CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Real-time Mission Critical Supercomputing with Cray Systems

Napa Valley, May 19 2013

Jason Temple and Luc Corbeil, CSCS

Introduction

- **MeteoSwiss' Context**
  - HPC Services
  - Client definition
  - Client needs
- **Design Considerations**
  - Partitions
  - Scheduler
  - Filesystems
  - Network

- **Sonexion**
- **Lustre issues**
  - Lustre Support
  - Silent Data Corruption
- **Implications**

HPC Services for MeteoSwiss

- **Maintenance of a 24/7 mission critical infrastructure within a research environmnent**
  - Leverage existing infrastructure where it makes sense
    - Mid-term storage and archiving
    - User environment (homes, etc.)
  - Put in place the required safeguards/failover mechanisms
    - Infrastructure, power and cooling
      - UPS
    - Hardware configuration
    - System configuration
    - Global systems monitoring (Nagios, Ganglia)
    - 24/7 on-call support (Pichetto and external inf. company)
  - Close collaboration between both organizations at all levels
    - Management
    - Operations

MeteoSwiss

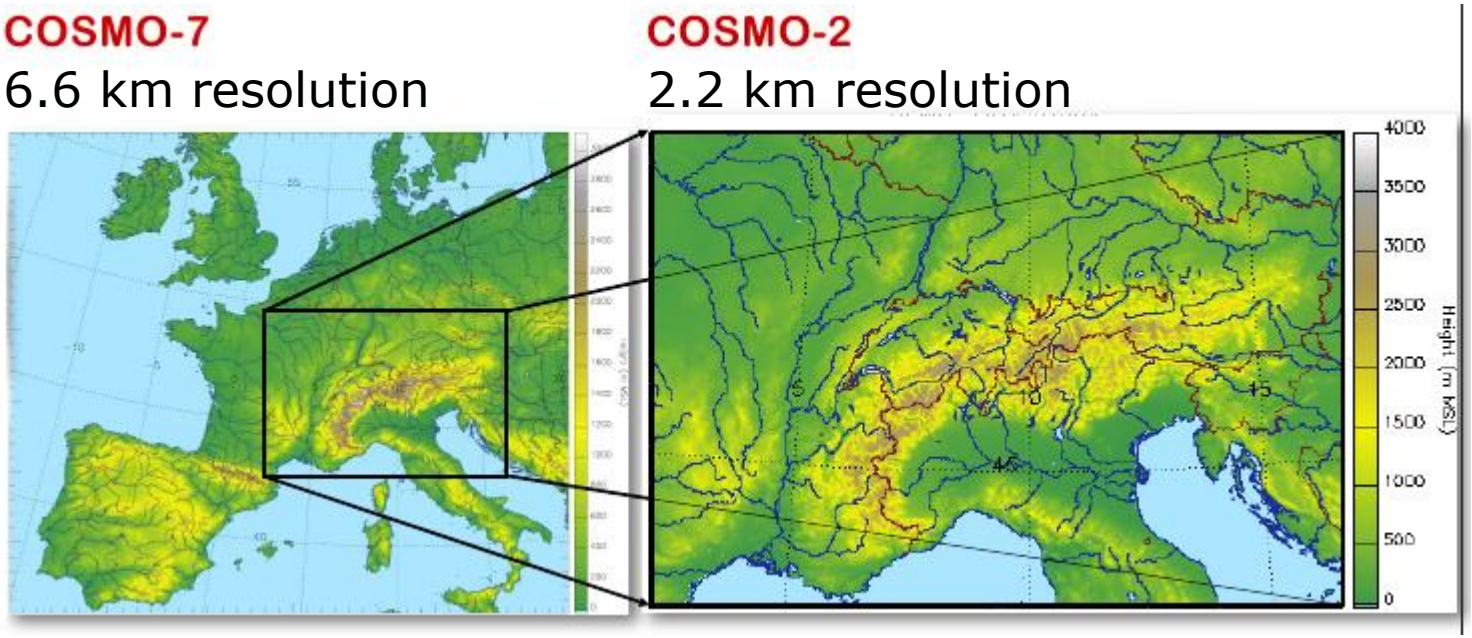
- "MeteoSwiss is the national weather and climate service for the Swiss public, for government, industry and science. With our public service we ensure the basic supply of weather and climate information in Switzerland."

