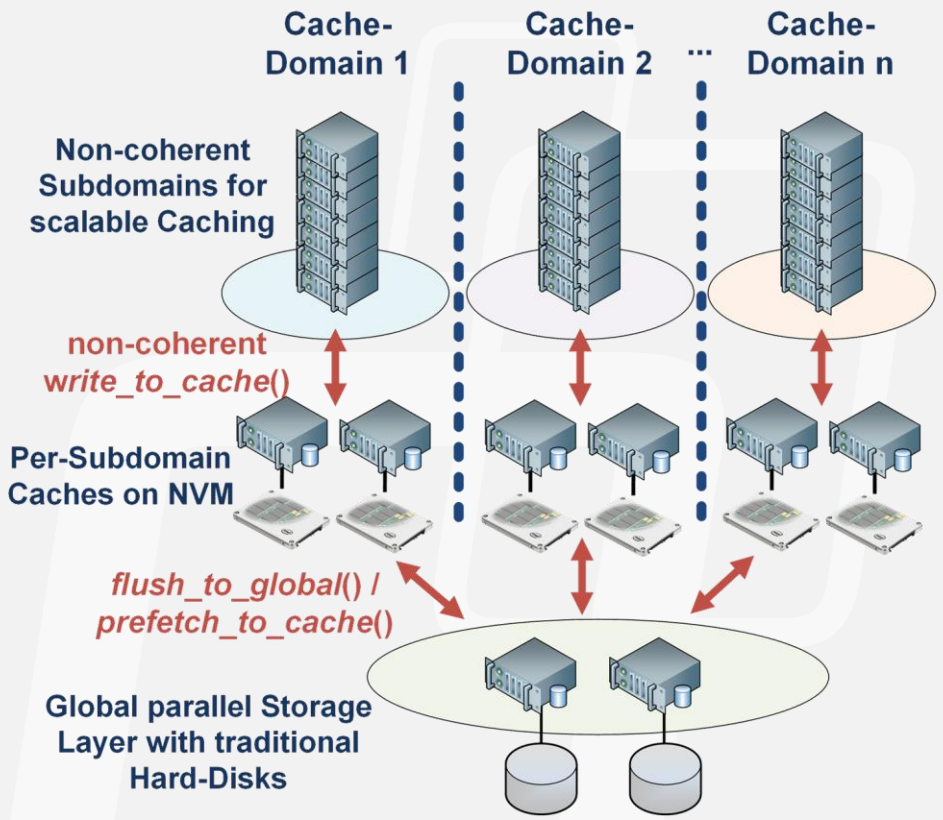
Cache API

- Two storage layers
  - Globally visible layer for capacity on traditional HDDs
    - Avoid accessing this layer as much as possible for maximum scalability
  - Partitioned cache layer for work-in-progress data
    - HDD or High-throughput solution with SSD or NVMe
    - Cache layer is accessed through new BeeGFS Cache API
    - Use especially for temporary files and data that is not written in large chunks



Cache API

- Include file deeper/deeper_cache.h
- Open()
  - Additional options which are executed during the close() ➔ discard or flush on close
- Close()
- Prefetch()
  - Copy a file/directory from global FS to cache FS
- Flush()
  - Copy a file/directory from cache FS to global FS



BeeGFS News

What's new in the recent Major Releases?

- BeeOND (BeeGFS On Demand)

- ***Enterprise Edition*** *(Codename: Trinity)*
  - Quota Enforcement
  - Access Control Lists (ACLs)
  - Built-in data mirroring and metadata mirroring

- Per-User Statistics in Admon GUI
- New setup tools (/opt/beegfs/sbin/beegfs-setup…)
- BeeGFS C API
- …

License Model

- **BeeGFS is free to use for end users: www.beegfs.com/download**
  - Ready-to-use binary packages
  - Complete source code also available (but: BeeGFS is intentionally not a community project)
- **System integrators/partners for turn-key solutions**
  - System setup and tuning
  - First point of contact (1st- and 2nd-level support)
  - Partners make back2back contract with ThinkParQ for 3rd-level support
- **Professional 3rd-level support contract**
  - Pricing based on number of servers and timeframe (e.g. 3 or 5 years)
  - Access to enterprise edition features
  - Special customer website area: www.beegfs.com/customerlogin

transtec

BeeGFS allows us to easily deliver peta-scale turn-key storage solutions
- transtec

