ATLAS Reprocessing

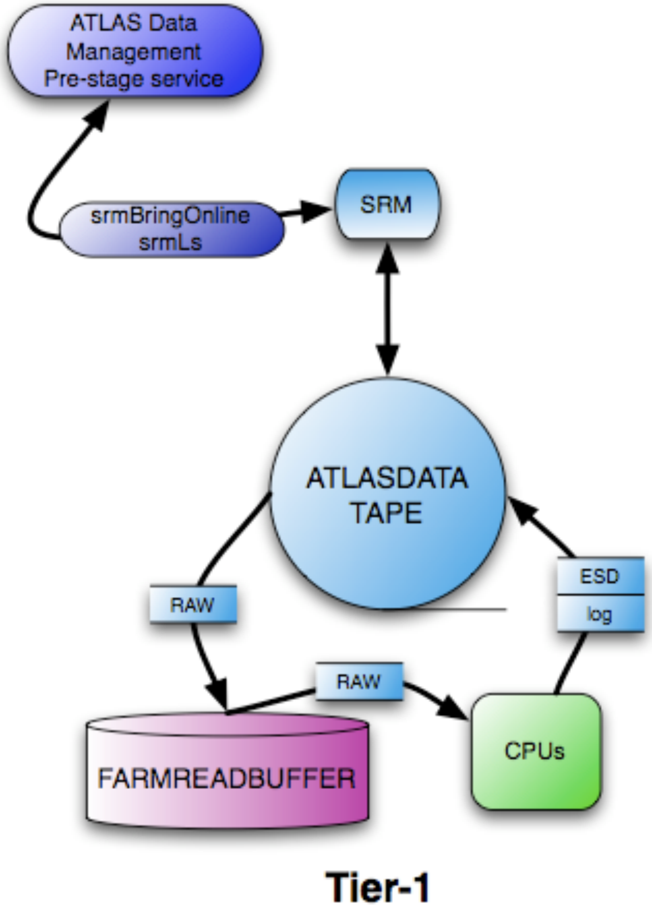
Xavier Espinal (PIC/IFAE)

Outline

- Reprocessing targets
- What robots want and what PanDA/DDM does
- Reprocessing jobs and data workflow
- Nprestage
- ATLAS reprocessing metrics
- Conclusions

Targets

- Considered running the whole spring reprocessing campaign
  - Too complex for operational issues (involving about 10k tasks)
  - Decided to do a single task per cloud (pseudo-reprocessing)
    - Run RAW to ESD jobs on special jumbo datasets
      - Using cosmics 2008 data
- Read RAW from tape (cache cleanup in advance). ESD_and_logs written to tape
  - Pseudo-repro: smaller output files but same amount
    - Metrics based on files/hour not throughput
  - No AOD/DPD production
- Exercise full tape recall machinery at the Tier-1s
- Exercise ATLAS reprocessing mechanisms, fully automated and based on:
  - PanDA: job workflow (defined-assigned-activated-running)
  - DDM: data workflow (Process assigned -> activated step using dataset pre-stage service)



What robots want ?

- Good robot usage is mandatory for efficient reprocessing
- Robots like to:
  - Receive bulk petitions for recall
    - Internal MSS reordering capability: minimize tape mounts and seeks
  - Ordered jobs:
    - Data is stored on tapes, using file families. Bulk of consecutive jobs asking for consecutive data is optimal.
  - Data pre-placement mechanism
    - Recalls can be slow. Data should be on disk before the job starts
      - ➡ Prevent jobs to wait during staging
        - Avoid potential problems: low efficiencies, walltime/cputime failures, etc.

What PanDA/DDM does ?

- Good robot usage is mandatory for efficient reprocessing
- Robots like to:
  - Receive bulk petitions for recall

    Job sent in chunks. Constant queue of recalls O(2k)
  - Ordered jobs:

    Input data blocks of 20 files (20 jobs)
  - Data pre-placement mechanism

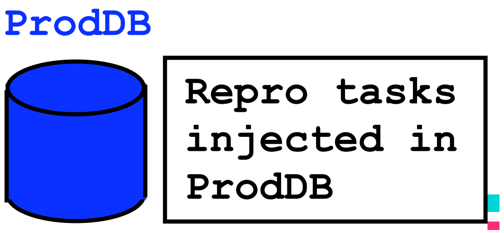
    Special subscription (DDM) issued for recall.
    Notice once file is on disk: DDM polling SE (srmLs)
    Callback to change job state: job activated once file is on disk

Workflow

- ATLAS reprocessing workflow is fully embedded in PanDA/DDM

Workflow

- ATLAS reprocessing workflow is fully embedded in PanDA/DDM
- Reprocessing coordinators insert the tasks (jobs collection) in the production DB

