Operational Experience Running HDFS XRootD Fallback

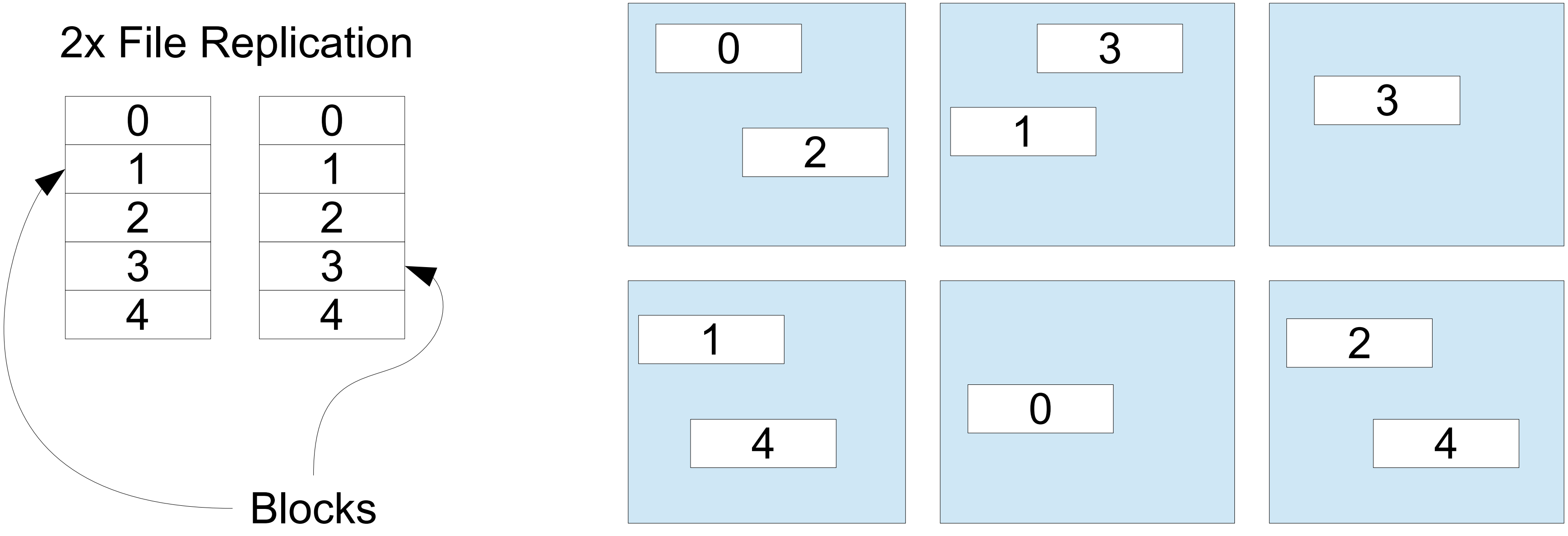
Jeff Dost, Alja Tadel, Matevz Tadel, Frank Wuerthwein
UC San Diego

Hadoop XRootD Fallback Project

- The Hadoop XRootD Fallback Project allows a site to offload the overhead of storage redundancy from their local system onto the redundancy provided by the XRootD Federation
- The software was developed at UCSD over the course of the past 2 years