- In addition, must provide on-demand monitoring for the Nuclear Regulatory Agency in the event of a nuclear incident somewhere in the world.

MeteoSwiss – some details

COSMO-7
6.6 km resolution

COSMO-2
2.2 km resolution



COSMO 2km: 8 times/day, within 25 minutes
COSMO 7km: 3 times/day, within 25 minutes

Correct results required to issue weather warnings in a timely manner

- Must be right the first time (unlike running linpack until success)

Two Distinct Partitions/One System

Albis

Cray XE6

5 Login Nodes
18 compute blades
-24 cores each
72 nodes
144 sockets
1728 cores
16.5 TFLOPS

Lema

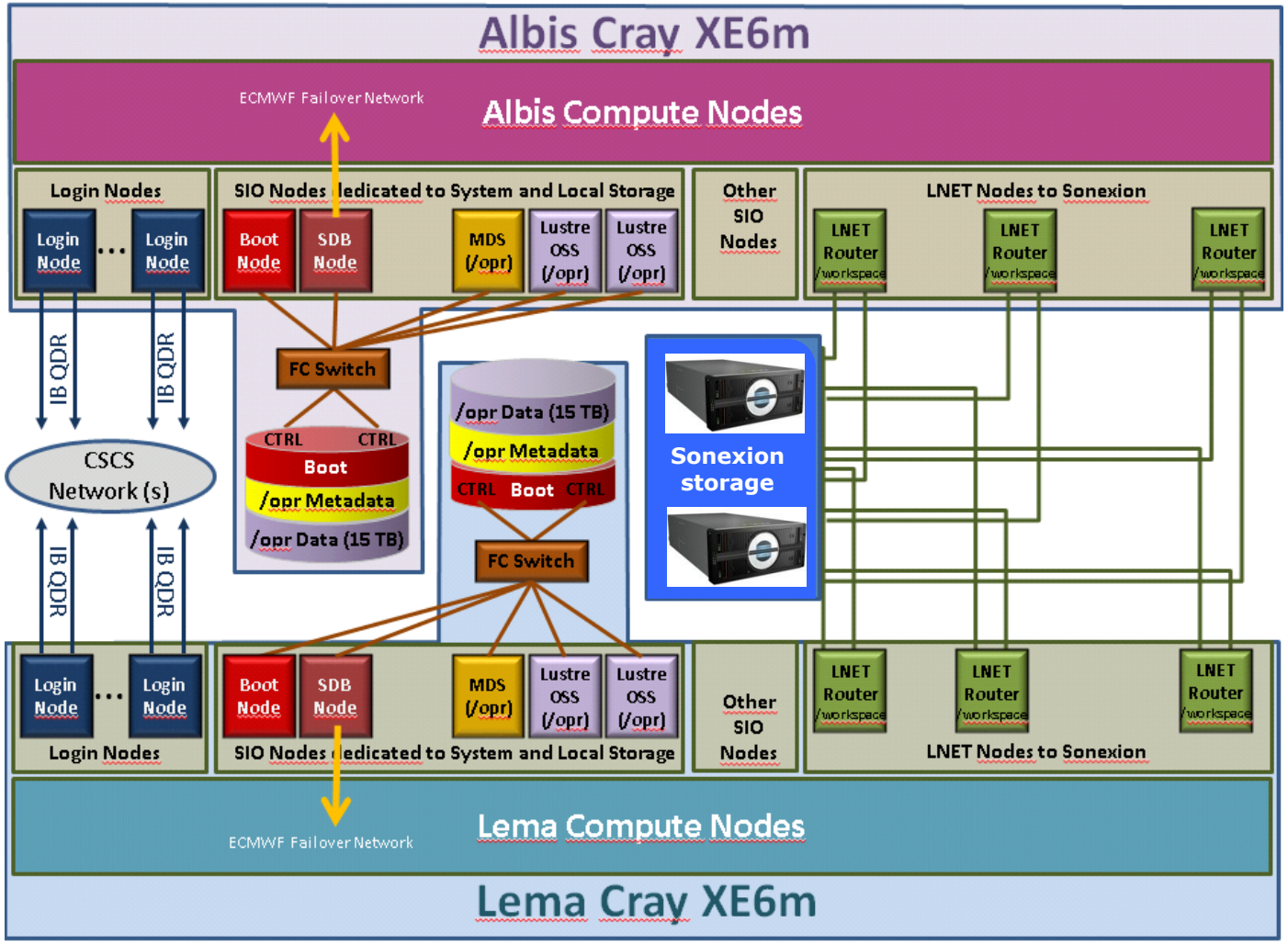
Cray XE6

5 Login Nodes
42 compute blades
-24 cores each
168 nodes
336 sockets
4032 cores
33.5 TFLOPS

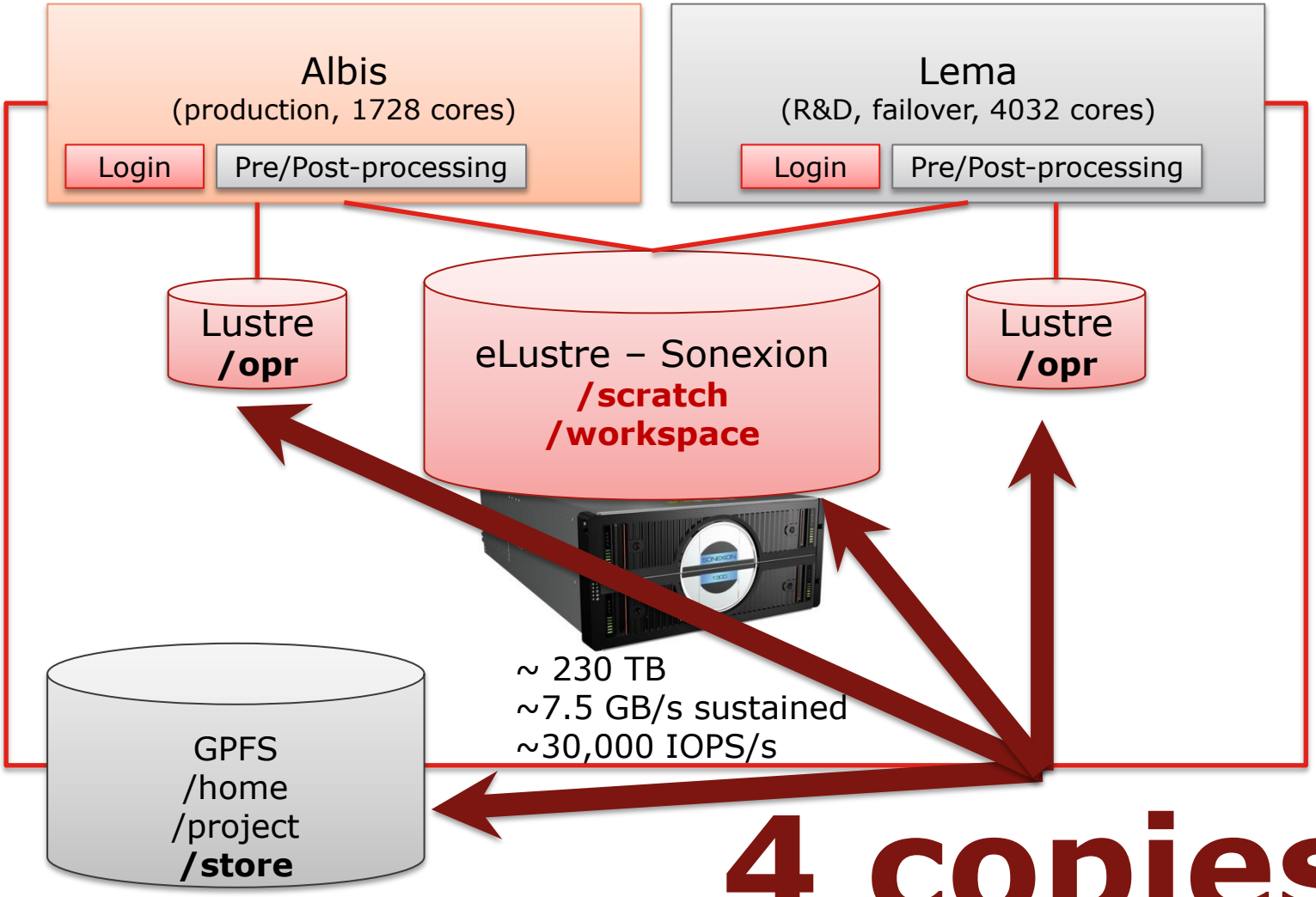
Cray XE6

- Two distinct systems (Albis/Lema)