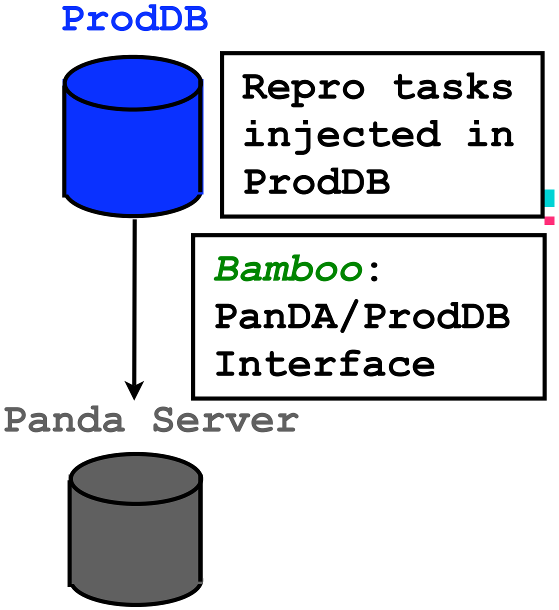


ProdDB

Repro tasks injected in ProdDB

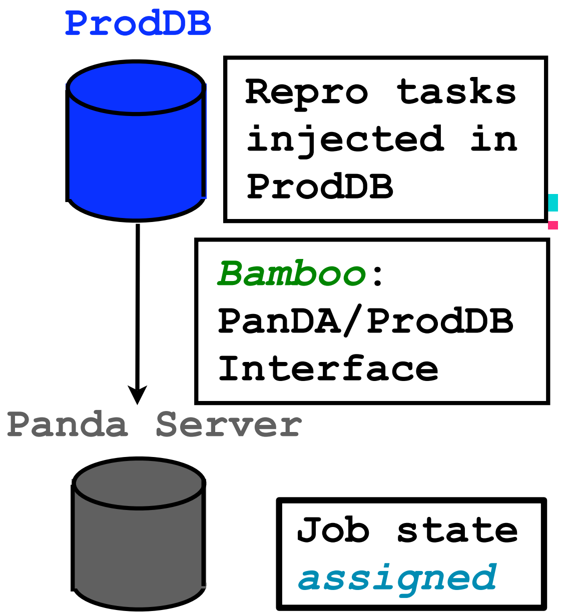
Workflow

- ATLAS reprocessing workflow is fully embedded in PanDA/DDM
- Reprocessing coordinators insert the tasks (jobs collection) in the production DB
- The supervisor (Bamboo) pick up jobs from ProdDB and feed PanDA server:
  - Ensure a good job keep-up
  - Pick-up jobs if :
    - queued/running <2 or
    - queued<Nprestage (minimum number of recalls per site)
    - This increase number of files to be requested at the sites and maintain it
      - Optimizing MSS efficiency (local reshuffling)


ProdDB
Repro tasks injected in ProdDB
Bamboo: PanDA/ProdDB Interface
Panda Server

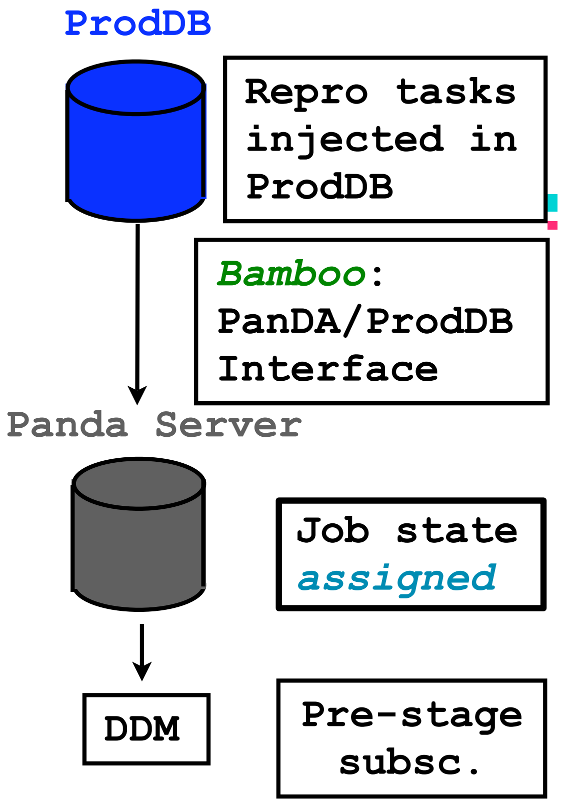
Workflow

- ATLAS reprocessing workflow is fully embedded in PanDA/DDM
- Reprocessing coordinators insert the tasks (jobs collection) in the production DB
- The supervisor (Bamboo) pick up jobs from ProdDB and feed PanDA server
- Pre-stage: jobs tagged in "assigned" state in PanDA


