D0 data reprocessing on OSG

Presenter: Amber Boehnlein

Authors

Andrew Baranovski (Fermilab) for

B. Abbot, M. Diesburg, G. Garzoglio, T. Kurca, P. Mhashilkar

Outline

- Computing Task
- The Process
- Issues
- Summary

Data reprocessing

- Improved detector understanding and new algorithms require re-reprocessing of the raw detector data

- Input: 90Tb of detector data + 250 Tb in executables
- Output: 60 Tb of data in 500 CPU years
  - DZero did not have enough dedicated resources to complete the task in the target 3 months

- D0 requested OSG to provide 2000 CPU for 4 month.

OSG

- Distributed computing infrastructure for large-scale scientific research
  - About 80 facilities ranging from 10s to 1000s of nodes
    - Agreed to share computing cycles with OSG users
    - No required minimal network bandwidth connectivity
    - Non uniform configuration environment
  - Opportunistic usage model
    - Exact amount of resources at any time can not be guaranteed

SAMGrid

- SAM-Grid is an infrastructure that understands D0 processing needs and maps them into available resources (OSG)
  - Implements job to resource mappings
    - Both computing and storage
  - Uses SAM (Sequential Access via Metadata)
    - Automated management of storage elements
    - Metadata cataloguing
  - Job submission and job status tracking
  - Progress monitoring

Challenge

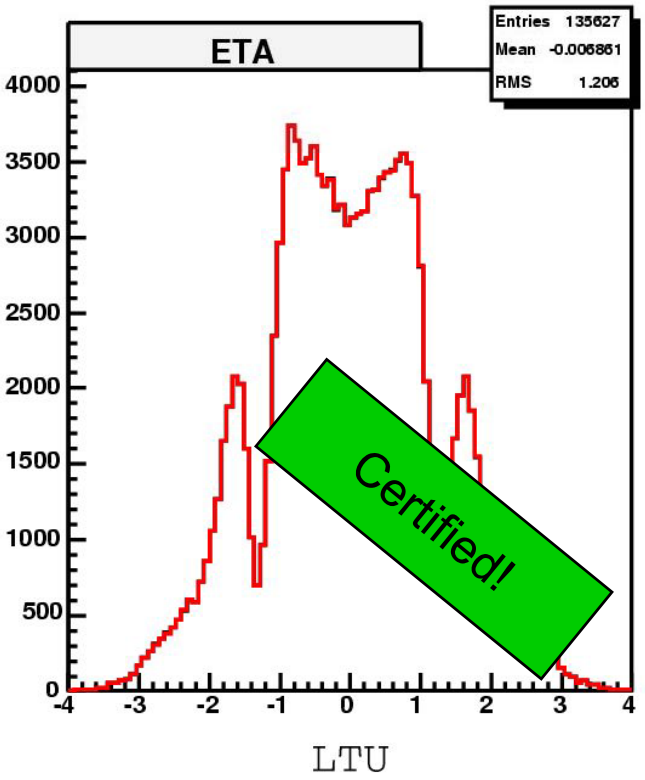
- Selecting OSG resources according to D0 requirements for availability, quality, and accessibility (of data)
  - Optimization work for SAMGrid
    - min (No of sites) && max (available CPUs) && max (IO bandwidth to CPU)
  - Work on the optimization as an iterative process to make the problem manageable

Process

- Add new resources iteratively
  - Certification
    - Physics output should be site invariant
  - Data accessibility
    - Tune data queues
    - Exclude sites/storages that do not provide sufficient data throughput
  - Enable monitoring views of site-specific issues

