Karlsruhe Institute of Technology

# Extraordinary HPC file system solutions at KIT

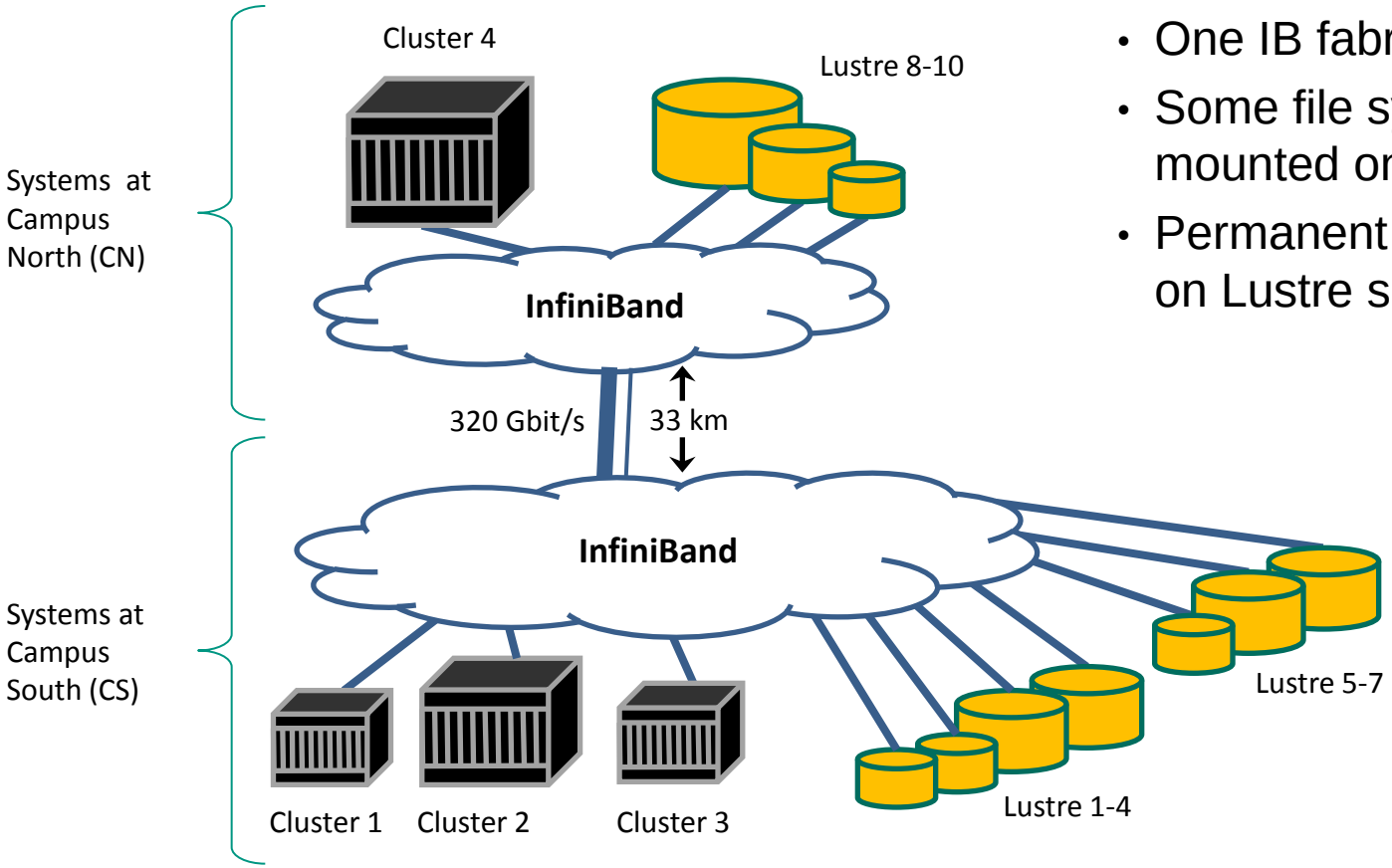
**Roland Laifer**

STEINBUCH CENTRE FOR COMPUTING - SCC

# Overview

- Lustre systems at KIT
  - and details of our user base
- Using Lustre over a 30 km InfiniBand connection
- Lightweight I/O statistics with Lustre
  - Helpful for users and administrators
- Disaster recovery for huge file systems
  - We recently had to use it