ProdDB
Repro tasks injected in ProdDB
Bamboo: PanDA/ProdDB Interface
Panda Server
Job state *assigned*

Workflow

- ATLAS reprocessing workflow is fully embedded in PanDA/DDM
- Reprocessing coordinators insert the tasks (jobs collection) in the production DB
- The supervisor (Bamboo) pick up jobs from ProdDB and feed PanDA server
- Pre-stage: jobs tagged in "assigned" state in PanDA
- Trigger special DDM subscription from TAPE ST to same TAPE ST
  - Pre -staging mechanism is DDM (used for all sites except US -PanDA Mover-)



ProdDB

Repro tasks injected in ProdDB

Bamboo: PanDA/ProdDB Interface

Panda Server

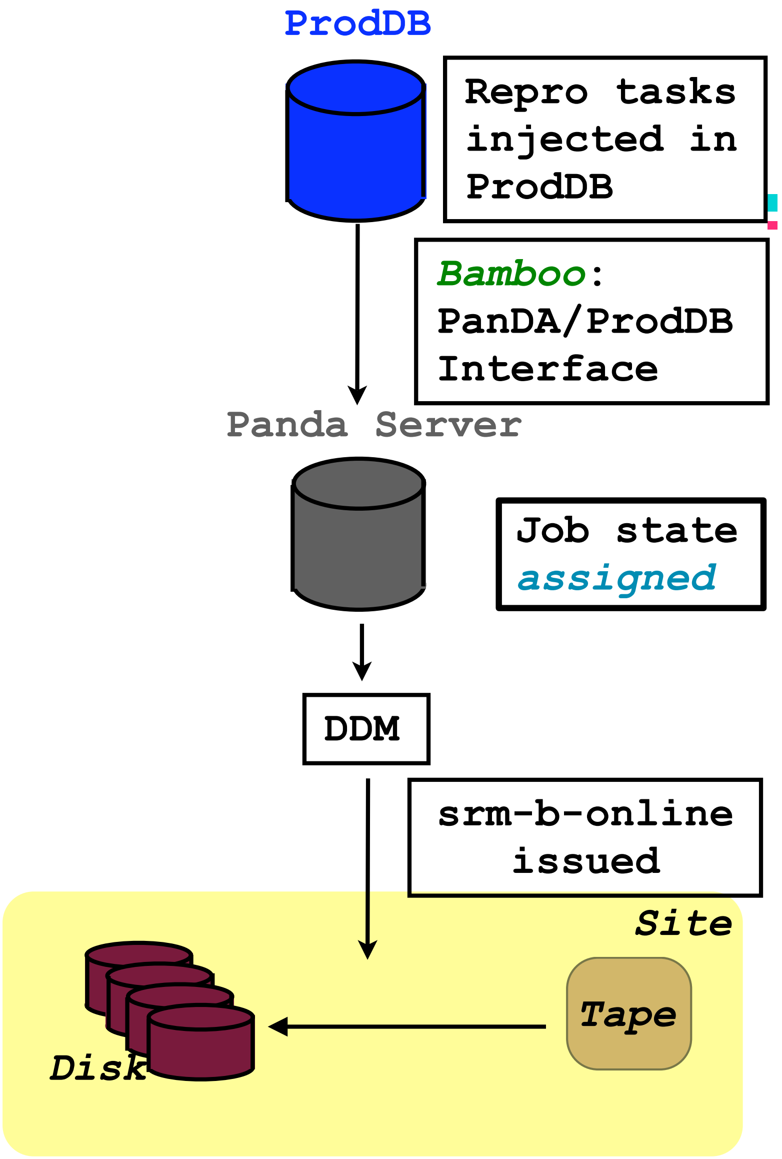
Job state *assigned*

DDM

Pre-stage subsc.