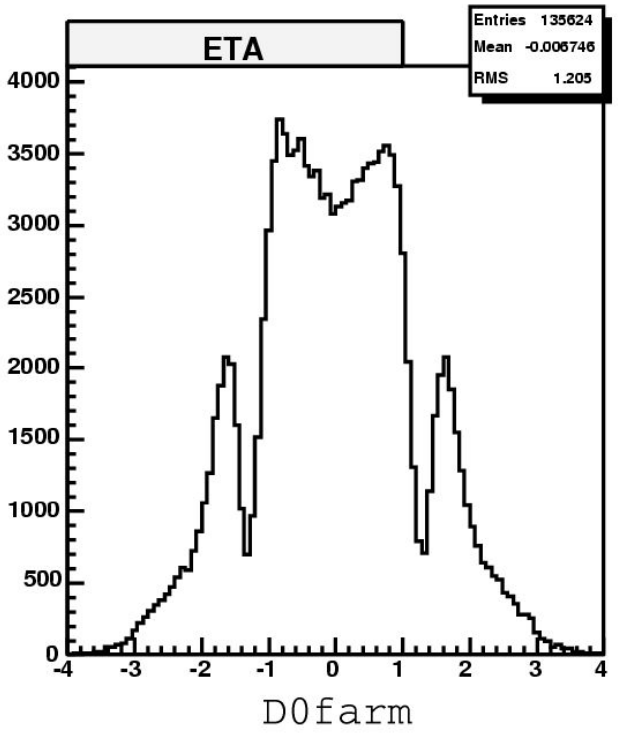
- Compare production at a new site with “standard” production at the D0 farm

OSG Cluster



“the same”

Reference



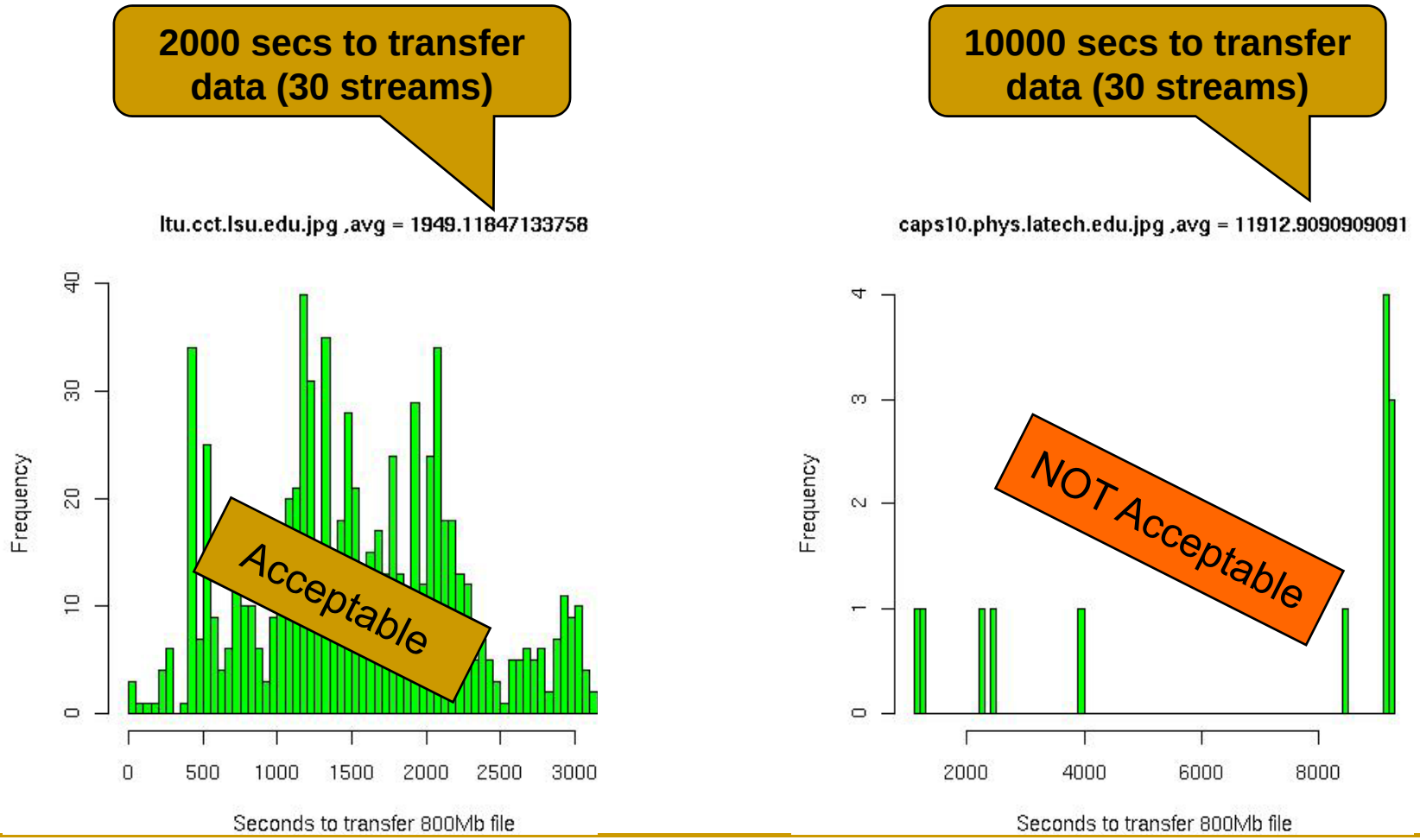
Note: Experienced problems during the certification on virtual OS.

