# WBS 2.3 Topics

## Distributed Computing Operations: Issues and Challenges

Frederick Luehring
Indiana University

ATLAS Distributed Computing Technical Interchange Meeting

January 21, 2025

# Disclaimer

- I am not trying to embarrass anyone but I find it useful to use real incidents to illustrate the talk.
  - I do not name names or sites but I expect you will figure out who was responsible.
  - Given the title I was assigned, the incidents will be those that affected the US sites.
- In order to do better we have to look carefully at previous incidents and learn.
  - Let’s keep the discussion on doing better and not on deflecting blame or punishing the guilty.
- I do understand that a lot of effort goes into operating the production system and many people are already working hard to make sure we make good use of resources. My goal is for us to do better.
- I have drawn freely from previous presentations and you may recognize some ot the slides because they have been shown before.

# Introduction and Motherhood Statements

- This talk will focus on two topics:
  - How we waste resources because of undetected problems. PROBLEMS
  - How we can improve our problem detection and communication. SOLUTIONS
- Our resources will always be limited funding even in the US! ;-)
  - Resources not being used are wasted.
  - Resources used that do not produce useful results are wasted.
    - If we run jobs with no useful results, then we we have warmed the globe for no benefit.
    - We still seem to be producing a lot of data that is WORN: Write Once - Read Never.
      - WORN wastes energy producing the data and wastes storage storing the data.
      - Storage ain't cheap or unlimited!
- The ratio that matters is total **USEFUL** Walltime over Calendar time.
  - Having servers sitting idle with no work is a loss but still better than running jobs that fail after consuming a lot of time or finish and produce output that is not useful
  - The Physics Community needs to provide enough validated work to keep the grid busy.

# About WBS 2.3

- Work Breakdown Structure 2.3 is the US designation for the activity and funding for offline computing.
  - The area encompasses:
    - The BNL Tier 1
    - The 4 Tier 2 sites (AGLT2, MWT2, NET2, and SWT2)
      - Now includes the Taiwan TW-FTT tier 2
    - The 3 analysis facilities (BNL, UC, and SLAC)
    - The US HPC work (currently at NERSC/Perlmutter and TACC)
    - Operations for the US cloud including R&D
  - WBS 2.3 collaborates with other HEP projects (e.g. IRIS-HEP) and the HEP SW Foundation.
  - WBS 2.3 does include development of code for monitoring / running the production system.
  - WBS 2.3 does not include software development for ATLAS offline code (e.g. Gaudi/athena.) That is the responsibility of WBS 2.2.

# WBS 2.3 Is the Proverbial Canary in the Coal Mine

- WBS 2.3 runs a lot of work, so we see a lot of issues.
  - While we are not over-staffed, we do have more people than other sites and sometimes spot problems that might be missed at smaller sites.
  - Ivan works on US operations and leads a daily meeting.
    - This meeting contributes greatly to spotting issues.
  - And of course I can be a PITA if I think that there is a problem.
- I do have one request: Could y'all please stop referring to problems as a US problem or US site problem until it is conclusively shown to be a local problem.
  - I realize this does not really matter but in some way it is dispiriting to have problems referred to as US problems until it's clear they are.
- The rest of this talk will be more general and less focussed on WBS 2.3.

# Part 1: Problems

# Sources of wasted/unused CPU cycles (not site related)

- Central production system issues (e.g. Harvester/Panda/Authentication/Pilot.)
  - Lots of proxy expirations and firewall issues.
  - Misconfiguration in CRIC and Panda.
  - Issues with external projects like cvmfs or root.
- Job sets that either fail or produce results that are not used.
  - I saw a job the was retried 972 times but after the ~4th attempt the chance of success is low.
  - I saw a user running jobs for 6 weeks with a 100% failure rate. This is not sensible,
- Network issues preventing the download/upload of files used/produced.
  - Stage out failures are bad: nearly the full CPU/wall time has been used and it's all wasted.
- User jobs: Running huge productions with lightly or untested code, users not monitoring the results of their jobs, etc.

# Problems with Production Jobs (Easy)

- Last week Judith messaged me that MWT2 servers were going OOM.
  - I used a brute force Panda Monitor query see what production jobs had failed in the last 12 hours anywhere in the world: https://bigpanda.cern.ch/jobs/?hours=12&jobtype=prod&jobstatus=failed&mode=nodrop&display_limit=100&limit=100000
  - I saw that there were 47k failed jobs and 46k of the failures were for excessive memory.
  - After a quick discussion with Ivan and Ofer, I notified the production manager and they realized that somehow the transform had changed the submission from being to the VHIMEM queue to a regular queue.
- Finding this problem would be possible for even a beginning shifter.

# Problems With User Jobs (Easy)

- Two weeks ago Judith and Aidan message me that some of the MWT2 storage servers were overloaded.
  - Judith and Aidan quickly traced the issue to a single user with thousands of I/O requests.
  - Looking at that user, I could see that a large fraction of their jobs were failing and had been for a day or two. I reported this to the DPA who passed the issue onto Miguel and Mayuko.
  - The user was asked to stop submitting broken jobs that had not been tested sufficiently.
  - The user did not respond.
  - A day or two later Ivan killed the user's jobs and temporarily blocked the user.
  - Eventually the user responded but seemed confused,
- Again a beginning shifter could have spotted this issue.
- We need to educate the users that resources are not infinite/free, that they need to run small test job sets initially, and that they need to look at the failure rates of their tasks and not keep submitting blindly.