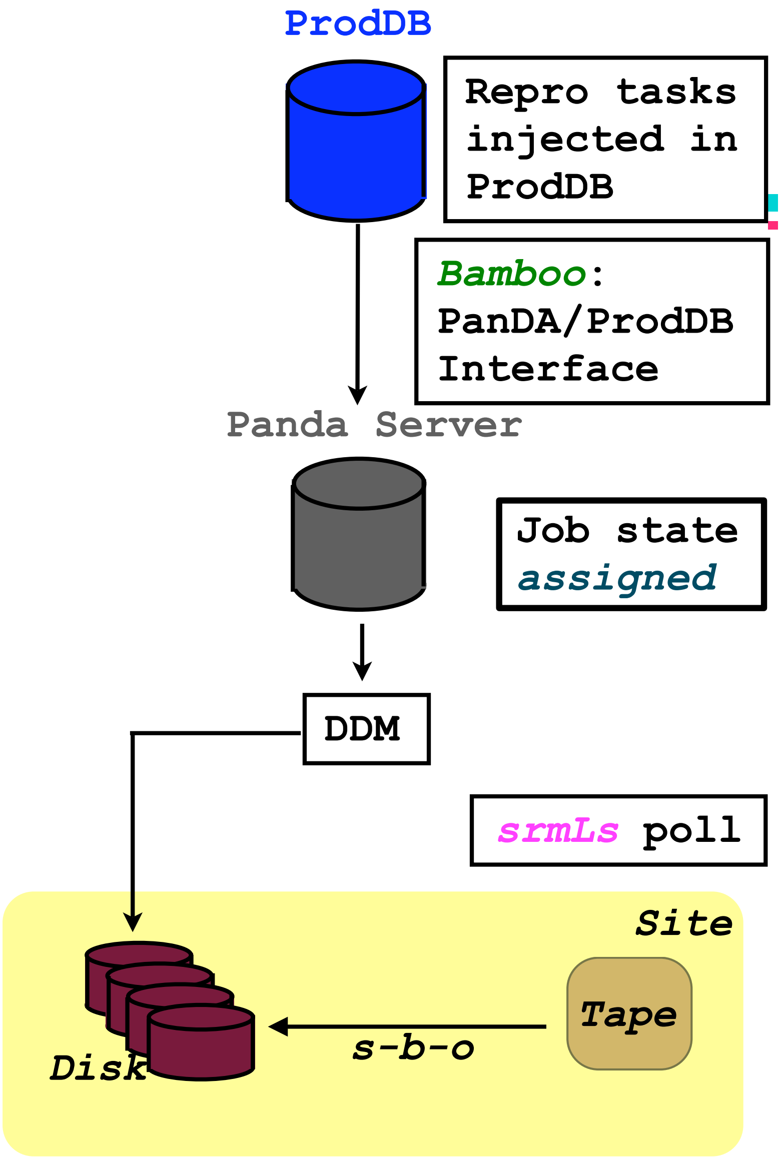
Workflow

- ATLAS reprocessing workflow is fully embedded in PanDA/DDM
- Reprocessing coordinators insert the tasks (jobs collection) in the production DB
- The supervisor (Bamboo) pick up jobs from ProdDB and feed PanDA server
- Pre-stage: jobs tagged in "assigned" state in PanDA
- Trigger special DDM subscription from TAPE ST to same TAPE ST
- srm-bring-online issued (in bulks)