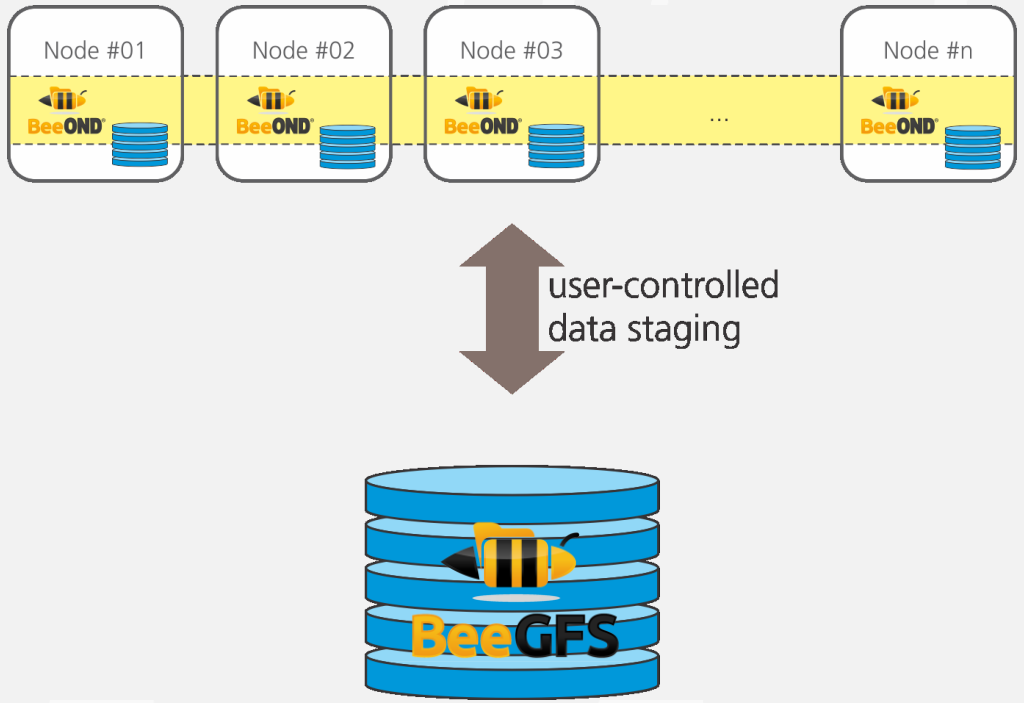
→ Support contracts are also the financial basis for development of great new features