# Issues with Production System Packages (Very Hard)

- There have been a series of issues that required changing the pilot/wrapper.
  - Pilots not recognizing that the payload failed and continuing to hold the allocated job slots until the token expired after 96 hours while waiting for the now defunct jobs to end.
  - Pilots not recognizing that the payload was still running and starting a second payload.
  - Pilots returning CPU efficiency data based on crazy data.
  - Pilots setting up cgroups memory limits in a way that the pilot itself was SIGKILL-ed. Since the pilot killed itself, it was possible to return info about what happened to Harvester.
    - It took a meeting with an OSG/Condor developer to understand what was going on.
- Most of these problems have been mitigated/fixed but it is not clear to me that the underlying cause has been fully understood in all cases.
  - The pilot seems to accrete complexity organically. Does Paul ever sleep?

# Problems With an External Project: cvmfs (Hard)

- One site in particular and all US sites at some level were affected by bugs in the initial releases of the cvmfs 2.11.x series that hung servers.
  - It took months to identify that the issue was on the the cvmfs side and for the cvmfs team to fix the underlying problems.
    - It’s not clear to me that even today all of the underlying issues are identified and fixed.
- One site was setting an environmental variable to increase the file cache size.
  - The documentation said that setting values this way was possible for all cvmfs settings.
  - However an oversight by a cvmfs developer caused cvmfs to read only a few variables.
- Responsibility for cvmfs problems is unclear: ADC does not support cvmfs, OSG distributes cvmfs, and the cvmfs team does the actual support.
  - It took some time to sort out who had to do what.

# Problems (site related)

- Local batch queue problems
- Gatekeeper problems (stuck gatekeepers, not enough gatekeepers)
- Storage system problems
  - dCache and xRootD spread files from a dataset over many pools and if a pool goes down many jobs fail because one or more of their input files is no longer accessible.
  - File corruptions and losses will cause jobs to fail and recovery can be really painful when a large number of files need to be recreated.
- Black hole nodes (one node can cause the whole site to be put offline)
- Firmware and software version problems.
- Local networking (switch issues, dirty fibers, DNS problems, IPV6 problems)

# Part 2: Solutions

# The Existential Issue

- Our operations tend to be very siloed.
  - I call this the upstream/downstream problem.
    - Someone finds a problem and they decide it is a problem at a site (downstream problem) and they don't worry further about it.
    - Someone finds a problem and they decide it is a problem with central services (upstream problem) and they don't worry further about it.
  - IMHO we are all the ATLAS production team and any problem is our problem.
    - Problems should always be reported - they often are not.
    - People are heavily loaded and quick to say that something is some else's problem to deal with.
- We need to work together and not say "Oh it's just those guys again…'

# Change Management

- If I were the king of production, I would insist that all changes were:
  - Proposed well in advance of deployment.
  - Discussed with the clients.
  - Scheduled and done at an announced time.
  - That the clients be informed both immediately before and after the change.
  - Tracked in some sort ticketing type system.
- All too often changes are put in with no visibility outside of central team (think CRIC parameter changes.)
- Similarly sites change things without informing anyone.

# Communication

- We need clearly defined reporting and notification channels.
  - There should be a single, push mechanism for notifying everyone of outages.
    - The notifications should be in real time to prevent people from looking for problems at their site that are known external issues.
  - There should be single system that everyone uses to report problems.
    - This will encourage reporting of issues and make it easy to do so.
    - It will cut down on the amount of information spread around all of the ways we use now (email, ggus, Jira, N different chat systems, Discourse, smoke signals, etc.)
      - We need to avoid information about problems being hidden and/or lost.
- We also need good communication channels with external projects: e.g. OSG, IAM, and cvmfs.

# Monitoring

- There are (too) many monitoring tools available to non-experts.
  - I constantly hear for the site administrators that they want a single, clearly defined set of monitoring tools.
  - Busy people are unwilling to sort through all the available tools to figure out what works to provide them with the information they need.
  - My suggestion is that there be some clear, simple starting place with links a small set of monitoring tools that work and are useful.
  - I could actually see see several starting pages each with clearly defined audiences:
    - One for the users running jobs on the grid
    - One for site administrators running sites.
    - One for people working on operations.
    - One like the current ADC Livepage for the experts.

# Conclusion / Worries

- I wonder whether we buy too much hardware and do not pay for enough skilled people to run & monitor production system.
  - There is a very small cadre of long term experts in the center of operations.
  - A key link is our heavily loaded system administrators who are focused their own sites.
    - We need to make it easy for the site administrators to report things they find.
- I find problems that it takes someone experienced to spot.
  - There are a small group of US people looking for and analyzing problems
- The whole approach is a rather unstructured way to operate a project with 600k job slots, an exabyte of data, its own private high speed network, and sites distributed around the world.

# Bonus

- I do understand from Jammel Brooks (IU Grad Student) that she is looking into creating tools to tell site problems from central problems.
  - I know Jammel well and have agreed to meet with her every two weeks especially since she seems to be addressing issues that I have asking about.