Hadoop Distributed File System

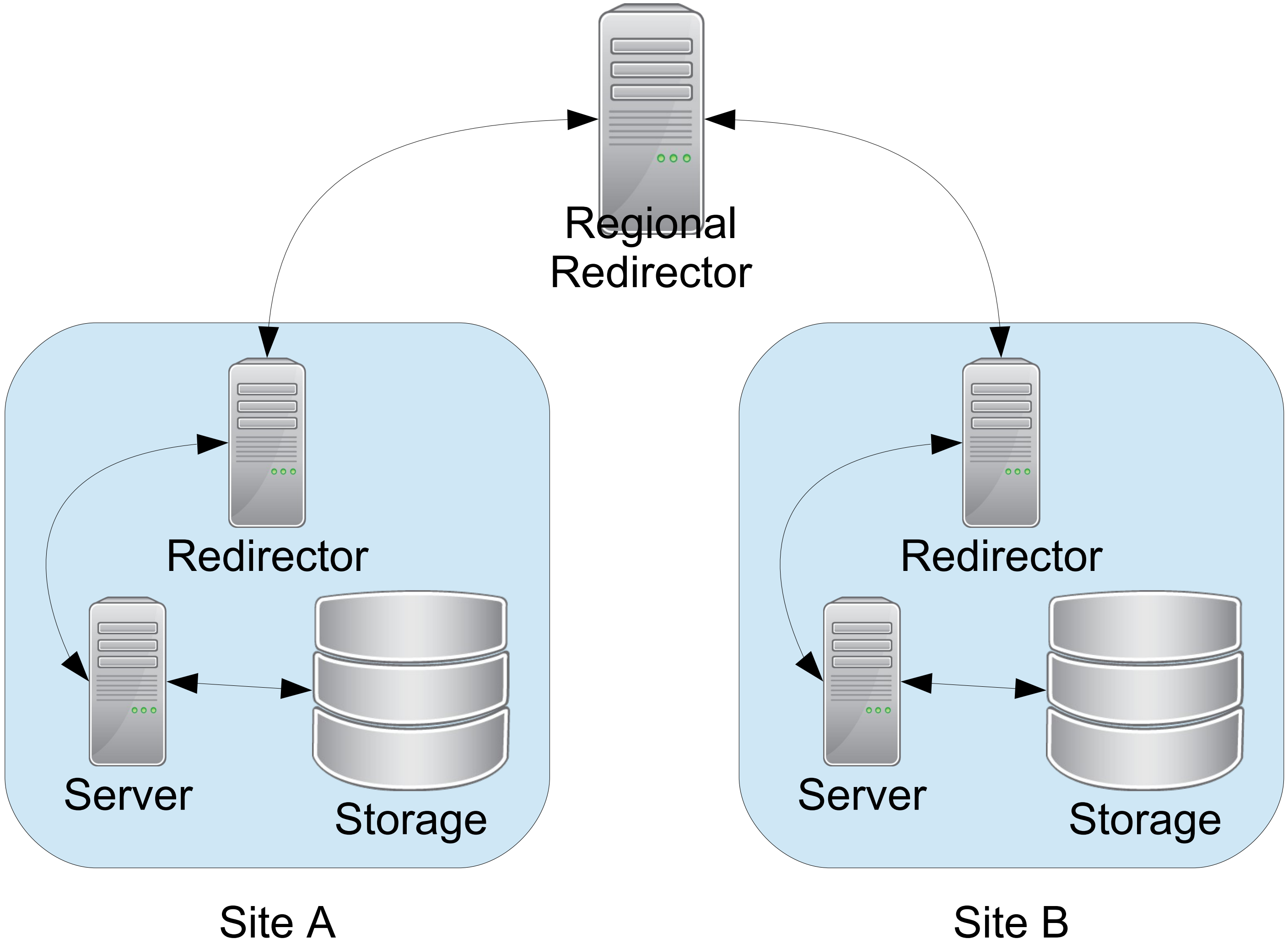
2x File Replication



Blocks

Data Nodes

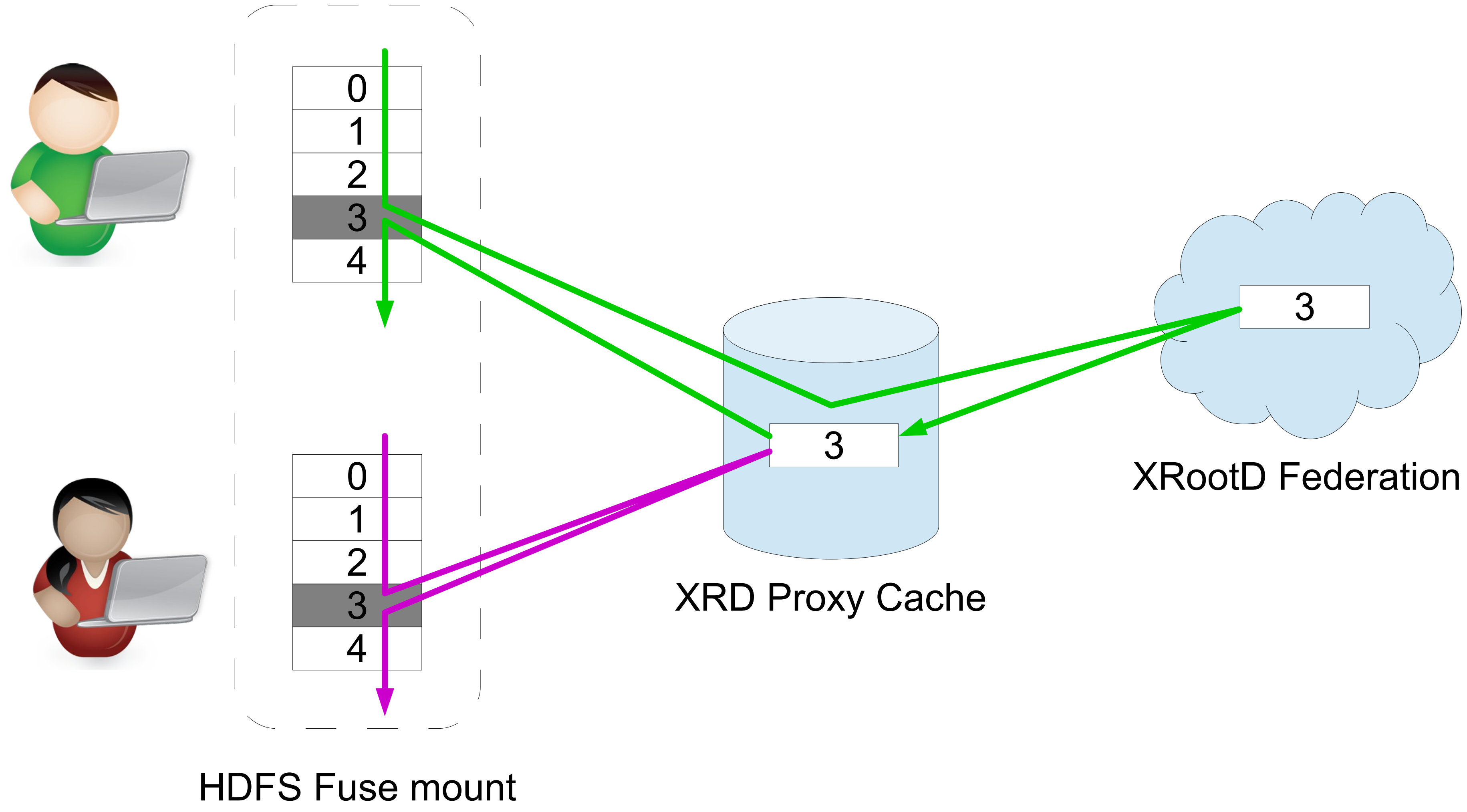
XRootD Federation



Prerequisites

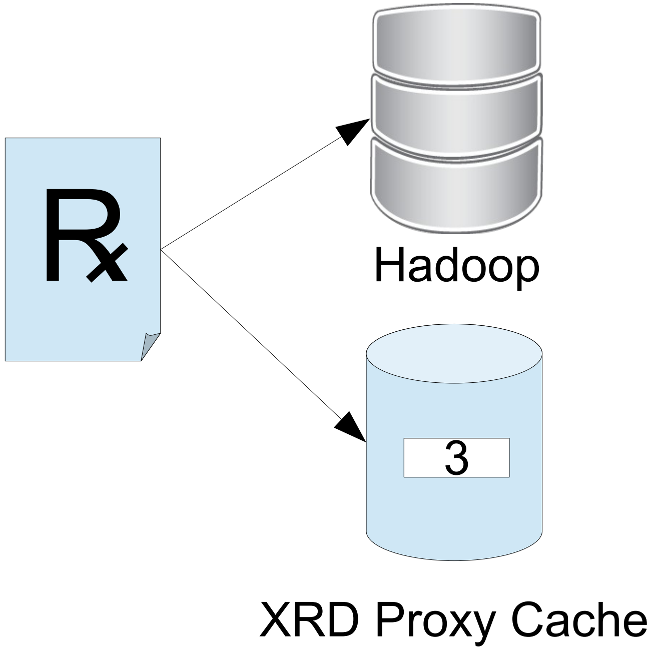
- Site uses Hadoop as its local storage backend
- Site belongs to an XRootD Federation
  - Other sites in participating federation do not need to be running HDFS!
- Data is guaranteed to be replicated on at least one other site in the Federation