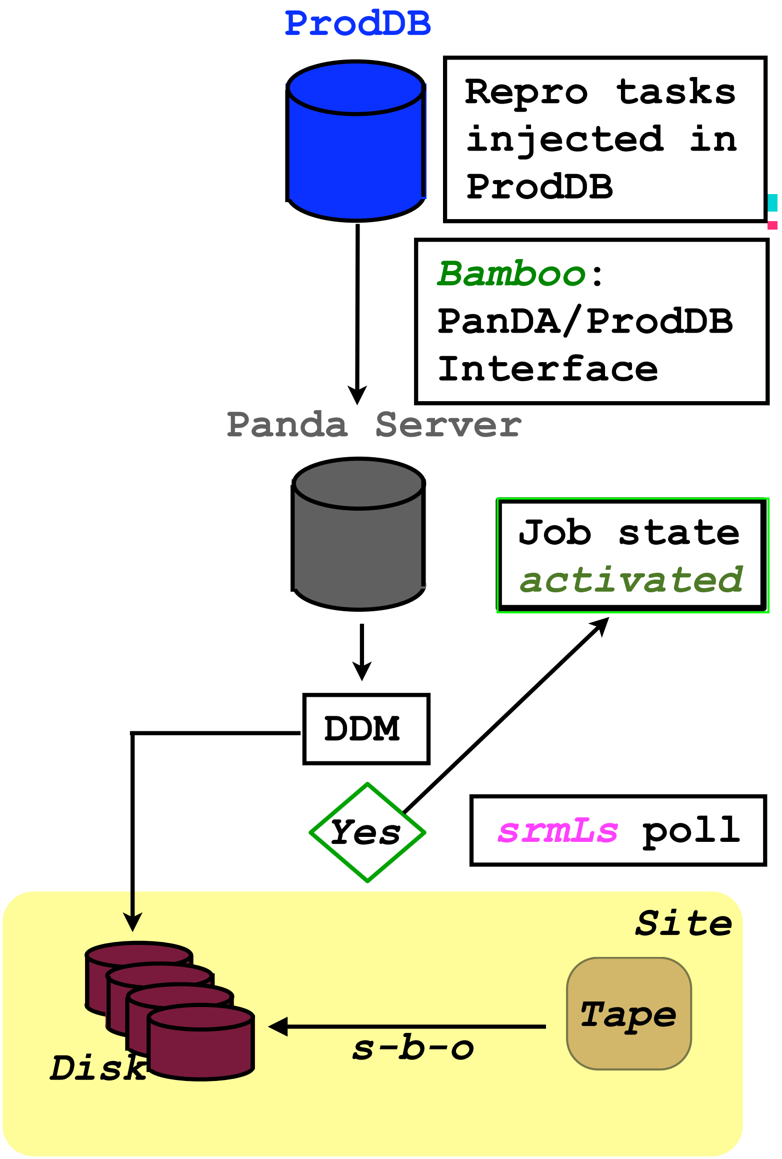
Workflow

- ATLAS reprocessing workflow is fully embedded in PanDA/DDM
- Reprocessing coordinators insert the tasks (jobs collection) in the production DB
- The supervisor (Bamboo) pick up jobs from ProdDB and feed PanDA server
- Pre-stage: jobs tagged in "assigned" state in PanDA
- Trigger special DDM subscription from TAPE ST to same TAPE ST
- srm-bring-online issued (in bulks)
- Check when file is ONLINE (disk):
  - polling with bulk(50) srmLS



