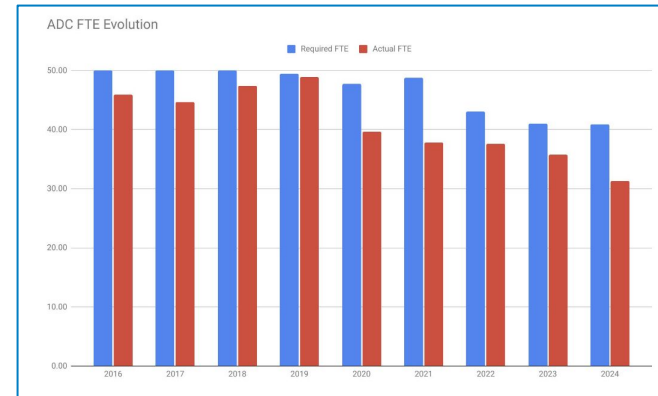


Automation: ADC

Ivan Glushkov (BNL)
on behalf of ADC

- In an ideal world we would need shifters.. or Ops team
- **Strongly dependent on the person power available**
- Should be a priority within every team. Is it?
- Depends on teams loosely connected to ADC - FTS, CRIC, dCache, HTCondor, xrootd, HammerCloud, Site admins, HPC experts..
- Requires an inter-team collaboration within different teams - WFMS, Rucio, FTS, CRIC, Central Services..
- Once in place, there should be automation of the recovery of the automation recovery (why the storage dumps are not there for the last three days?)

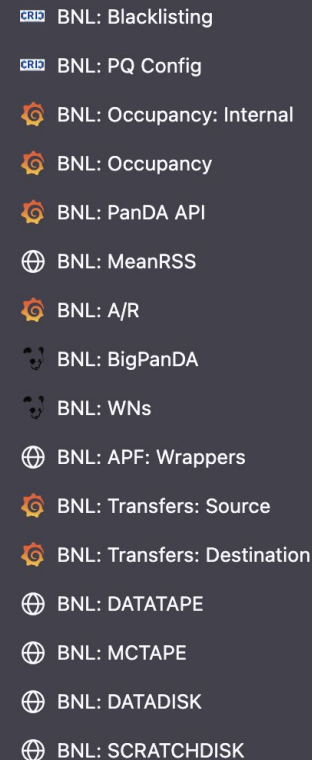


















- **Communication automation:**
 - Central infrastructure problems: WFMS, DDM, FTS, Hammercloud, Network..
 - Expected workflows and their implications on the sites: Tape loads, Network loads, massive VHIMEM, SCORE campaigns. Workloads that would require high-volume data movement
- **Failure rate:**
 - Automation of task/request management based on previous failure rate.
 - Pause task
 - Forbid further retries
 - Finer automatic retry actions
 - Automation of reverting software based on failure rate? (Ex: pilot and Rucio CLI)
- **Sites:**
 - Commissioning and decommissioning of sites
 - A/R tests. But better. Is this acceptable for WLCG@HL-LHC?
 - HammerCloud tests. But better. Take into account production jobs?
 - Dynamic update of CRIC parameters based on the load / failure rate / utilization of the system (do you want to keep this limit of number of VHIMEM slots and be empty, or shall I ignore it for the moment?)
 - Average IO per PQ, limit on transferring jobs, any associated parameter in CRIC
 - I have dark data - how can I be sure that this is expected? A ticket to DDM Ops...

- DAST & ADCoS
 - LLM?
- VO problems
 - It should be trivial to link ATLAS secretariat, CERN HR DB, and IAM, shouldn't it?
- Bad users
 - Limit retries, pause tasks, kill running
 - Ban users automatically based on critical metrics - overloading a site storage, failing rate, etc
 - We need finer grained authorization im IAM (see user status, ban user, etc)

- Person power
 - We need dedicated ATLAS Rucio person power.
 - We need automation (not to be stolen for operations!) person power.
- Potential tasks
 - Rucio/DDM self consistency checks and recovery automation for situations which currently are noticed only via external factors - Panda, DPA, users
 - Automation of commissioning/decommissioning of a storage
 - Processing traces to handle bad files
 - current half-implementation is not flexible enough
 - Automate any additional diagnostics and choose remedy
 - Transfer efficiencies and single file rates, e.g. 10GB files always timeout on some links
 - Reassess sources or add hop to speed up transfer
- FTS
 - Follow and steer FTS4 development
 - A [list of ATLAS requirements to FTS](#) was collected after DC24
 - Best network path
 - Spot misconfigurations: Not using LHCONE / LHCOPN, not using IPV6 when possible, etc.

- Monitoring: the service
 - We do not need more new monitoring! (in general)
 - We need to:
 - Ensure service reliability
 - Ensure data consistency and reliability
 - Employ what we have in automation
- User experience
 - New monitoring, even with more information does not mean better user experience (people still prefer apfmon over os-atlas for some functionalities)
- Consolidation
 - All site monitoring in one place needed. Currently one needs 15 (ATLAS) web pages for full site overview.
 - Internal site monitoring. Do we need a standardized interface between the two?
- Automation on actions based on monitoring
 - Panda / gdpconfig changes (the system is not filled since we limit the evgen in gdpconfig?)
 - CRIC changes: Dynamic corepower based on HS24 tests
 - Site specific changes - blacklisting CEs/gatekeepers, rebooting black hole nodes, declaring lost files
 - Opening tickets. Automatic pinging of tickets.



-  BNL: Blacklisting
-  BNL: PQ Config
-  BNL: Occupancy: Internal
-  BNL: Occupancy
-  BNL: PanDA API
-  BNL: MeanRSS
-  BNL: A/R
-  BNL: BigPanDA
-  BNL: WNs
-  BNL: APF: Wrappers
-  BNL: Transfers: Source
-  BNL: Transfers: Destination
-  BNL: DATATAPE
-  BNL: MCTAPE
-  BNL: DATADISK
-  BNL: SCRATCHDISK

- Static documentation (Current status)
 - Pros:
 - It is actually written and verified by an expert
 - Cons:
 - It is true at the moment of writing. It requires constant reviewing (Documentation week, etc.)
 - No way to enforce updates in the system to be reflected in updates of the documentation
 - Orphaned/obsolete documentation
- Can AI do that?
 - Generate (user) documentation updates based on:
 - Code changes (functionality changed)
 - System performance (nucleus requirements should be changed based on current nuclei performance and on the needs of the system)
 - Obsolete documentation (the steps described here do not work)

- Data analytics
 - Follow up on errors (8% waste of pledge)
 - Underperforming infrastructure - CEs, SEs, network links,..
- LLM. For user support and documentation. Definitely.
 - There are many questions from users that we do not even see. There is no stupid questions to ask a chat
 - Collect and clean training data:
 - E-mails / e-group archives
 - Tickets: GGUS, JIRA
 - Twikis, codiMD
 - Documentation - Panda, Rucio, HTCondor, Harvester, CRIC, etc..
 - Indico - presentations, discussions
 - Codebase: gitlab, github
- People to automate daily ops
 - Ops side: “Tell me not how it was solved. Tell me why it won’t happen again”
 - Mgmt side: “Automation will not solve everything”
 - It would improve the utilization of resources
 - It would decrease operational team load