Global Turn-Key Solution Partners

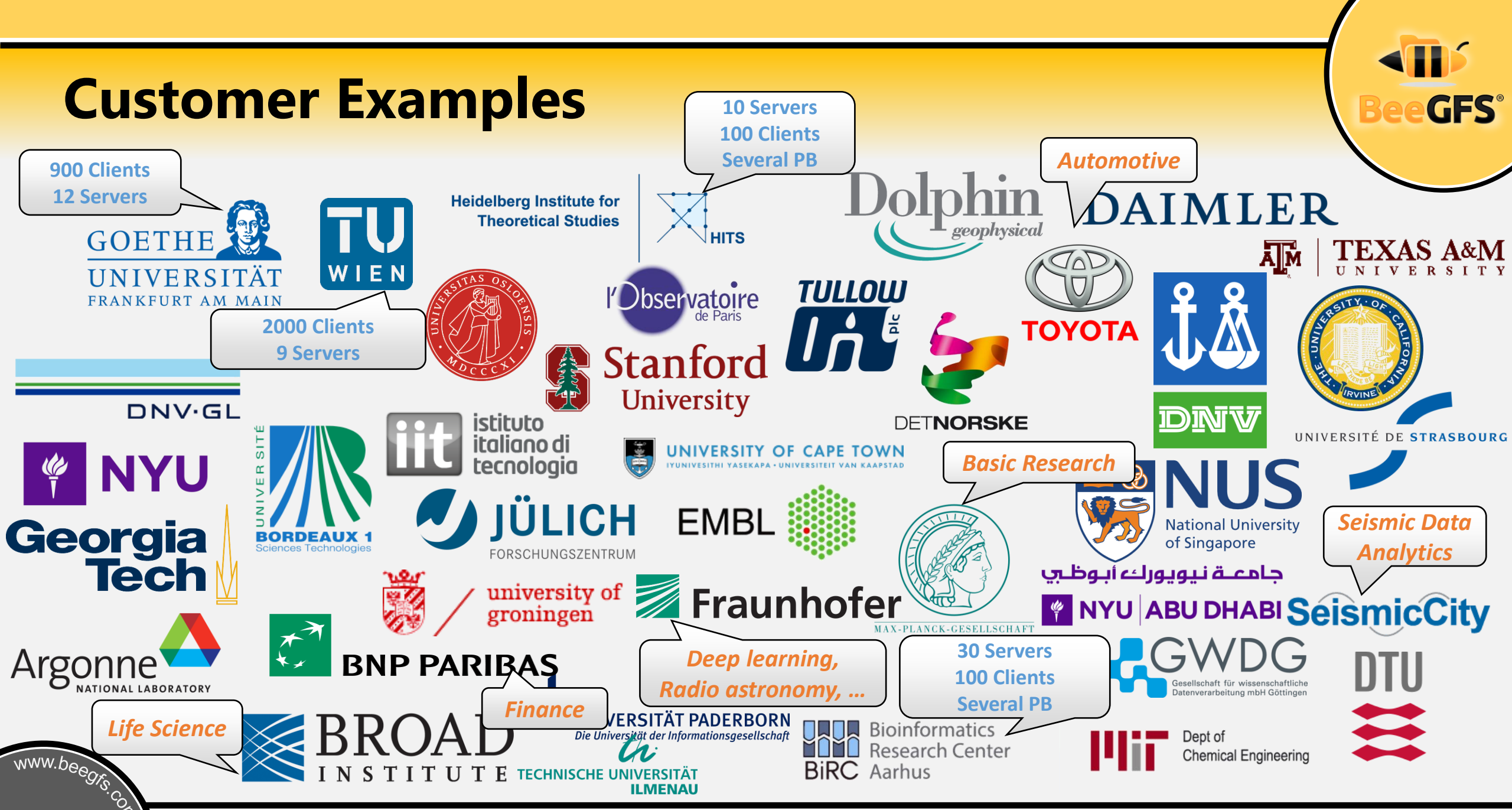
BeeGFS in use

BeeOND - Use in Fraunhofer Seislab

- Fraunhofer Seislab
  - In-house cluster of Fraunhofer HPC Department
  - 100 compute nodes with 1 TB of SSDs each
  - Global BeeGFS storage on spinning disks
- Create BeeOND on SSDs on job startup
  - Integrated into Torque startup
- Stage-in input data, work on BeeOND, stage-out results



Customer Examples



Why do people go with BeeGFS?

"We are extremely happy with our 3.1PB BeeGFS on 30 servers – it runs rock-solid."
- Bioinformatics Aarhus, Denmark

FRED HUTCH
CURES START HERE™

"We're now at one year of uptime without a single hitch of the BeeGFS."
- Fred Hutchinson Medical Institute, USA

"After many unplanned downtimes with our previous parallel FS, we moved to BeeGFS more than 2 years ago. Since then we had no unplanned downtimes anymore."
- University of Halle, Germany

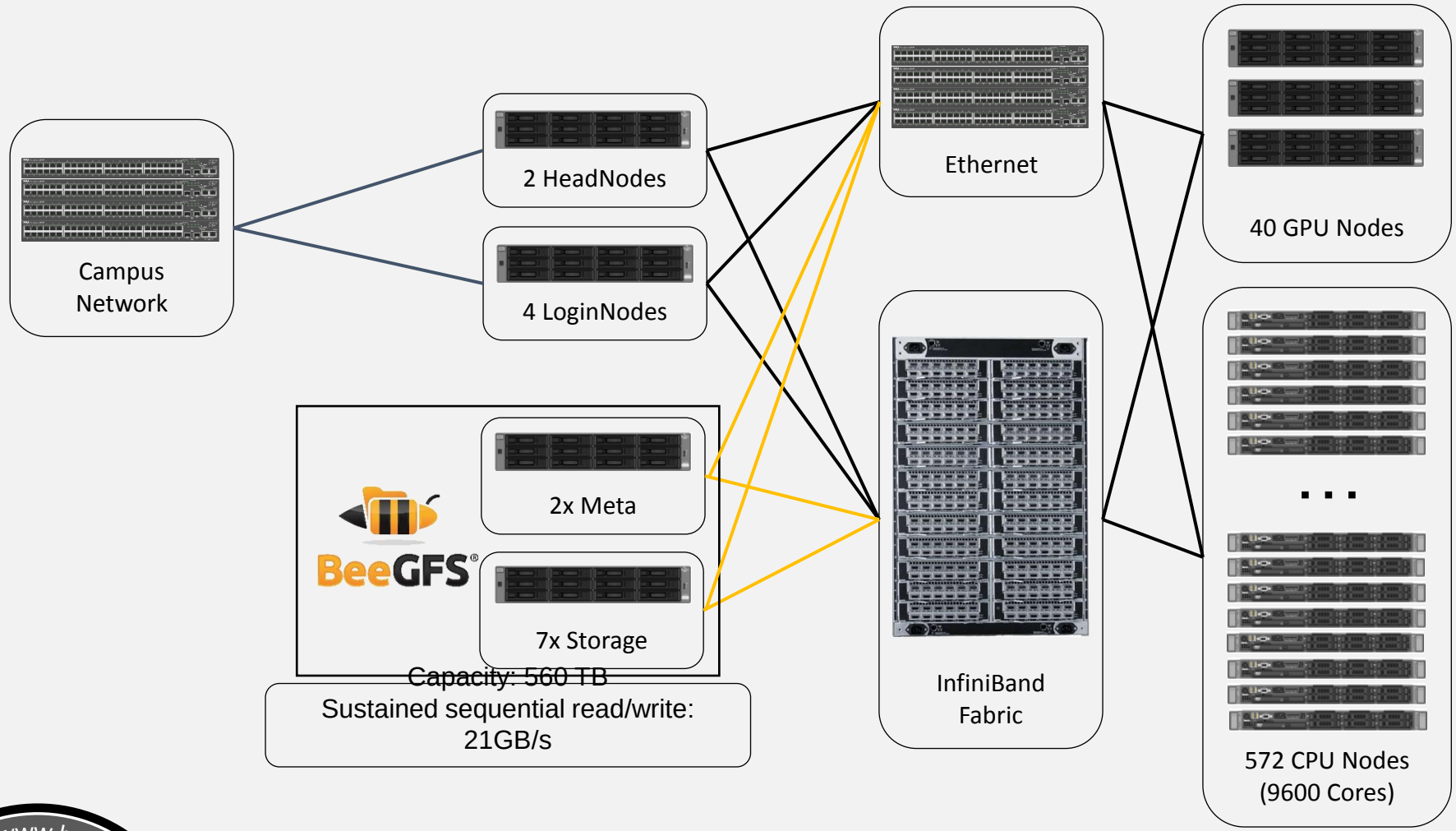
"We've seen BeeGFS performing very well and also saw its easy and robustness – which we did not see in many other parallel file systems."
- ClusterVision

clustervision

Currently about 250 supported installations spread all around the globe.
(By the way: all of them are happy)