# Lustre systems at KIT - diagram



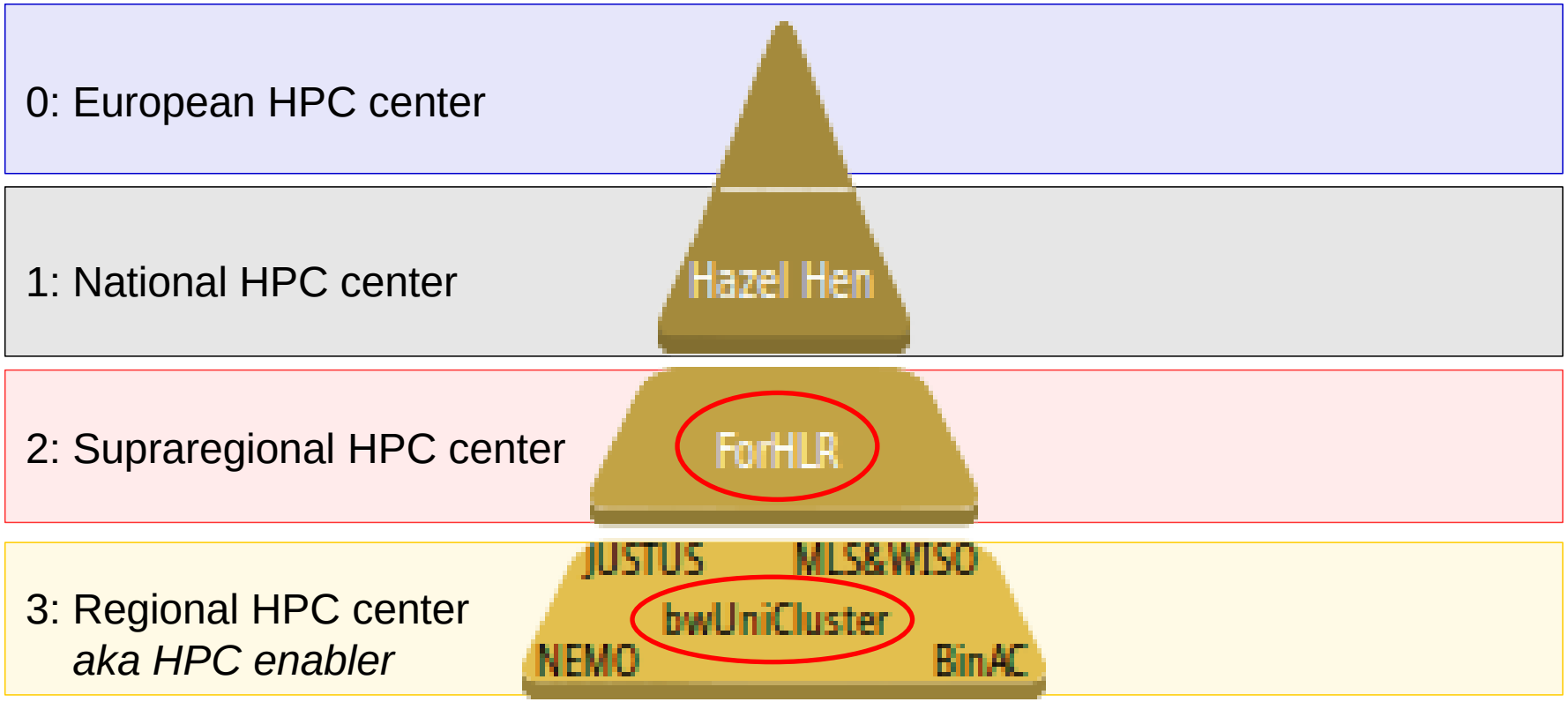

- One IB fabric
- Some file systems mounted on all clusters
- Permanent data (HOME) on Lustre since 2005

# Lustre systems at KIT - details

| System name | pfs2 | pfs3 | pfs4 |
|---|---|---|---|
| **Users** | universities, all clusters | universities, tier 2 cluster (phase 1) | universities, tier 2 cluster (phase 2) |
| **Lustre server version** | DDN Exascaler 2.3 | DDN Exascaler 2.4 | DDN Exascaler 2.3 |
| **# of clients** | 3100 | 540 | 1200 |
| **# of servers** | 21 | 17 | 23 |
| **# of file systems** | 4 | 3 | 3 |
| **# of OSTs** | 2*20, 2*40 | 1*20, 2*40 | 1*14, 1*28, 1*70 |
| **Capacity (TiB)** | 2*427, 2*853 | 1*427, 2*853 | 1*610, 1*1220, 1*3050 |
| **Throughput (GB/s)** | 2*8, 2*16 | 1*8, 2*16 | 1*10, 1*20, 1*50 |
| **Storage hardware** | DDN SFA12K | DDN SFA12K | DDN ES7K |
| **# of enclosures** | 20 | 20 | 16 |
| **# of disks** | 1200 | 1000 | 1120 |