Albis and Lema Complex Configuration



Albis and Lema Filesystem Configuration



SLURM for Compute and Pre/Post Proc Scheduling

SLURM
- «Simple Linux Utility for Resource Management»
  - Open Source from LLNL
  - Free
  - Very configurable, extensible

"Classic Cray Environment"

Compute Node SLURM
↓
Alps
↓
Compute Nodes

"Converted Pre/Post Environment"

Post Processing SLURM
↓
Direct Access to
Converted Compute Nodes
↓
Resource Control

Problems for Mission-Critical Supercomputing

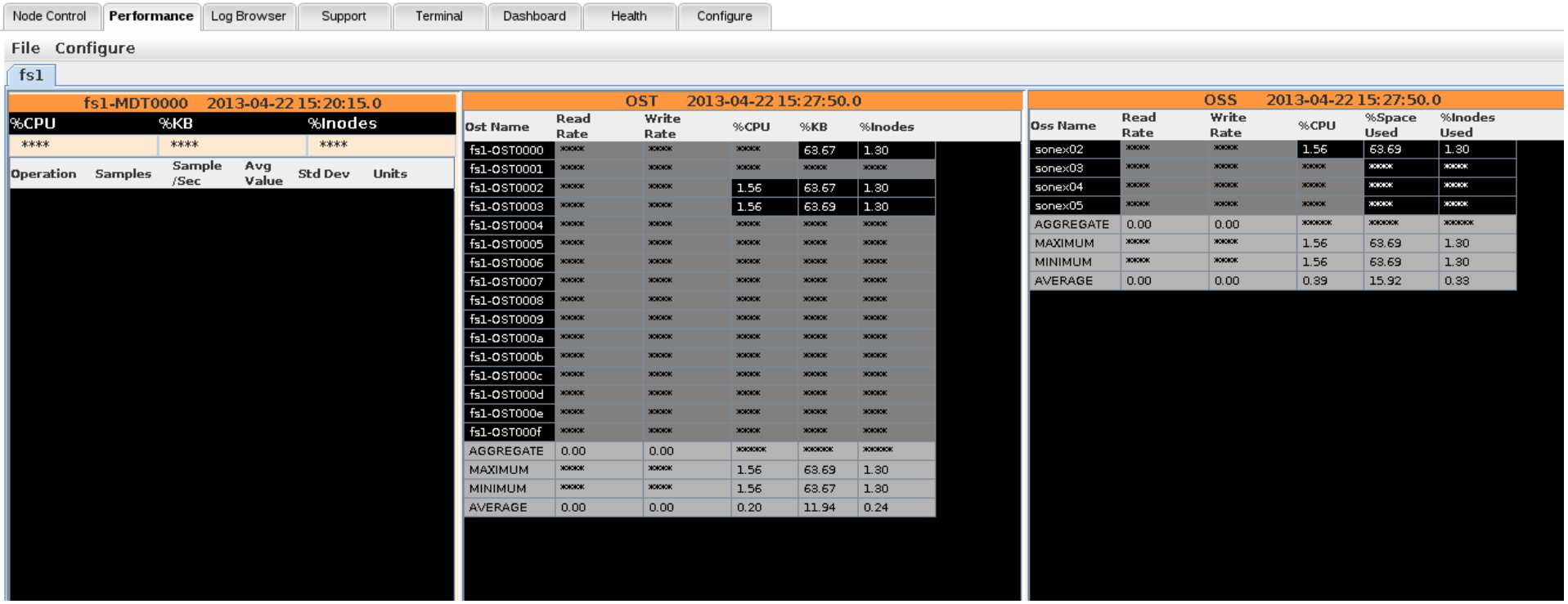
Problems with the Sonexion 1300

- Management GUI is not very useful in our version
- Incorrect installation
  - Failover patch not on all servers
- Substandard switch hardware installed (unmanaged switches)
  - Occasionally froze, needed rebooting
- Difficult to administer
  - No external ports
  - Puppet/certificate setup non-trivial
  - No «reliable» performance metrics
  - Basically a black box
- Apparent communication problems between Cray and Xyratex
- No «smooth» upgrade path between 1.0 -> 1.2.1
- **Silent Data Corruption** (not isolated to Sonexion, Lustre in general)