*default random seed in python was set to the same value on all machines

Data accessibility

- Use dedicated data queues for each site and production activity type
  - If required. see below
- Monitor throughput of data in all directions
  - Decrease max number of data streams to high bandwidth / low latency sites
  - Increase max number of data streams to high bandwidth / high latency sites
  - Exclude sites that do not provide enough bandwidth

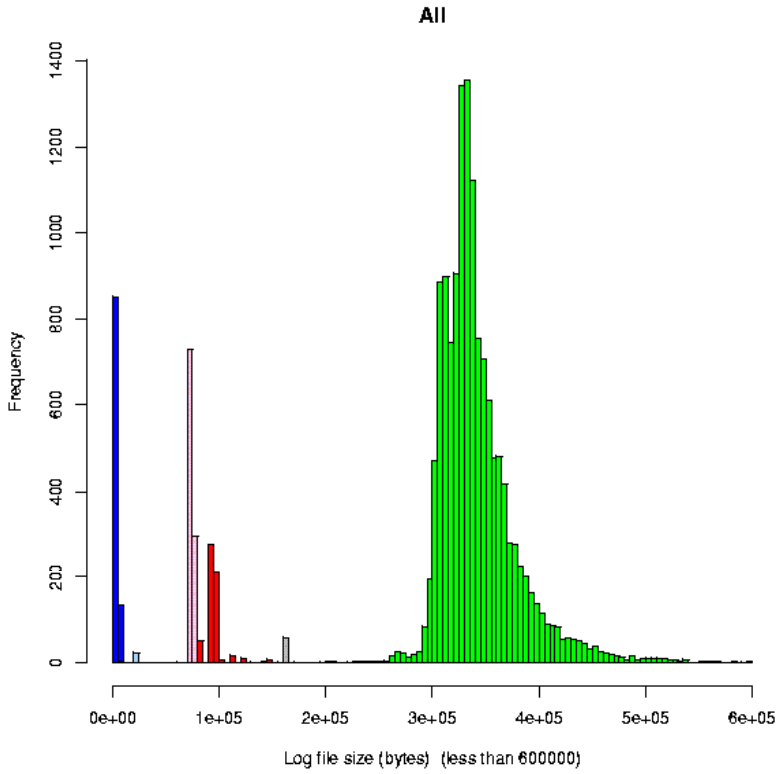
Data accessibility tests



Issue Tracking and Troubleshooting

- Support and tracking of site specific issues is an issue by itself
  - Dozen of sites involved at a time
  - Initially, few people available to troubleshoot and follow-up
    - Time critical activity as backlog of production quickly grows -> more trouble
- The OSG Grid Operation Center helped with the tracking and triaging of the issues

- We monitored log file size distribution to classify and prioritize current failures.
  - Failure free production generates log files that have a similar size



blue 0-8k Worker node incompatibility, Lost standard output, OSG no assign

aqua 8k-25k Forwarding node crash, service failure, could not start bootstrap executable.