# bwHPC

Baden-Württemberg's implementation strategy for HPC

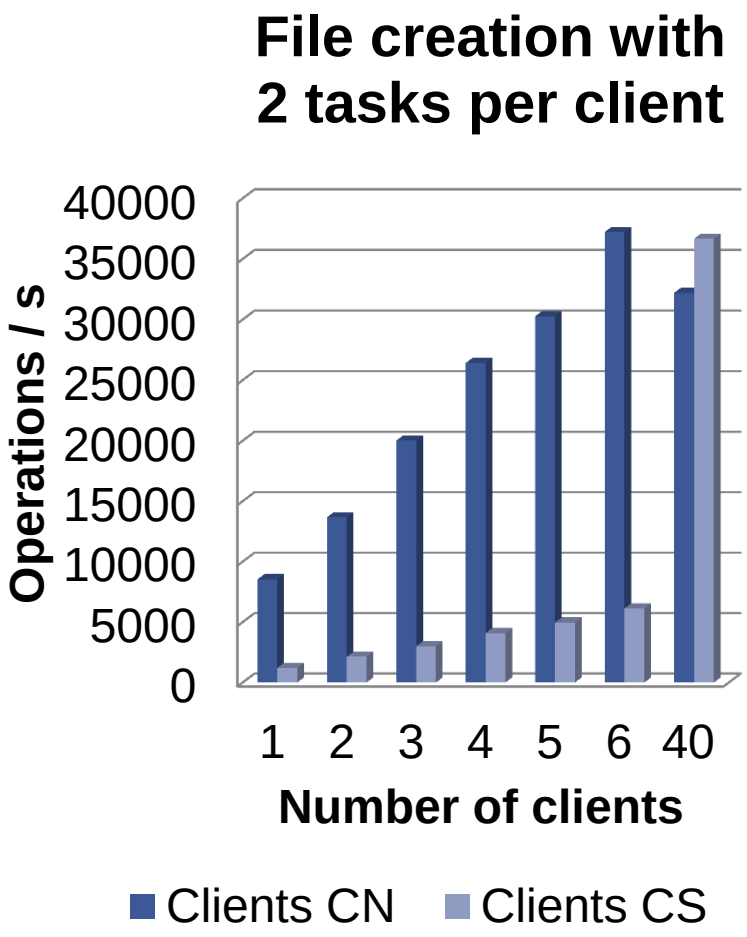


# What is special with our tier 2 / tier 3 systems?

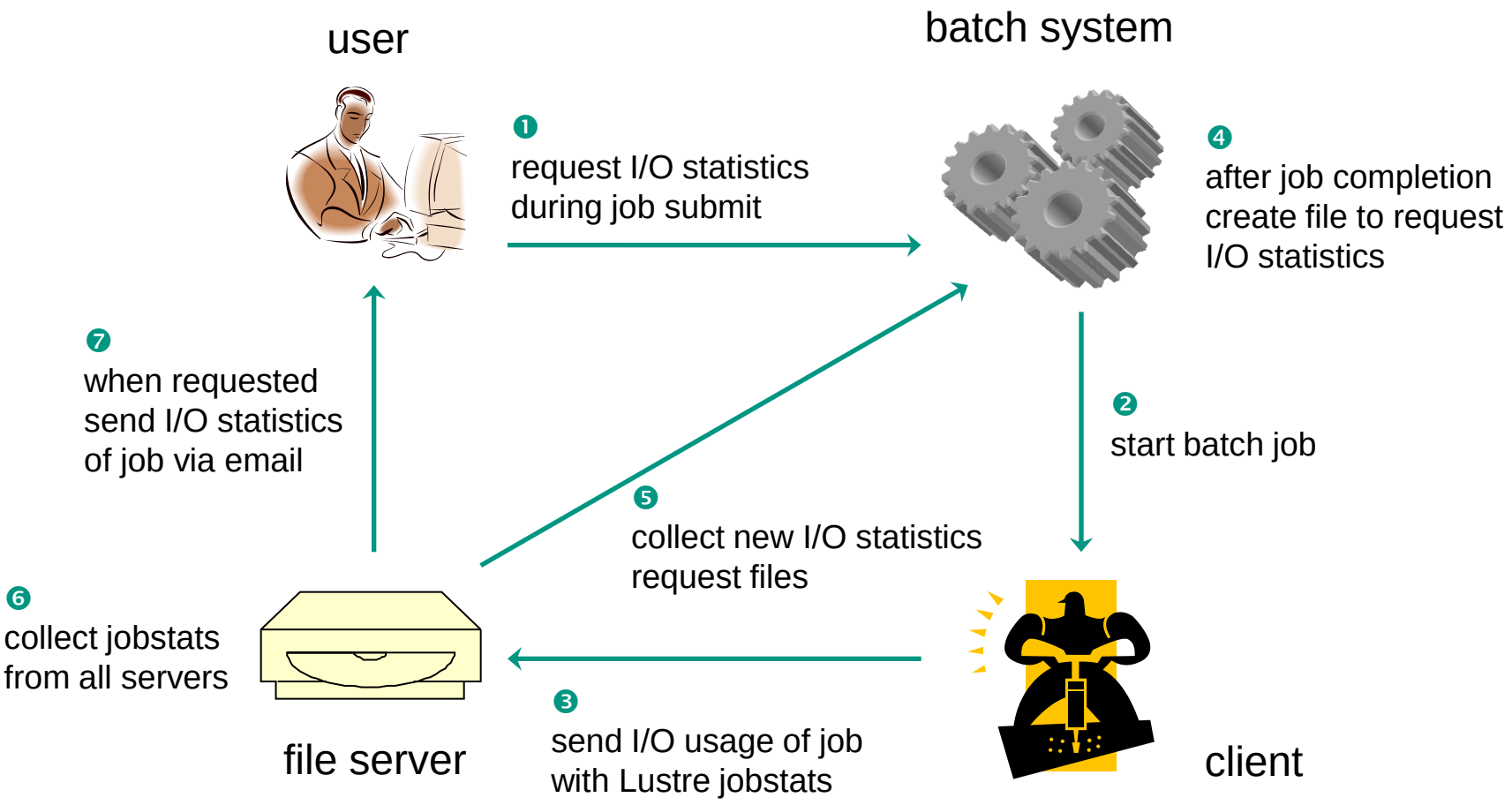
- Scalability of applications
  - Still lots of applications which only scale up to 10s of nodes
  - Higher level I/O libraries (HDF5, …) rarely used
  - File per process is beneficial
    - Usually less than 10K cores, file system can quickly handle 10 K files
    - Omits locking conflicts with writes from many clients to single file
- User community
  - Hundreds active users and running batch jobs
    - Many students and employees from 9 Baden-Württemberg universities
  - Many less experienced users
    - No experience with Linux
    - Just use an existing program (Matlab, …)
  - Result is a lot of bad I/O

# Using Lustre over a 30 km IB connection

- 320 Gbit/s delivered by
  8 Mellanox MetroX IB switches
  - Up to now no downtime due to long distance connection
- Feels like working locally
  - No reduction in throughput performance
- Some metadata operations loose factor 3
  - See diagram on right side
  - With many clients delay on server is dominating



# Lightweight I/O statistics – diagram



# Lightweight I/O statistics – example email

Subject: Lustre stats of your job 1141 on cluster xyz

Hello,

this is the Lustre IO statistics as requested by user john_doe
on cluster xyz for file system home.

Job 1141 has done ...
... 1 open operations.
... 1 close operations.
... 1 punch operations.
... 1 setattr operations.
... 10 write operations and sum of 10,485,760 byte writes (min IO size: 1048576, max IO size: 1048576).