Plus 1000's of unsupported installations
(free download, open-source)

OcULUS – A Typical HPC Installation

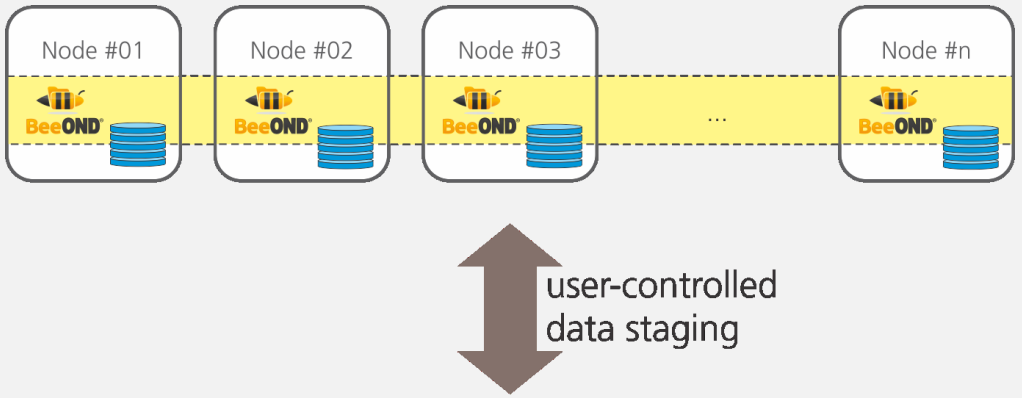


UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

BeeOND - Use at Alfred-Wegener-Institute

- Cray CS-series Cluster
  - Global BeeGFS storage on spinning disks (***4 servers, 16GB/s***)
  - 300 compute nodes with a 500MB/s SSD each

    150GB/s aggregate BeeOND speed "for free"
- Create BeeOND on SSDs on job startup
  - Integrated into Slurm prolog/epilog script
- Stage-in input data, work on BeeOND, stage-out results



ALFRED-WEGENER-INSTITUT
HELMHOLTZ-ZENTRUM FÜR POLAR- UND MEERESFORSCHUNG

A workload-optimized solution (1/3)

Customer Situation

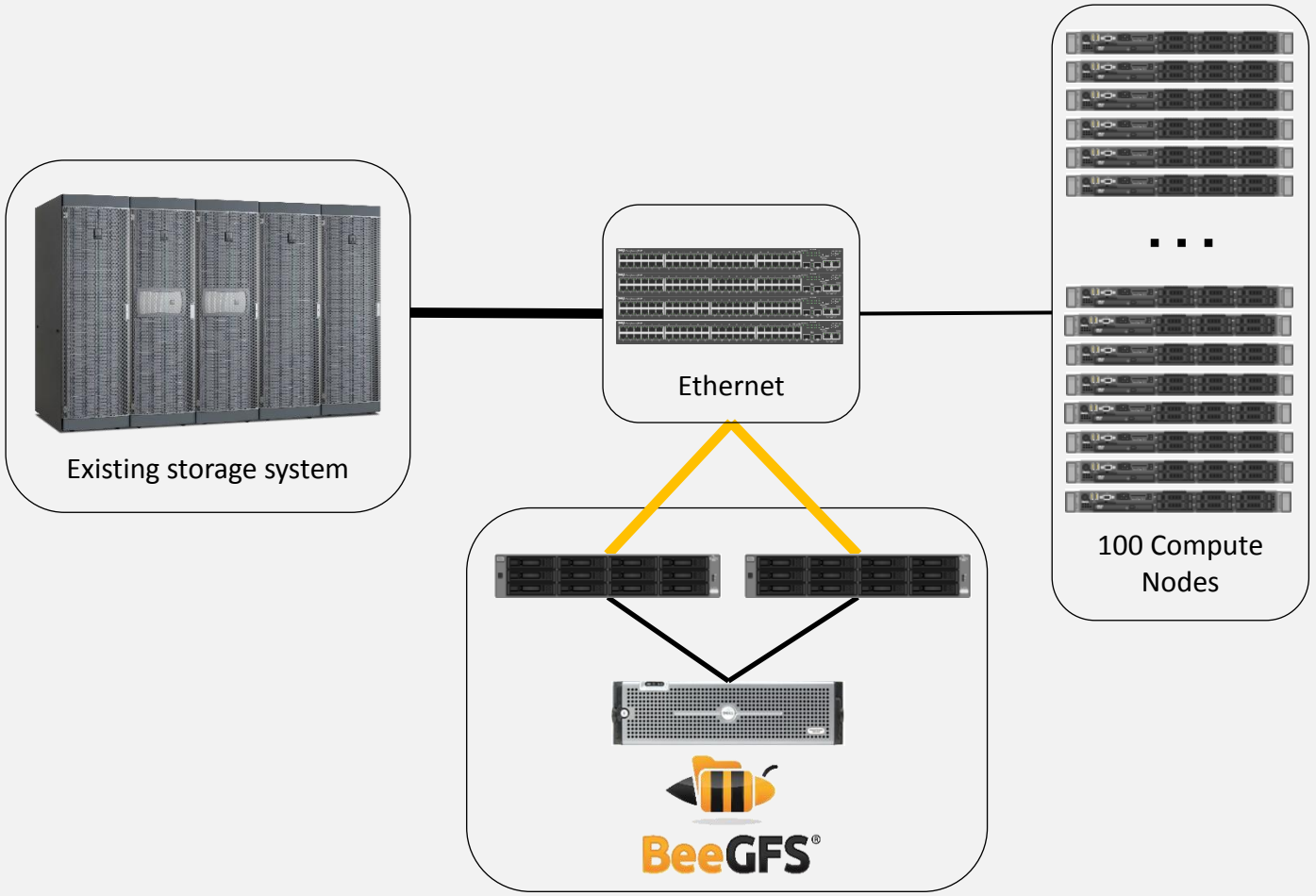
- 2 Mio EUR proprietary solution
- served home directories and HPC work storage
- HPC compute jobs were all single core (life science) working on the same data
- Compute nodes uplinked with 1GbE
- 30 to 40 single core compute jobs saturated the storage, making it completely unresponsive