Problems with the Sonexion 1300

Hostname	Node Type	Power State	Mounted (1)	Targets (17)	HA Partner
sonex00	MGS	On	0	0	sonex01
sonex01	MDS	Unknown	1	1	sonex00
sonex02	OSS	On, Offline	0	4	sonex03
sonex03	OSS	On, Offline	0	4	sonex02
sonex04	OSS	On, Offline	0	4	sonex05
sonex05	OSS	On, Offline	0	4	sonex04

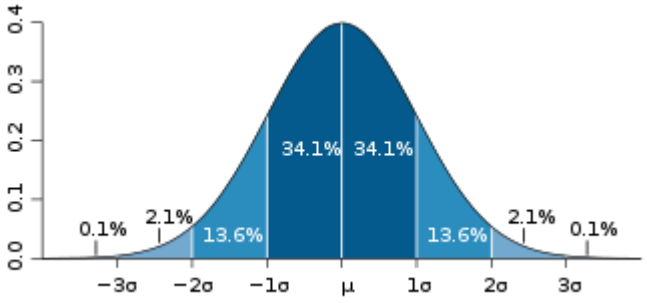
Problems with the Sonexion 1300



Reliability of Tools and Scientific Computing

How can you trust your scientific results if the tools you use are not 100% reliable?

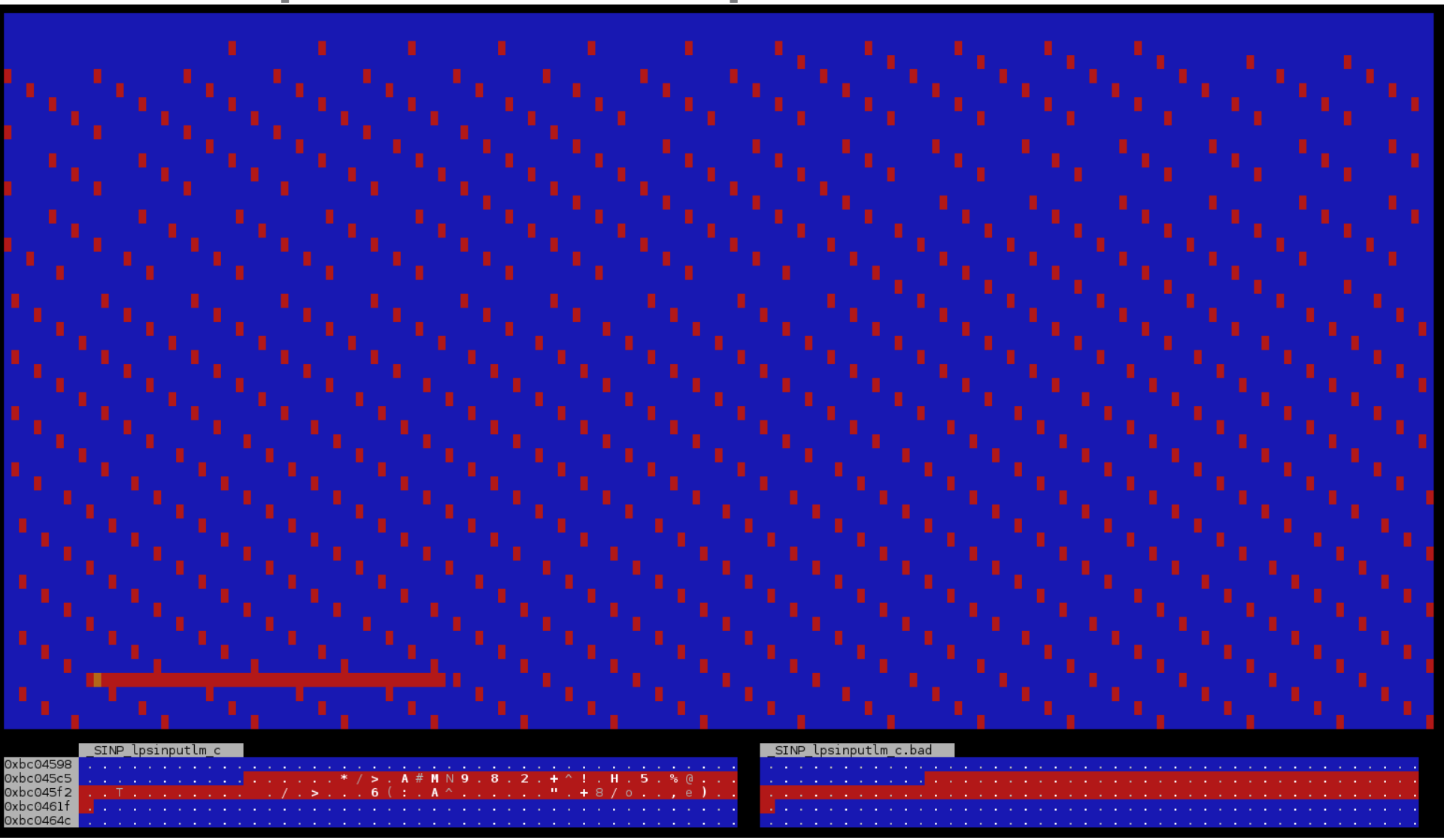
Do you make many runs, then choose a 95% confidence level from the normal distribution?