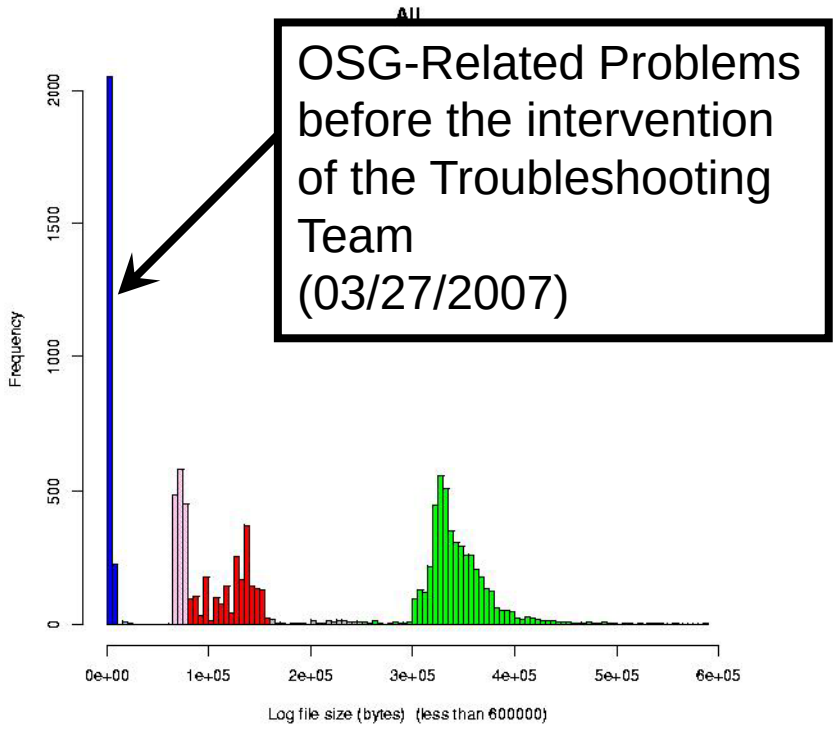
pink 25k-80k SAM problem. Could not get RTE, possibly raw files

red 80k-160k SAM problem. Could not get raw files, possibly RTE

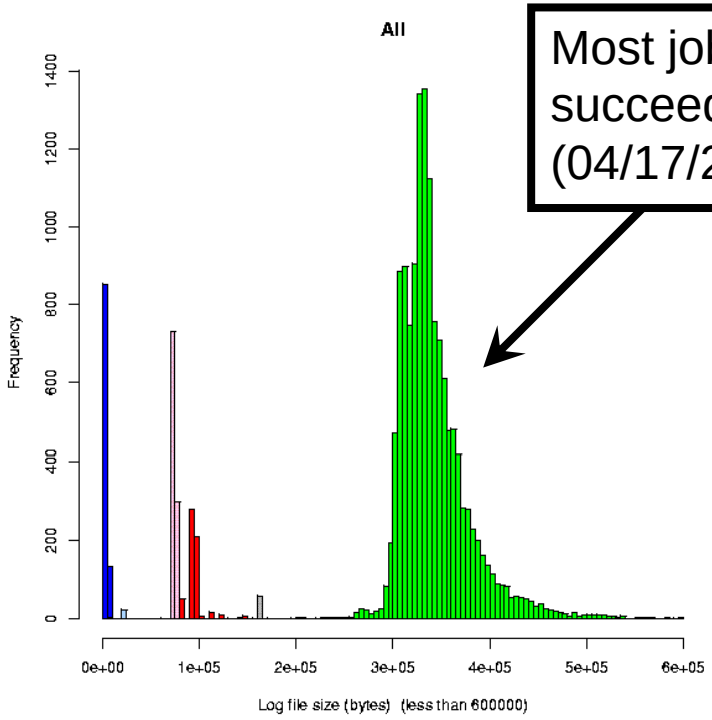
gray 160k-250k Possible D0 runtime crash