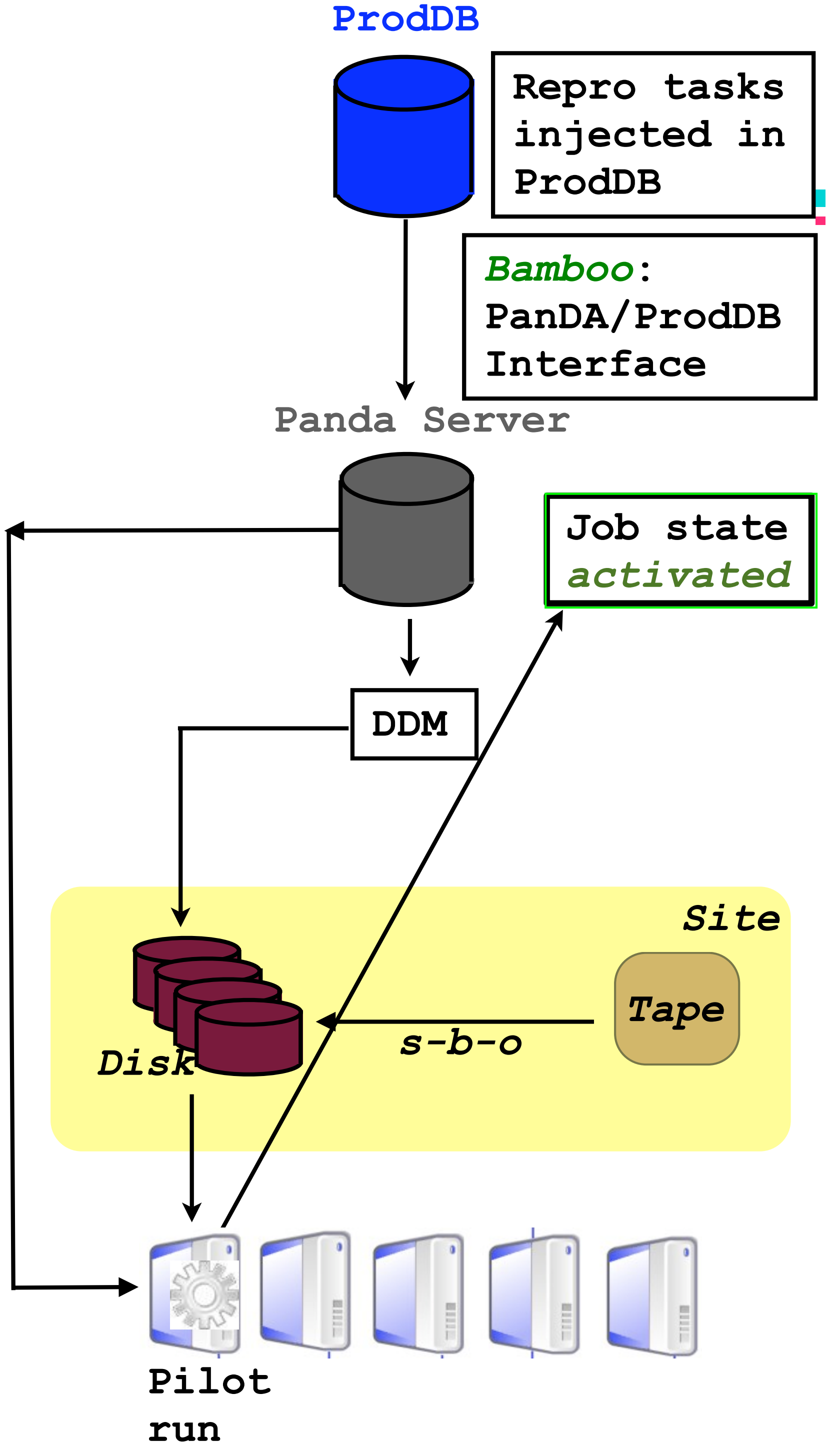
Workflow

- ATLAS reprocessing workflow is fully embedded in PanDA/DDM
- Reprocessing coordinators insert the tasks (jobs collection) in the production DB
- The supervisor (Bamboo) pick up jobs from ProdDB and feed PanDA server
- Pre-stage: jobs tagged in "assigned" state in PanDA
- Trigger special DDM subscription from TAPE ST to same TAPE ST
- srm-bring-online issued (in bulks)
- Check when file is ONLINE (disk)
- Once files are on disk: change of job state in PanDA: from "assigned" to "activated"
  - "activated" means that jobs can be pulled by pilots

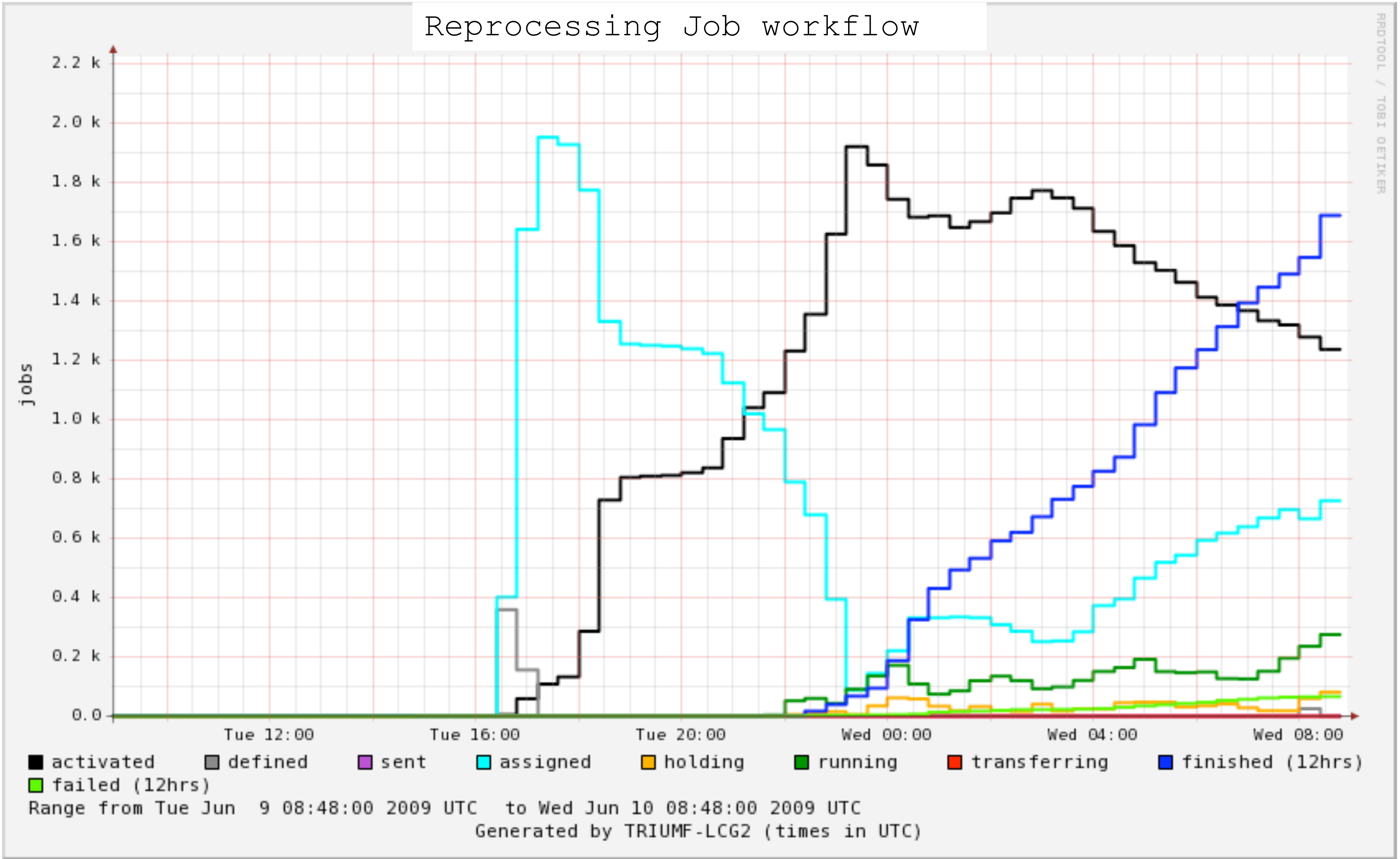


Workflow

- ATLAS reprocessing workflow is fully embedded in PanDA/DDM
- Reprocessing coordinators insert the tasks (jobs collection) in the production DB
- The supervisor (Bamboo) pick up jobs from ProdDB and feed PanDA server
- Pre-stage: jobs tagged in "assigned" state in PanDA
- Trigger special DDM subscription from TAPE ST to same TAPE ST
- srm-bring-online issued (in bulks)
- Check when file is ONLINE (disk)
- Once files are on disk: change of job state in PanDA: from "assigned" to "activated"
- Wait for pilots to pull payload and job run