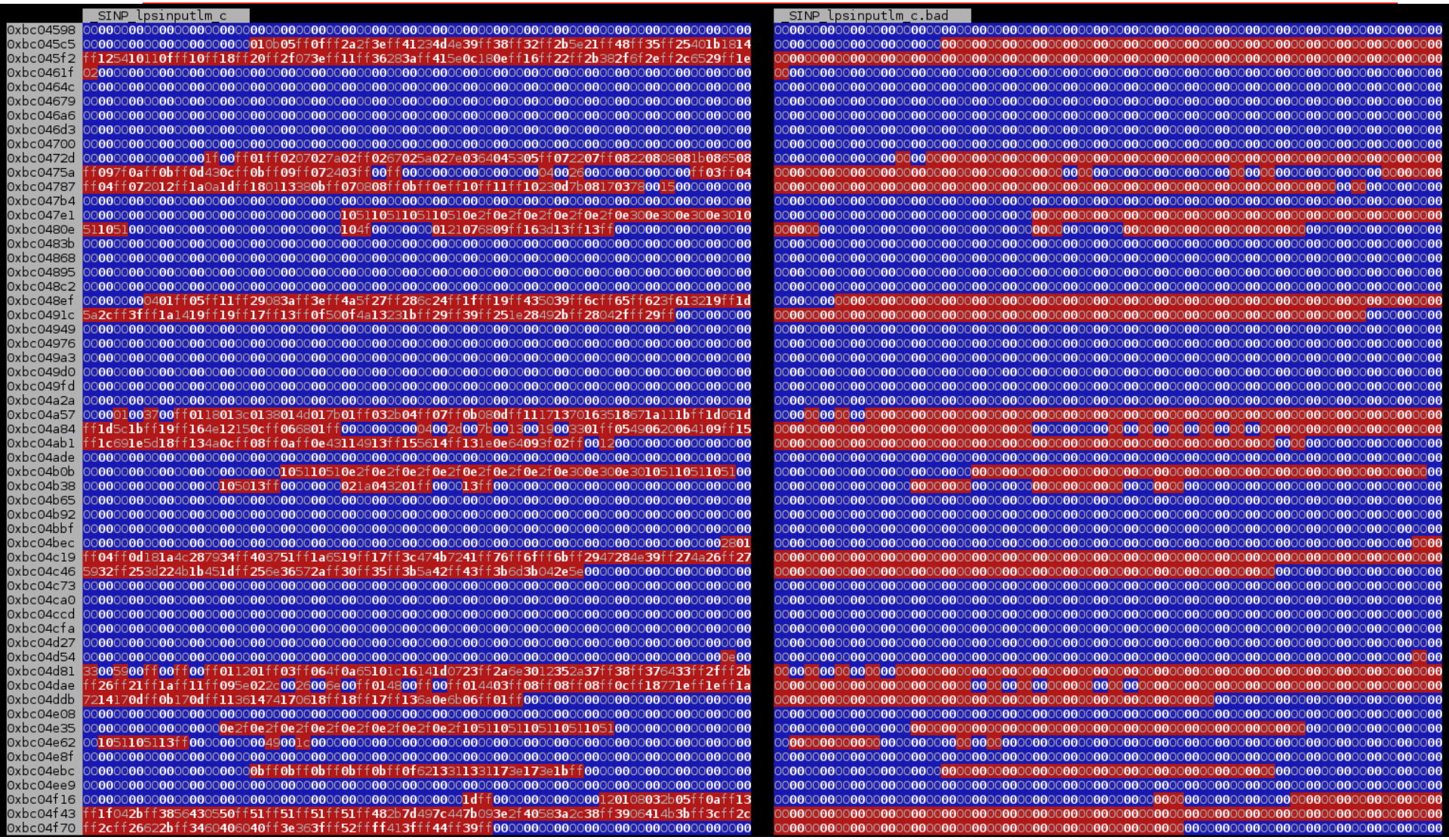
Silent Data Corruption

- After going into production, MeteoSwiss started to experience data corruption
- Absolutely silent in the Lustre logs, at any level
- 3 different types of corruption
  - Zero size files resulting from a simple untar of text files
  - Corrupted data in the middle of files, either zeroes or random
  - Truncated files.
- Random occurrences in random types of files
- Caused MeteoSwiss to send corrupted product output files to their clients
- Problem lasted more than 10 months
  - First reported in June ‘12
  - Cray got involved in August
  - Still corruption in February ‘13