Customer's Expectation

- Build a dedicated storage just for the HPC cluster
- Use fast 10k or 15k disk drives – or even SSDs
- Create a system serving 100k+ random IOPS

EMBL

A workload-optimized solution (2/3)



EMBL

A workload-optimized solution (3/3)

Findings through workload analysis

- About 0.5 TB of cache should be sufficient
- Disk performance becomes irrelevant then
- BeeGFS can make use of 40GbE very well

Solution

- 2 storage servers – together 1 TByte RAM
- 40GbE uplinks for the storage servers
- 60 disk drives for persistent storage
- Tested with 600 single core compute jobs and no drop in performance

EMBL

Another workload-optimized solution (1/2)

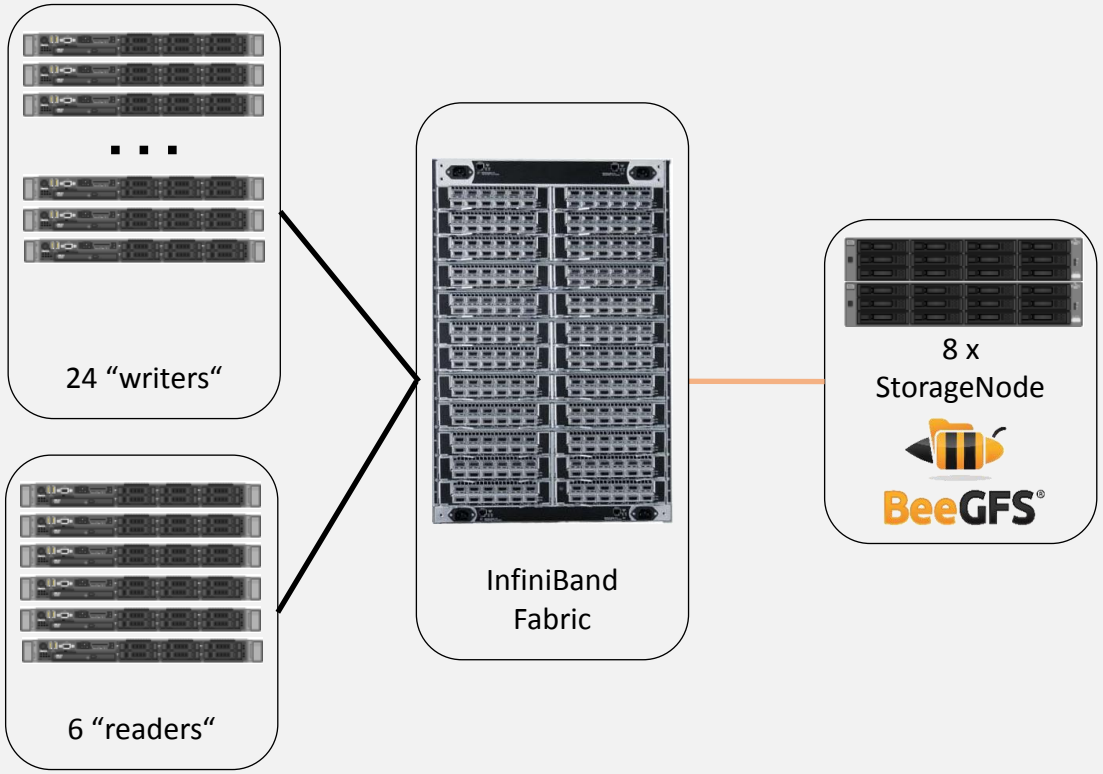
Situation

- 16 to 32 clients write sequential data, each client a unique data set
- 2 to 8 clients read through (some of) these datasets
- No “writer“ needs to access data of any other writer, “readers“ have to see everything
- Reads are executed when no writes are done
- **Required sequential writes of 30+ GB/s**