HDFS XRootD Fallback

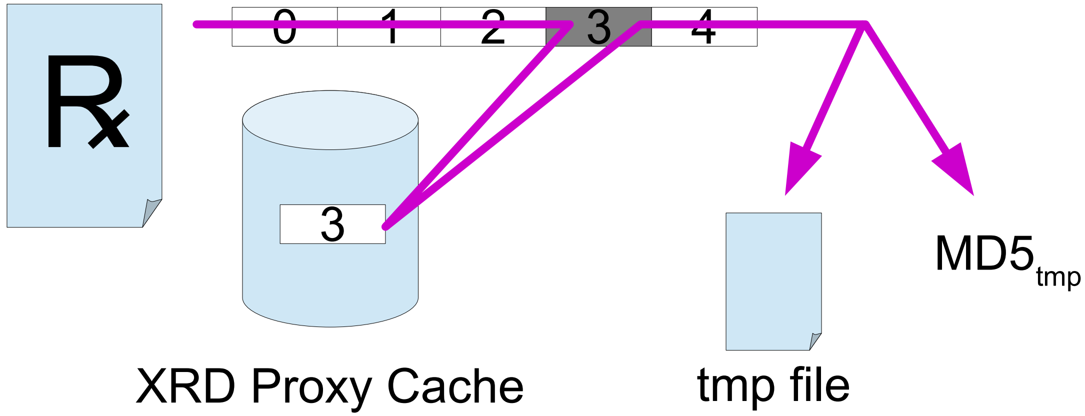


HDFS XRootD Healer (v2)

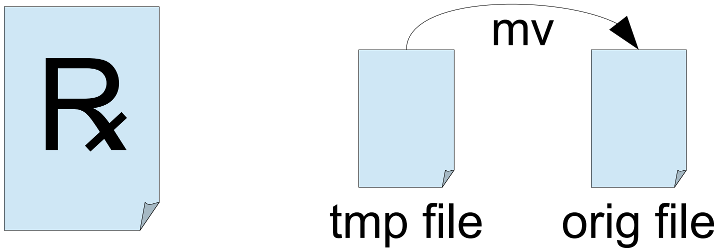
1. Obtain list of broken files that have been accessed:



2. For each broken file:



3. If $MD5_{tmp}$ == $MD5_{orig}$:



UCSDT2 Deployment

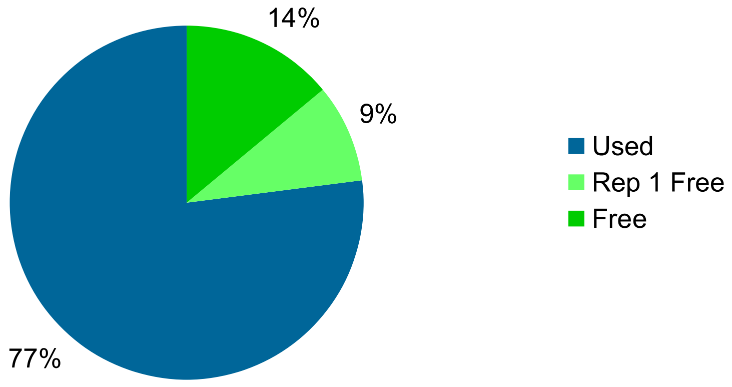
- Apr 2014 – hdfs-xrootd-fallback put into production
- Aug 2014 – reduced replication factor to 1 for subset of CMS data:

  /cms/phedex/store/data/Run2012A
  /cms/phedex/store/data/Run2012B
  /cms/phedex/store/data/Run2012C
  /cms/phedex/store/data/Run2012D
  /cms/phedex/store/data/Fall13
  /cms/phedex/store/data/Summer13
  /cms/phedex/store/mc/Summer12_DR53X
  /cms/phedex/store/himc
  /cms/phedex/store/relval

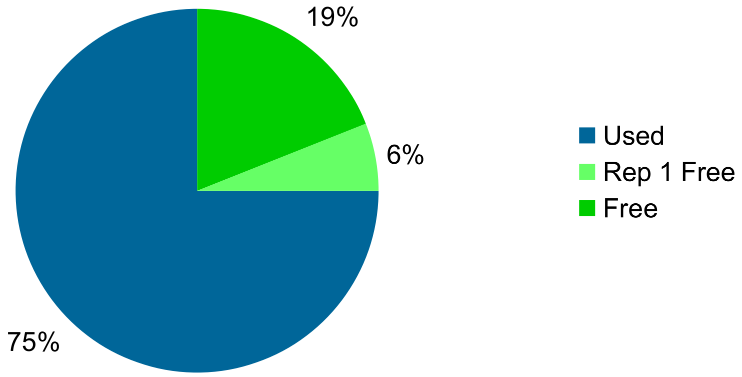
- Sep 2014 – hdfs-xrootd-healer (v1) put into production
- Mar 2015 – upgraded hdfs-xrootd-healer to v2