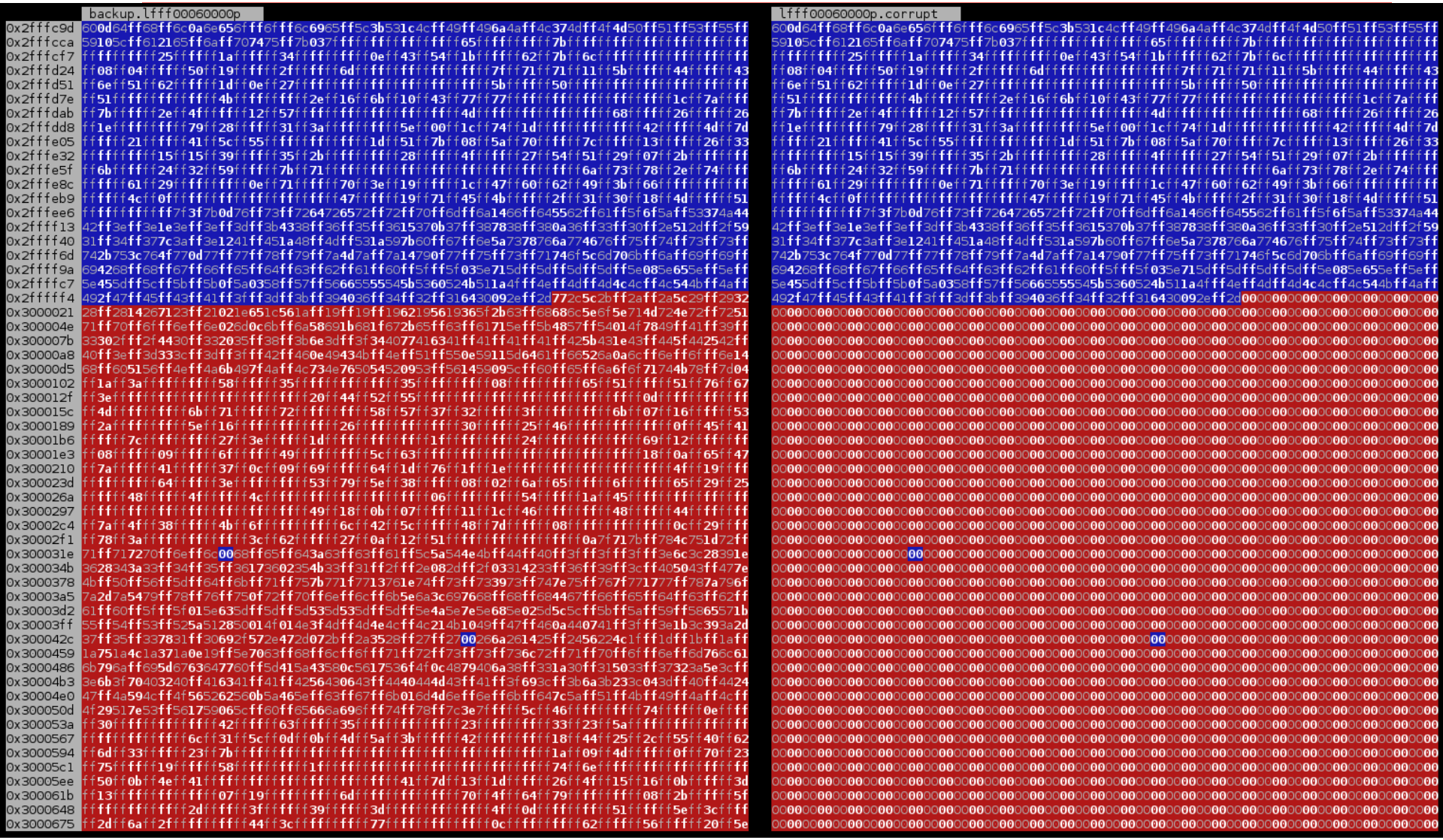
Examples of Data Corruption



Examples of Data Corruption



Examples of Data Corruption



Difficulties Capturing the Problem

- The fact that it was silent and random made it almost impossible to troubleshoot.
- Was not easily reproduceable, therefore, not easy to capture.
- CSCS managed to reproduce the zero-size file by untar issue one time after over 50,000 attempts, but nothing was seen in the logs
- MeteoSwiss was forced to fsync() almost every write operation in an attempt to flush the cache
  - No discernable effect, other than slowing down I/O
- Most Vexing: Happened on internal Lustre, as well as on the Sonexion!!!
  - Lustre versions 1.8.x and 2.0 (Sonexion 1300)
- Despite CSCS' & Cray's efforts, no serious progress on the case
- What next?

Using the Lustre Mailing Lists

- The Lustre mailing lists are a fantastic resource for people using Lustre
  - Very quick response time from experienced Lustre engineers (Andreas Dilger, now with Intel, is the most prominent)
- At our wits' end, a question describing out data corruption issue was sent to the mailing lists.
  - Almost immediately, we received a response from another Cray user that had experienced almost the exact same problems, with links to lustre bug reports
- This email coincides with sudden renewed interest on the part of Cray
- Weekly con-calls were implemented in order to corner the problem

Finally a Solution

- After more than 10 months of silent data corruption, Cray fast-tracked some more-than-year-old Lustre patches:

  (from the patch readme files)
  – Handle network errors during bulk I/O.
  – Lookup returns wrong inode following rename by another client
  – Modify LND message send/recv rx timeout policy