Customer Expectations

- **Based on previous systems the customer expected to need 800 to 1000 disks in several fibre channel based systems**
- FDR InfiniBand to get enough throughput for the „readers“

Another workload-optimized solution (2/2)



Solution

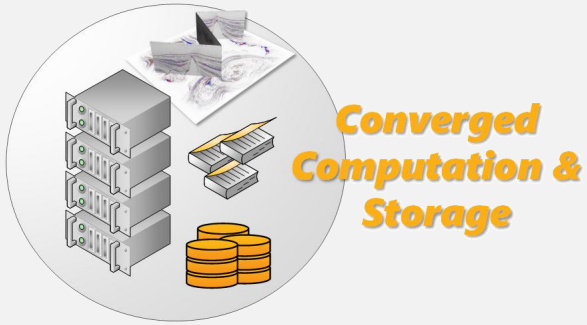
- 8 storage nodes, each with FDR Infiniband
- 60 disks per server,
  480 drives total
- 3 RAID controllers per server,
  RAID6 8+2

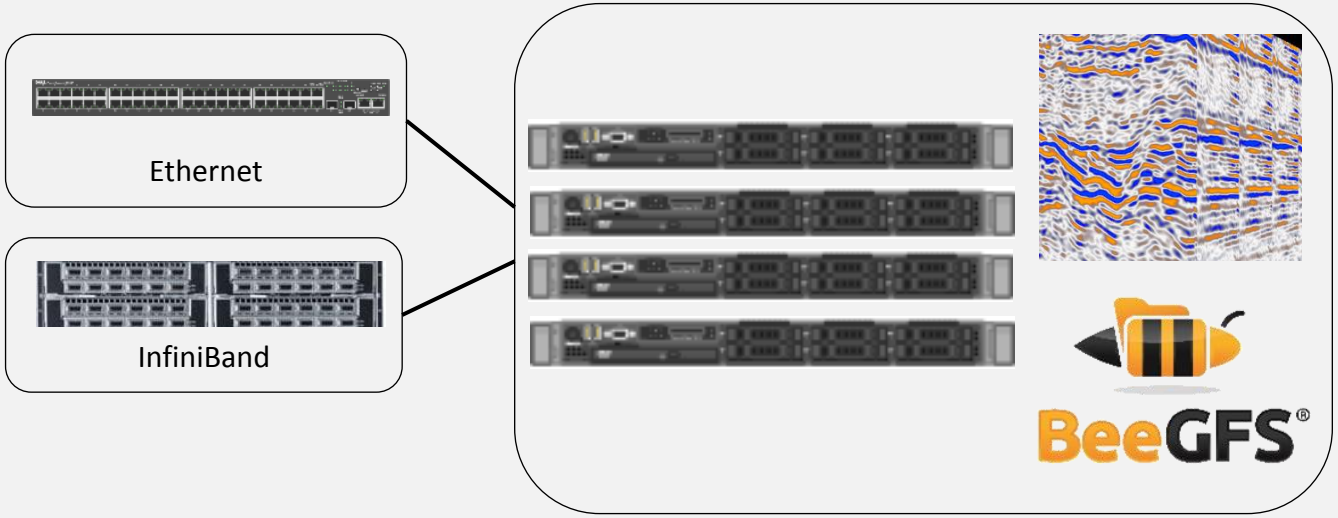
Performance

- **40 GB/s sustained** writes
- Each reader gets 4+ GB/s
- 1.5 PByte usable capacity

A hyper-converged solution

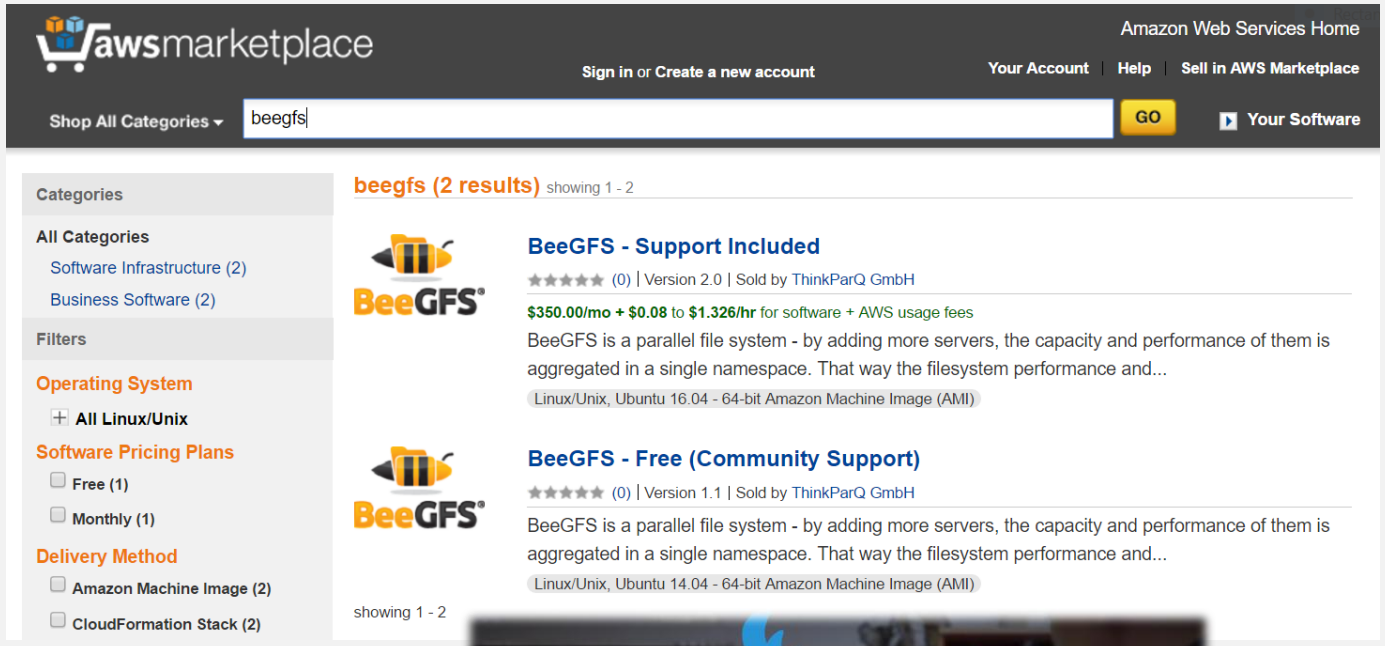
- Four compute nodes for seismic data interpretation
- 12x 3.5“ drives per compute node
- BeeGFS running on compute nodes,
  no separate/dedicated storage servers