Oct 2014 vs Apr 2015

- Storage increased due to recovering 11 dead nodes (not fallback related)



Configured Capacity	2.55 PB
DFS Used	1.96 PB
DFS Used%	77 %
Live Nodes	101
Rep 1 Namespace	236 TB

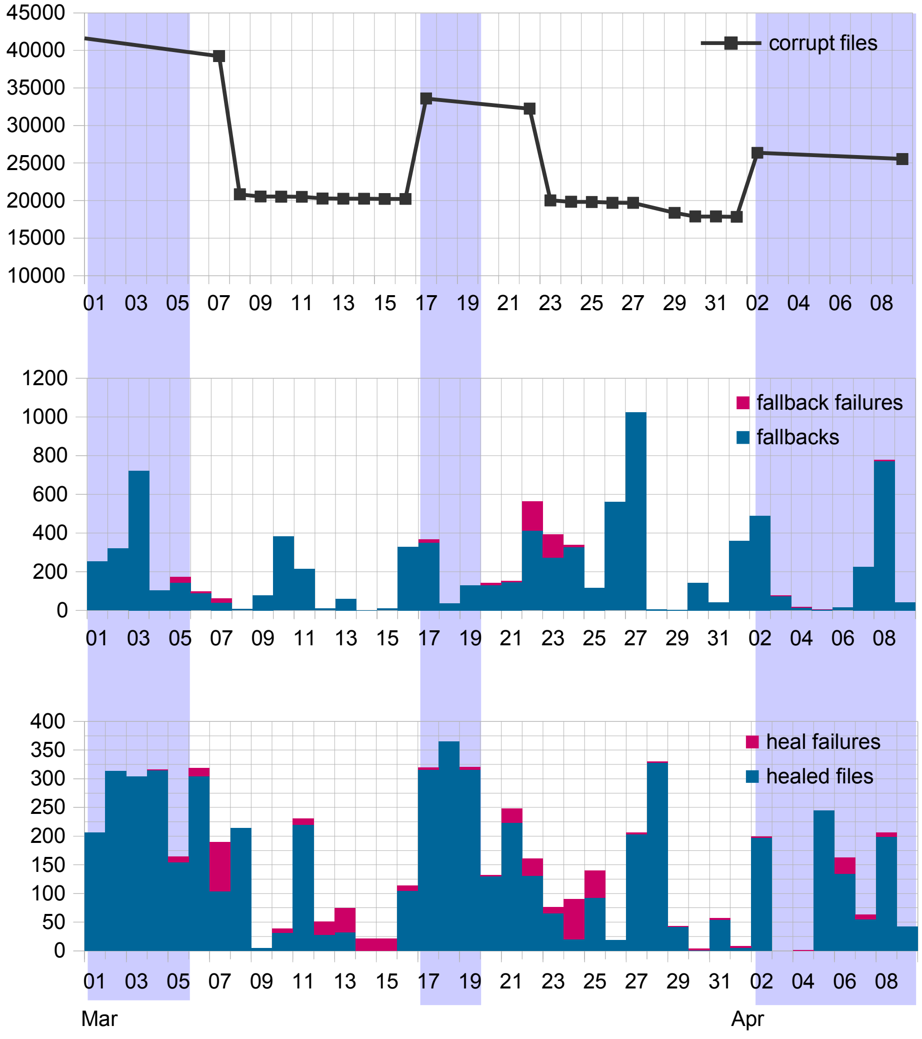


Configured Capacity	3.1 PB
DFS Used	2.32 PB
DFS Used%	75 %
Live Nodes	112
Rep 1 Namespace	190 TB

- DN recovery did reduce the impact of fallback, but it still is contributing a significant amount of space saved
- Unclear why replica 1 namespace shrunk by 46 TB, perhaps due to changes in CMS data popularity?