green >250k OK

Troubleshooting



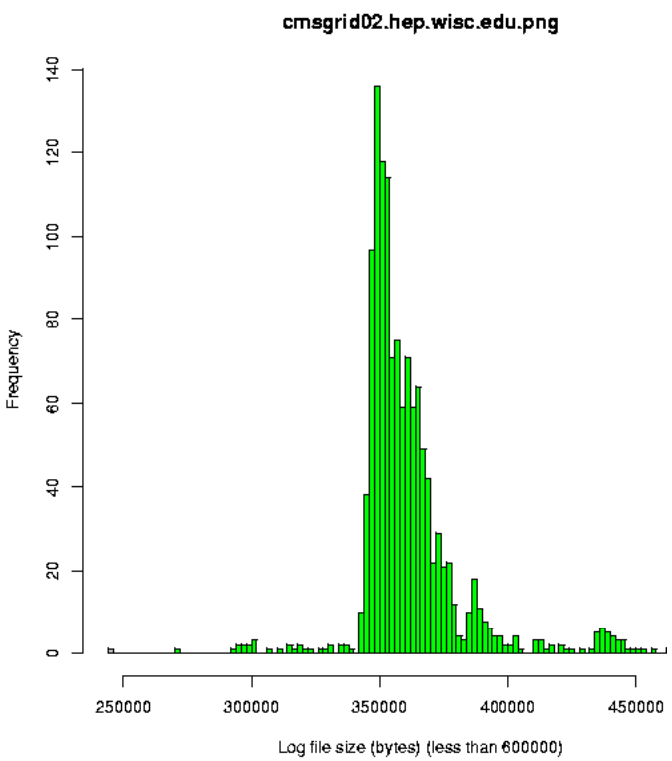
OSG-Related Problems before the intervention of the Troubleshooting Team (03/27/2007)

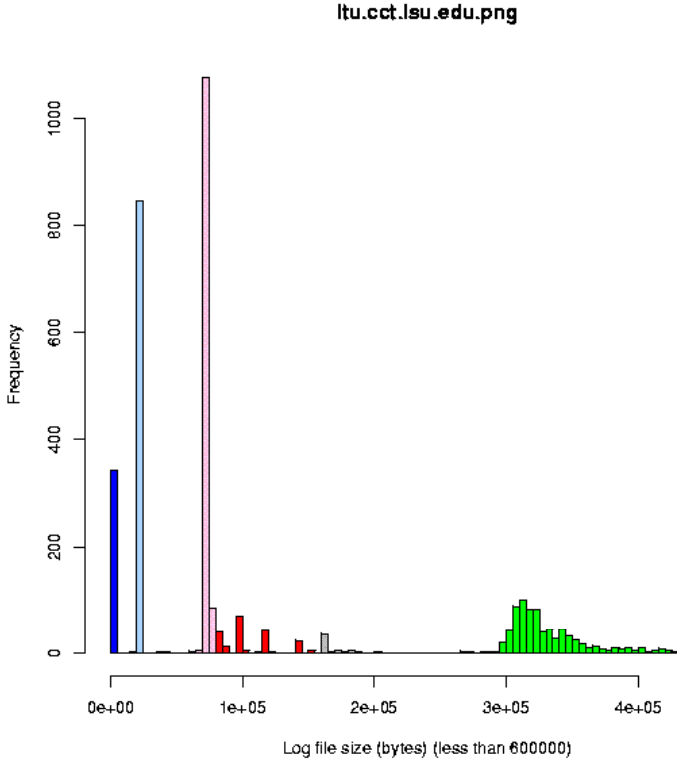


Most jobs succeed (04/17/2007)

The OSG Troubleshooting team was instrumental to the success of the project.

- Success rate at a particular site is not an indicator of the overall system health
  - Failures between sites may not correlate