# Lightweight I/O statistics – experiences

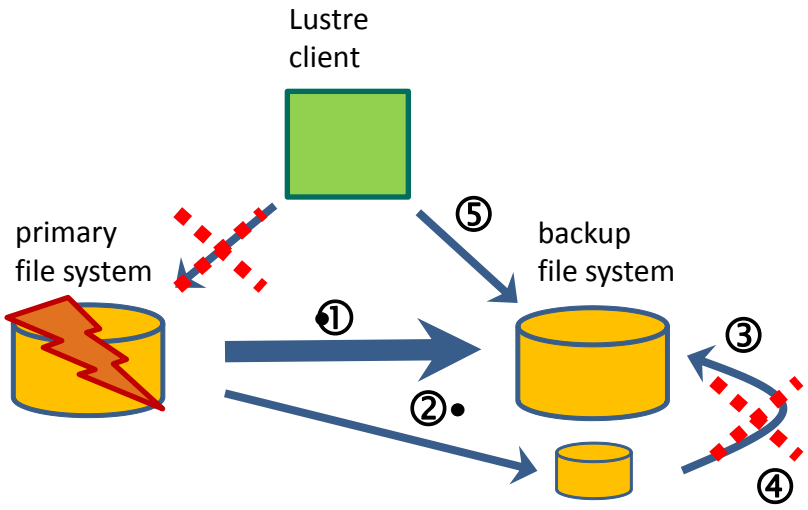
- No negative impact of jobstats activation
  - Running since 2 years
- Users do not care much about their I/O usage
  - Tool is not frequently used
  - Much better: Actively alert users about bad I/O usage
- Run another perl script to find jobs with high I/O usage
  - Collects and summarizes jobstats from all servers
  - Reports job IDs over high water mark for read/write or metadata operations
  - Extremely useful for administrator to identify bad file system usage
- Download our perl scripts to get Lustre I/O statistics
  - See jobstats chapters (6.2.7 and 6.2.8) at http://wiki.lustre.org/Lustre_Monitoring_and_Statistics_Guide

# Disaster recovery – problem statement

- A disaster can be caused by
  - hardware failure, e.g. a triple disk failure on RAID6
  - silent data corruption caused by hardware, firmware or software
  - complete infrastructure loss, e.g. caused by fire or flood
- Timely restore of 100s TB does not work
  - Transfer takes too long and rates are lower than expected
    - Bottlenecks often in network or at backup system
  - Metadata recreation rates can be limiting factor
  - We restored a 70 TB Lustre file system with 60 million files
    - With old hardware and IBM TSM this took 3 weeks
- Users should separate permanent and scratch data
  - Backup and disaster recovery only done for permanent data

# Disaster recovery – steps

- Idea: Use tool rsnapshot to create backup on other file system and change client mount point after disaster
  - rsnapshot uses rsync and hard links to create multiple copies




Note: Data created after last good rsync is lost.

Backup:

1. Use rsnapshot (rsync) to transfer all data to backup file system
2. Use rsnapshot (rsync + hard links) to transfer new data
3. rsnapshot removes old copies

Disaster recovery:

4. Use good rsnapshot copy and move directories to desired location
5. Adapt mount configuration and reboot Lustre clients

# Disaster recovery – experiences, restrictions

- Experiences
  - Backup done twice per week on one client with 4 parallel processes
    - For 100 mill. files and with 5 TB snapshot data this takes 26 hours
  - Disaster recovery needed for first time in January 2017
    - RAID controller on MDS delivered different data when reading twice
    - According to support reason was firmware bug
    - File system check was not able to repair, investigation took 4 days
    - After switching to backup file system everything worked as expected
    - Maintenance to switch back to newly created file system took 1 day
- Restrictions
  - Slow silent data corruption might pollute all backup data
    - Same problem for other backup solutions
  - Recovery does not work if both file systems have critical Lustre bug