Fallback / Healer in Action

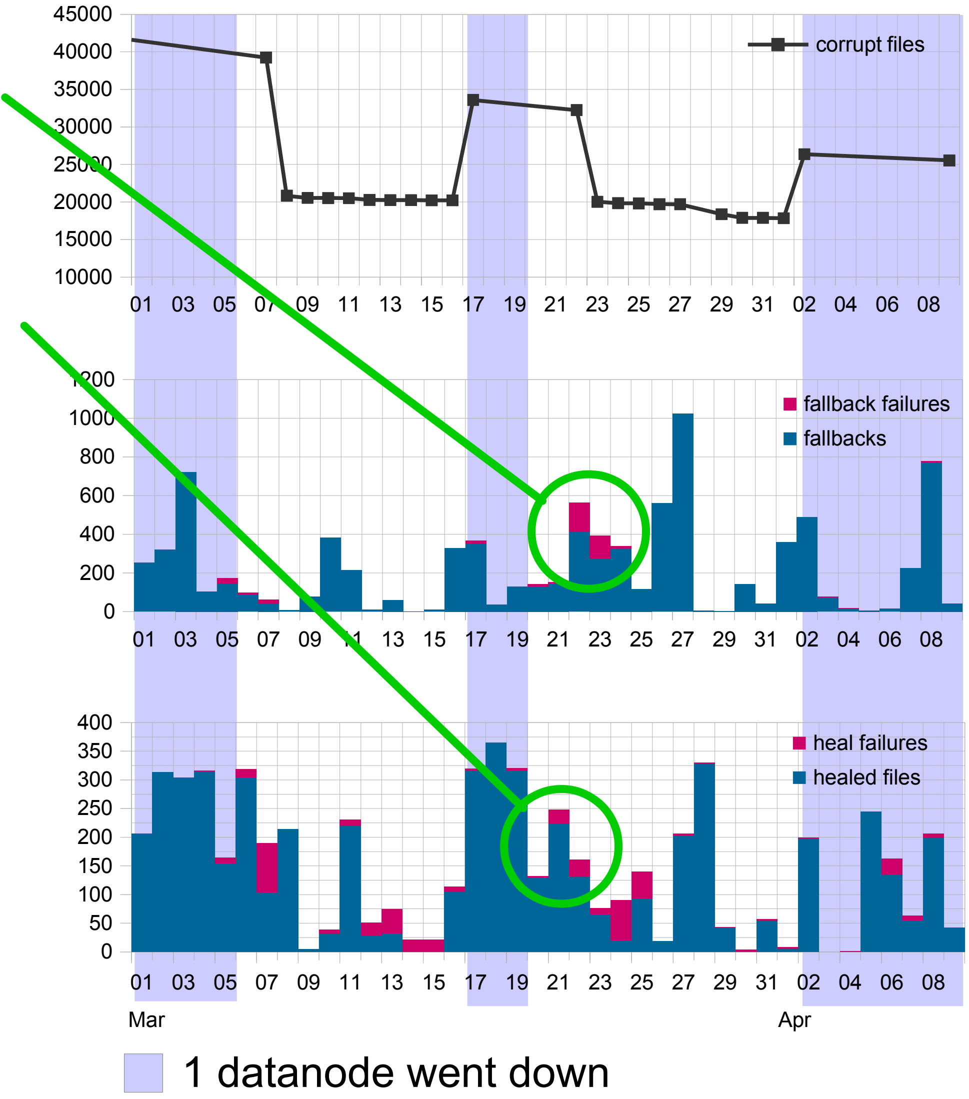
- Expected observations:
  - Total number of corrupt files decreases over time
  - File healing tends to increase after increased fallback triggers
- Surprising observation:
  - Healing does not significantly increase on downed datanode
  - Healing maxes out at 350 files a day; lack of data points in top plot is due to healer running over multiple days



▨ 1 datanode went down

Understanding Failures

- Fallback failures mean user jobs are failing on read, we take these seriously!
  - We want to understand why healer fails, but it is not as critical
- Investigating high number of fallback failures in March revealed some XRootD redirectors are ignoring our flag to avoid fetching from UCSD, even though we know the file is broken here
- We put in a software patch on March 25th to handle this, and the errors cleared up



What's Next?

- First order of business is get a proper fix into XRootD to deal with the self redirection problem
  - This must be fixed on the server side, and meta-managers will need to upgrade once fix is put into XRootD (4.2 release)
- hdfs-xrootd-fallback project is already hosted in the development OSG RPM repo, although shouldn't be marked as production ready until the above is solved
- Lower prio action items:
  - Investigate why healer bottlenecks at fixing 350 files / day
  - Increase replica 1 namespace to save even more space

Summary

- The Hadoop XRootD Fallback system gives sites a way to potentially obtain big savings on storage
- However this does not come 100% cost free
- The site must consider that by using it, their local storage becomes much more dependent on the stability of the XRootD Federation, in order to function optimally
- However if more sites participate, it will help us provide feedback to XRootD team to continue to make Federated storage even more stable and robust