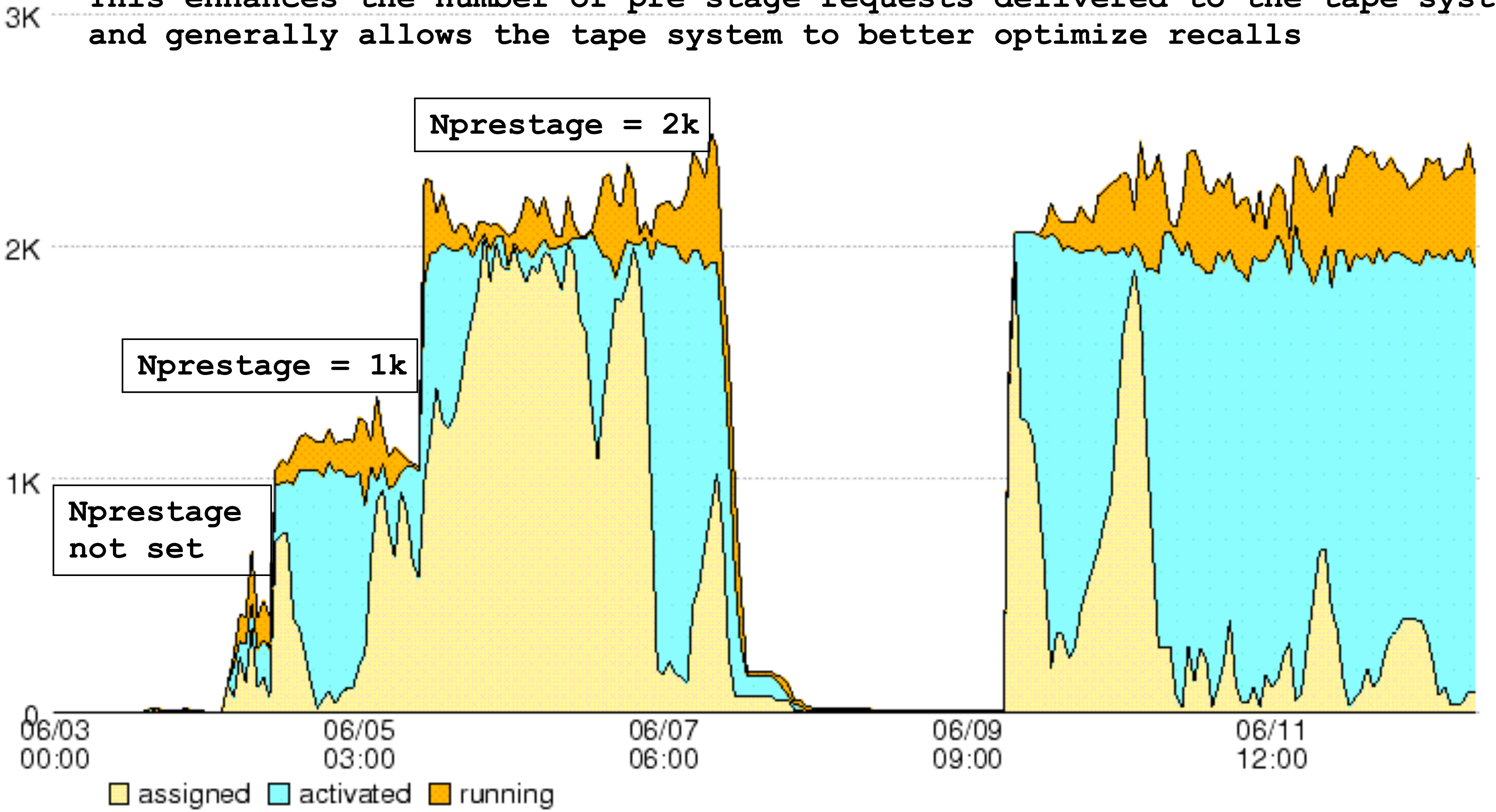


Workflow


Reprocessing Job workflow
jobs
2.2 k
2.0 k
1.8 k
1.6 k
1.4 k
1.2 k
1.0 k
0.8 k
0.6 k
0.4 k
0.2 k
0.0
Tue 12:00
Tue 16:00
Tue 20:00
Wed 00:00
Wed 04:00
Wed 08:00
■ activated
■ defined
■ sent
■ assigned
■ holding
■ running
■ transferring
■ finished (12hrs)
■ failed (12hrs)
Range from Tue Jun 9 08:48:00 2009 UTC to Wed Jun 10 08:48:00 2009 UTC
Generated by TRIUMF-LCG2 (times in UTC)
RRDTOOL / TOBI OETIKER