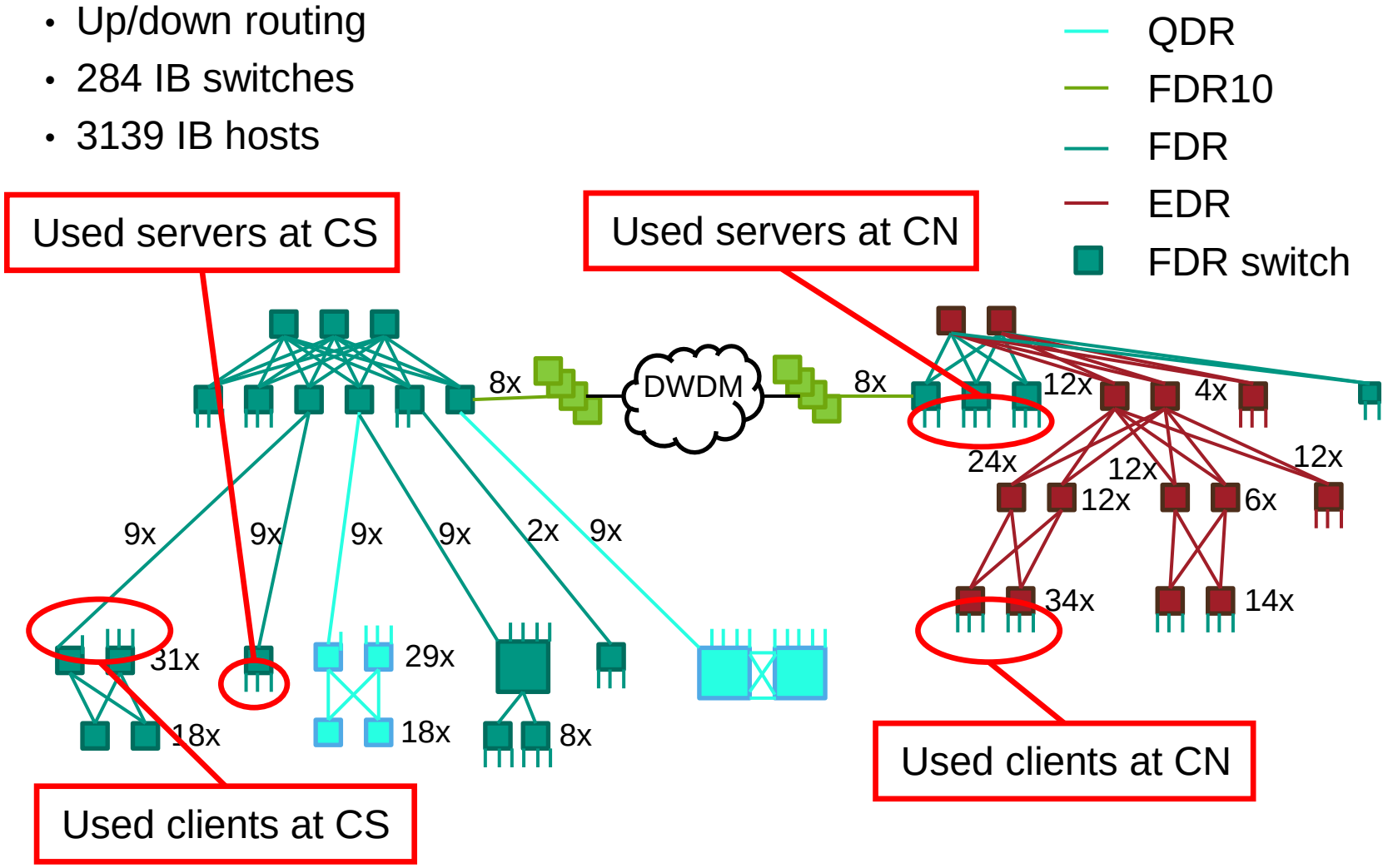
# Summary

- Extraordinary HPC file system solutions at KIT
  - File systems mounted on many clusters over complex IB network
  - File systems used by diverse user community and applications
  - Lustre over 30 km IB connection works fine
  - Special disaster recovery solution
  - Own lightweight solution to provide Lustre I/O statistics
- All my talks about Lustre
  - http://www.scc.kit.edu/produkte/lustre.php
  - Check talks from LAD for more details on presented topics
- roland.laifer@kit.edu