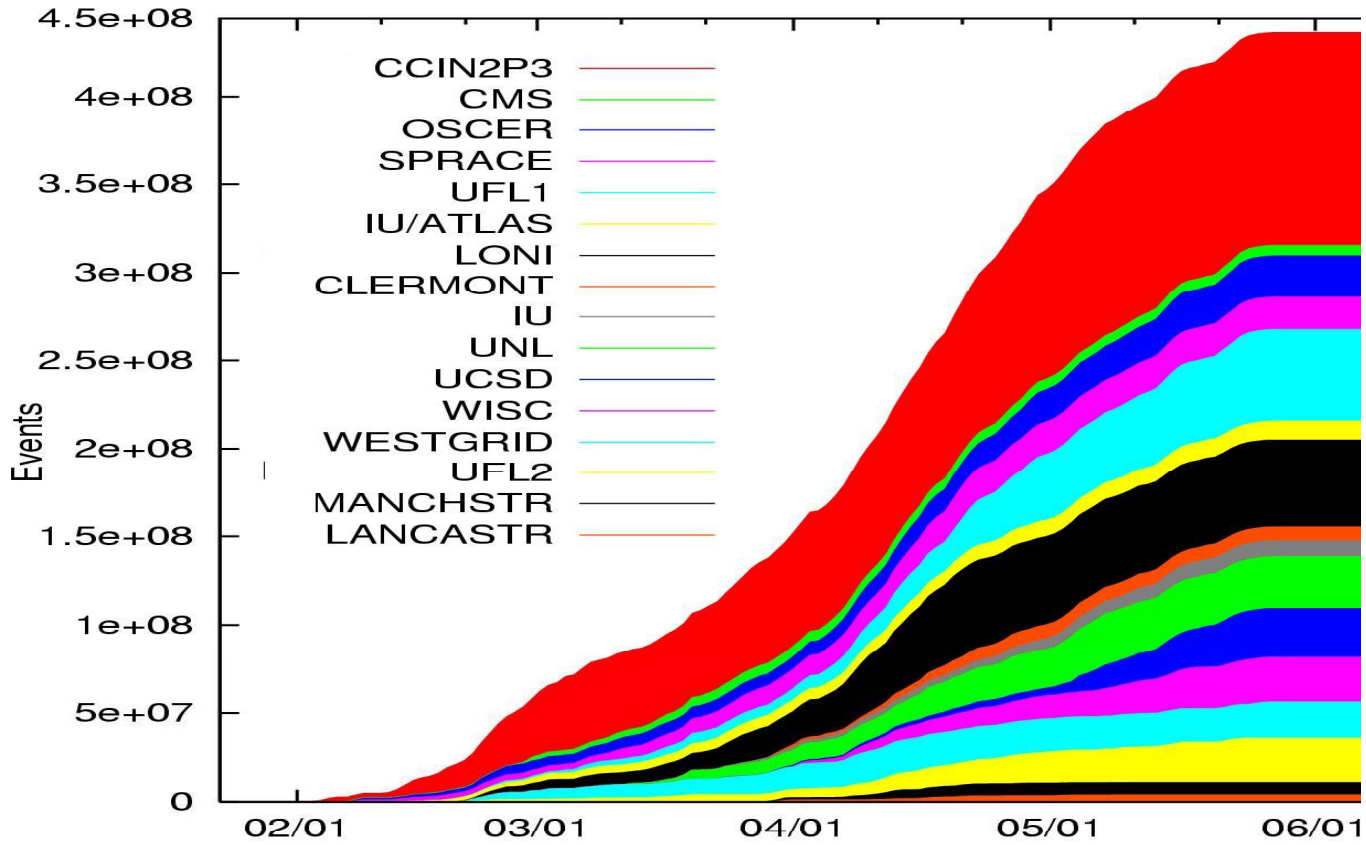
Issue Summary

- Start of the effort
  - 50% of all issues were related to OSG site setups
    - Local scratch and permission problems, Condor/job manager state mismatch, libc incompatibilities ,etc…
      - Direct communication with site admins and help from the OSG Troubleshooting team addressed those
  - 50% of data delivery failures
    - Tuning of data handling infrastructure
    - Maximize use of local storage elements (SRMs)
    - Site connectivity investigation

Issue Summary

- Middle of the effort
  - Coordination of resource usage
    - over-subscription / under-subscription of jobs at sites
      - OSG did not implement automatic resource selection at the time.
      - Job submission operators had to learn how many jobs to submit to each resource manually.
    - “unscheduled” downtimes (both computing and storage)
  - System experts had trouble keeping up with issues by themselves. DZero had to ask help to the OSG Troubleshooting team.
- End of the effort
  - Some slow-down due to insufficient automation in job failure recovery procedures