Bees up in the Clouds

Already available…
(Thanks to funding and technical support from Amazon)



Upcoming…
(Thanks to funding and technical support from Microsoft)

To infinity and beeond...

BeeGFS is participating in exascale projects to stay on the bleeding edge...

DEEP-ER

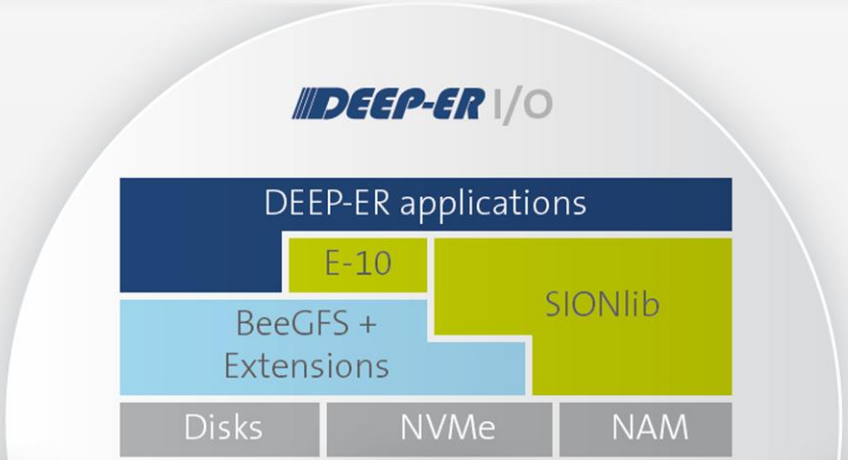
ExaNeSt

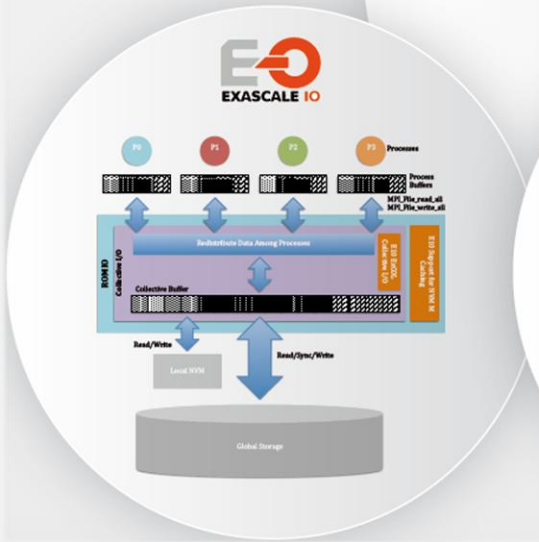
...

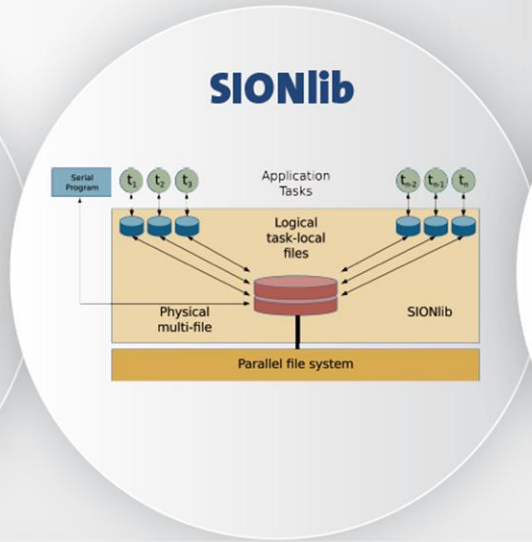
The DEEP-ER European Exascale Project (FP7)