- As of today, more than 2 months later, there have been no further incidents of corruption

Implications

- When a company freezes or forks Lustre, it **freezes** it
  - «Slow» access to recent bug fixes
  - «Even slower» access to recent developements
    - E.g. HA failover
    - Our bugs
- Public mailing lists: to post or not to post?
  - CSCS primary duty is to protect MeteoSwiss operations
- Other centers around the world can be impacted by these problems
  - How many systems are sold to this day without these patches?
  - How are customers supposed to know?
- Lustre is always advertised as **scratch space**
  - Implying «don't trust it, it can be lost at any time», but it must still provide data integrity – «fast vs reliable»

Example

Are **you** producing results with data corruption?

And Real-Time Mission Critical Supercomputing?

- Significant and respected scientific results are produced using Lustre
- For Real-Time operations, it must work the first time. All the time.
- No parallel filesystem is 100% reliable
  - But supportability is key, so issues are quickly addressed
  - The breach of trust occurs once the first byte of data is lost
- Sites must be made aware of major filesystem issues and be given the opportunity to mitigate
  - And reformatting the filesystem is not a viable upgrade path

How can we make this better?

- For CSCS:
  - Acceptance
    - Run the entire suite (not IOR)
  - Work with Cray to standardize bug reporting
  - Consider lobbying within OpenSFS to prioritize supportability

- For Cray:
  - Field Notices for critical issues
    - Admitting knowledge of a bug to clients is not a weakness
  - Consider lobbying within OpenSFS to prioritize supportability
  - Back-porting essential
  - Closer collaboration with Lustre entities (i.e. Xyratex)

Thank you for your attention.