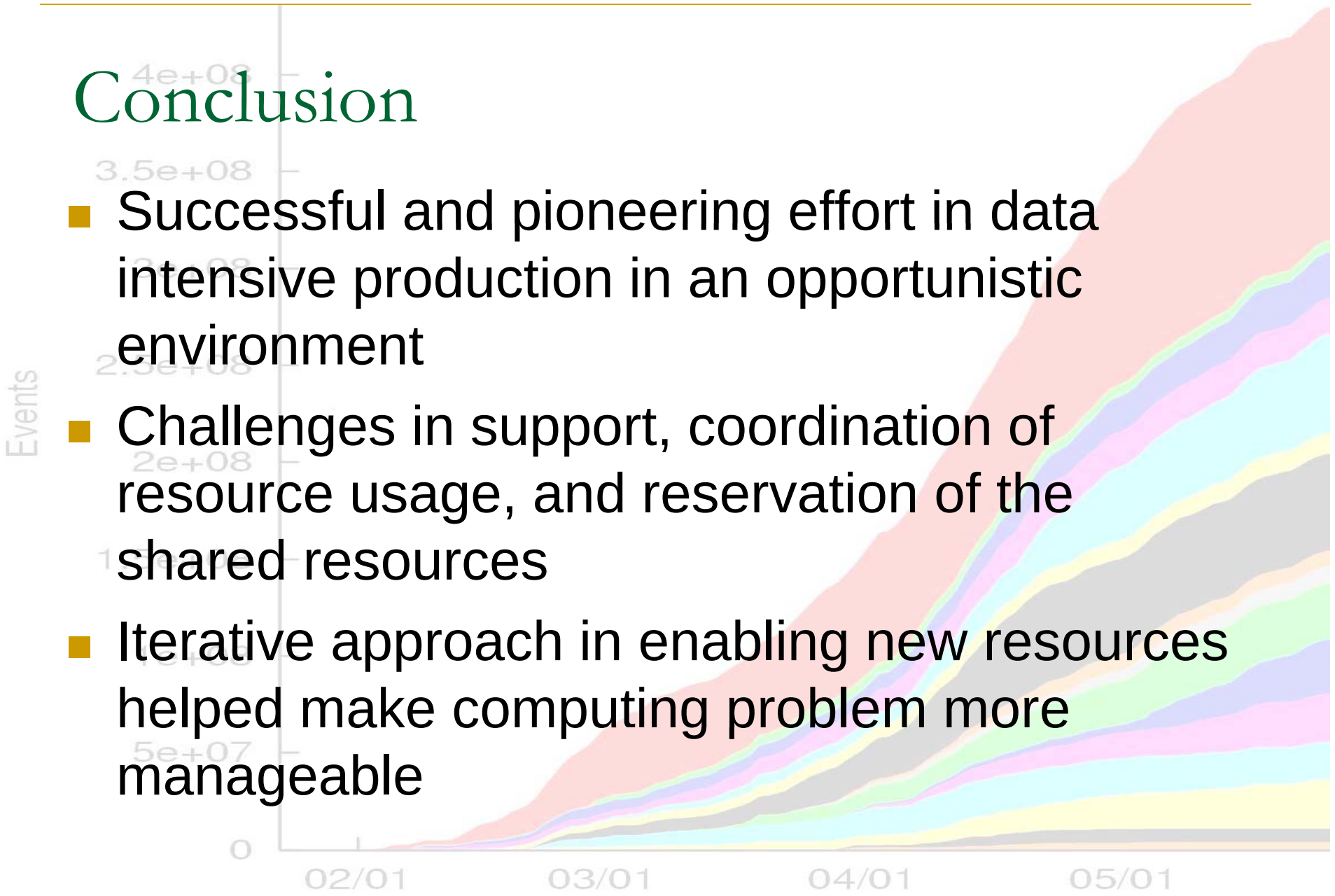
Our result



- 450M collider events delivered to physicists
  - Reconstructed in fully distributed, opportunistic environment

Conclusion

- Successful and pioneering effort in data intensive production in an opportunistic environment
- Challenges in support, coordination of resource usage, and reservation of the shared resources
- Iterative approach in enabling new resources helped make computing problem more manageable



Acknowledgments

- This task required the assistance of many beyond those listed, both at FermiLab and at the remote sites, and we thank them for helping make this project an accomplishment that it was
  - Special thanks to OSG for supporting DZero data reprocessing