# Backup slides

# Performance measurement details

- Up/down routing
- 284 IB switches
- 3139 IB hosts

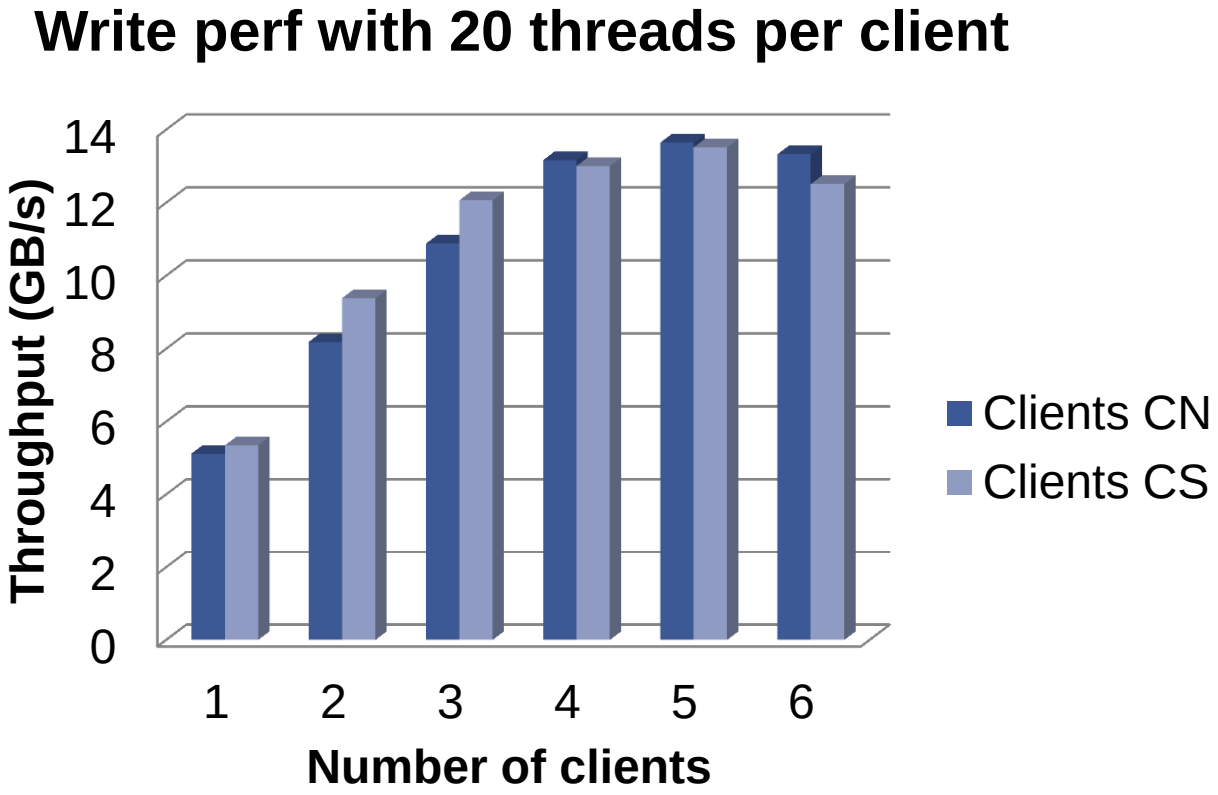


# Performance measurement details

- Done while some of the systems were in production
  - Just show trends, no focus on best performance
- Write performance measured with iozone
  - Options: -+m <file_name> -i 0 -+n -r 1024k -t <thread_count> -s 8g
- Metadata performance measured with mdtest
  - Options: -u -n 10000 -i 3 -p 10 -d <lustre_dir>
- Used clients
  - CN: RH7, Mellanox OFED, FDR Connect-IB, Exascaler 2.3
  - CS: RH6, RH OFED, FDR ConnectX-3, Exascaler 2.1
- Used file systems
  - CN: EF4024 (MDT), 28 OSTs on ES7K, 6 TB disks, Exascaler 2.3
  - CS: EF3015 (MDT), 40 OSTs on SFA12K, 3 TB disks, Exasc. 2.1

# Write performance to file system at CS

- Same performance from both sites



# Lightweight I/O statistics – steps in detail (1)

1) Enable jobstats for all file systems
   - on clients: lctl set_param jobid_var=SLURM_JOB_ID
     - Make sure clients have fix of LU-5179
     - Slurm job IDs are used by Lustre to collect I/O stats
   - On servers increase time for holding jobstats
     - E.g. to 1 hour: lctl set_param *.*.job_cleanup_interval=3600
2) User requests I/O statistics with Moab msub options:
   - -W lustrestats:<file system name>[,<file system name>]...
   - Optionally: -M <email address>
3) On job completion Moab creates files to request I/O stats
   - File name: lustrestat-<file system name>-<cluster name>-<job ID>
   - File content: account name and optionally email address

# Lightweight I/O statistics – steps in detail (2)

4) Perl script runs hourly on each file system
   - Uses different config file for each file system
     - Defines names of request files and of batch system servers
       - Allows to collect request files from different clusters
     - Defines which servers are used for the file system
   - Transfers files from batch systems and deletes remote files
     - Uses rsync and rrsync as restricted ssh command for login with key
   - Reads data including job IDs and account name
     - If not specified asks directory service to get email address of account
   - Collects and summarizes jobstats from all servers
   - For each job sends an email
     - Email is good since jobstats are collected asynchronously