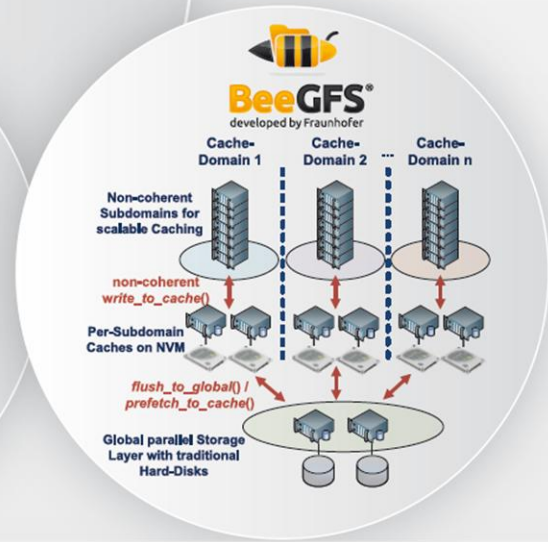
- DEEP-ER extends cluster/booster concept of DEEP project with focus on application resiliency/recovery and scalable I/O

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement n°610476.





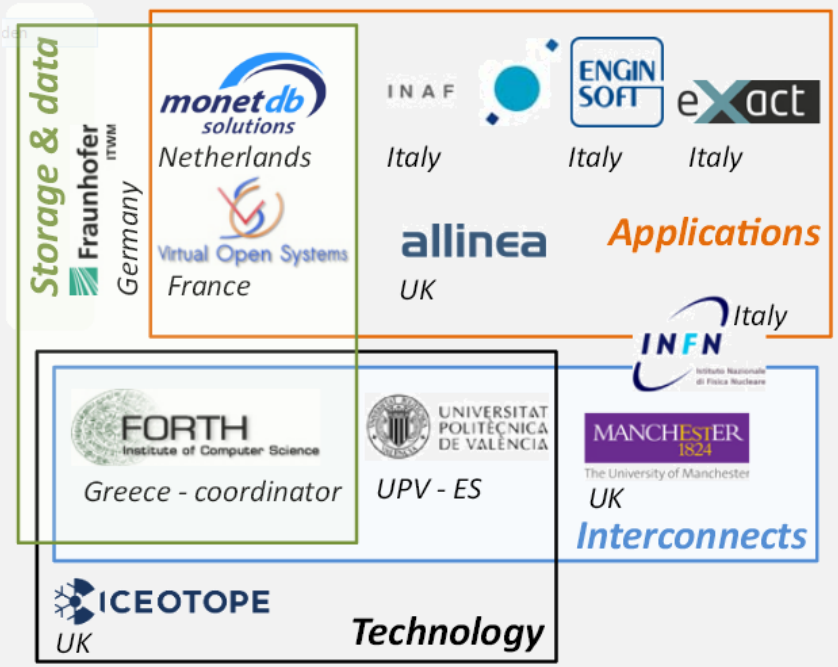




ExaNeSt (Horizon2020)

- **Exascale Network & Storage**
  EU Project

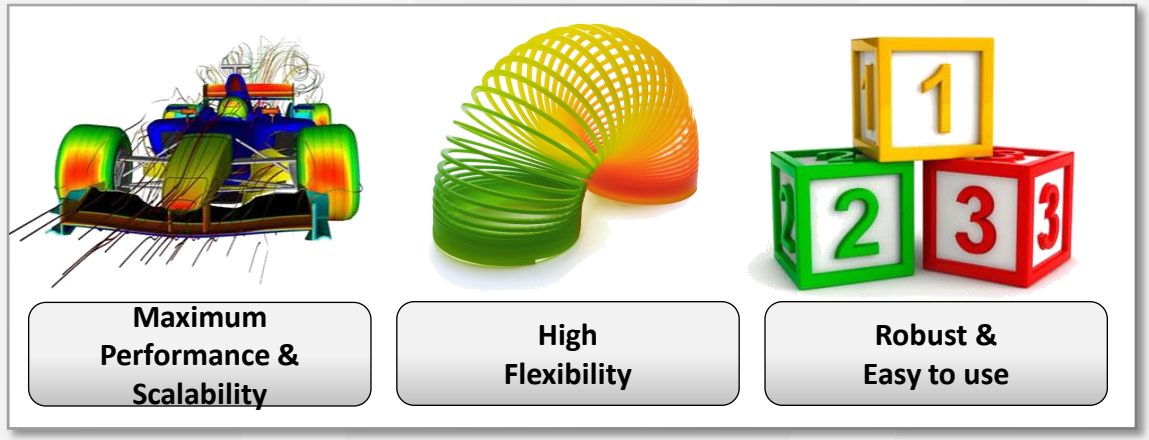
- Based on **ARM servers**
  with new UNIMEM
  architecture/interconnect for
  dynamic memory sharing



Conclusion

Conclusion

- Easy to use and robust parallel file system
- Excellent performance based on highly scalable architecture
- Maximum flexibility with some unique features for special workloads
- Turn-key solutions available, backed by excellent professional support from development team



Questions? / Keep in touch

- **Web** www.beegfs.com
  www.thinkparq.com
- **Mail** sales@thinkparq.com
  support@beegfs.com
- **Twitter** www.twitter.com/BeeGFS
- **Newsletter** www.beegfs.com/news