Nprestage

- Nprestage parameter helped a lot for optimization:
  - Keep constant number of assigned+activated jobs at each T1
    - Can be tuned per site
  - This enhances the number of pre-stage requests delivered to the tape system and generally allows the tape system to better optimize recalls

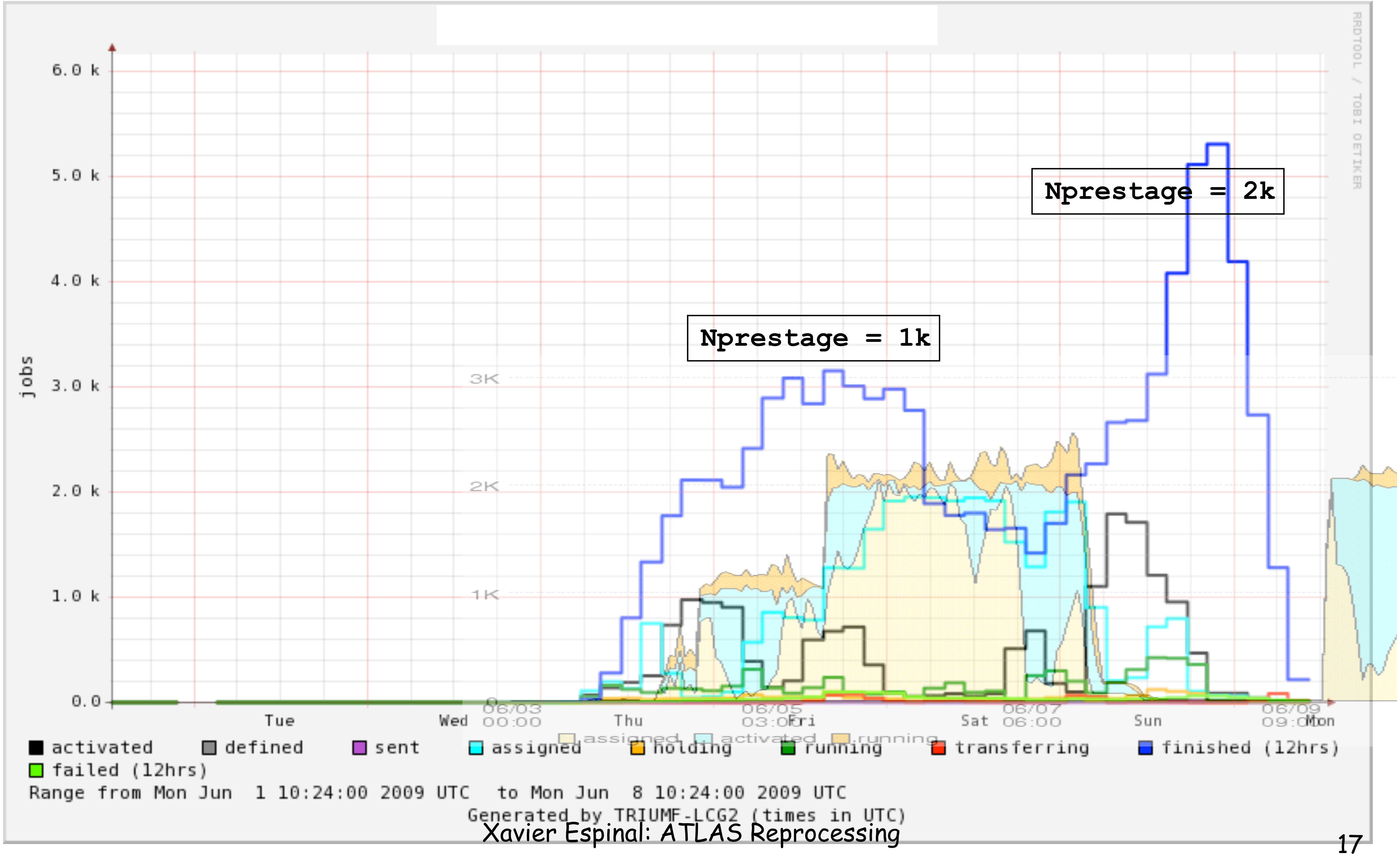


Nprestage = 2k

Nprestage = 1k

Nprestage not set

Nprestage



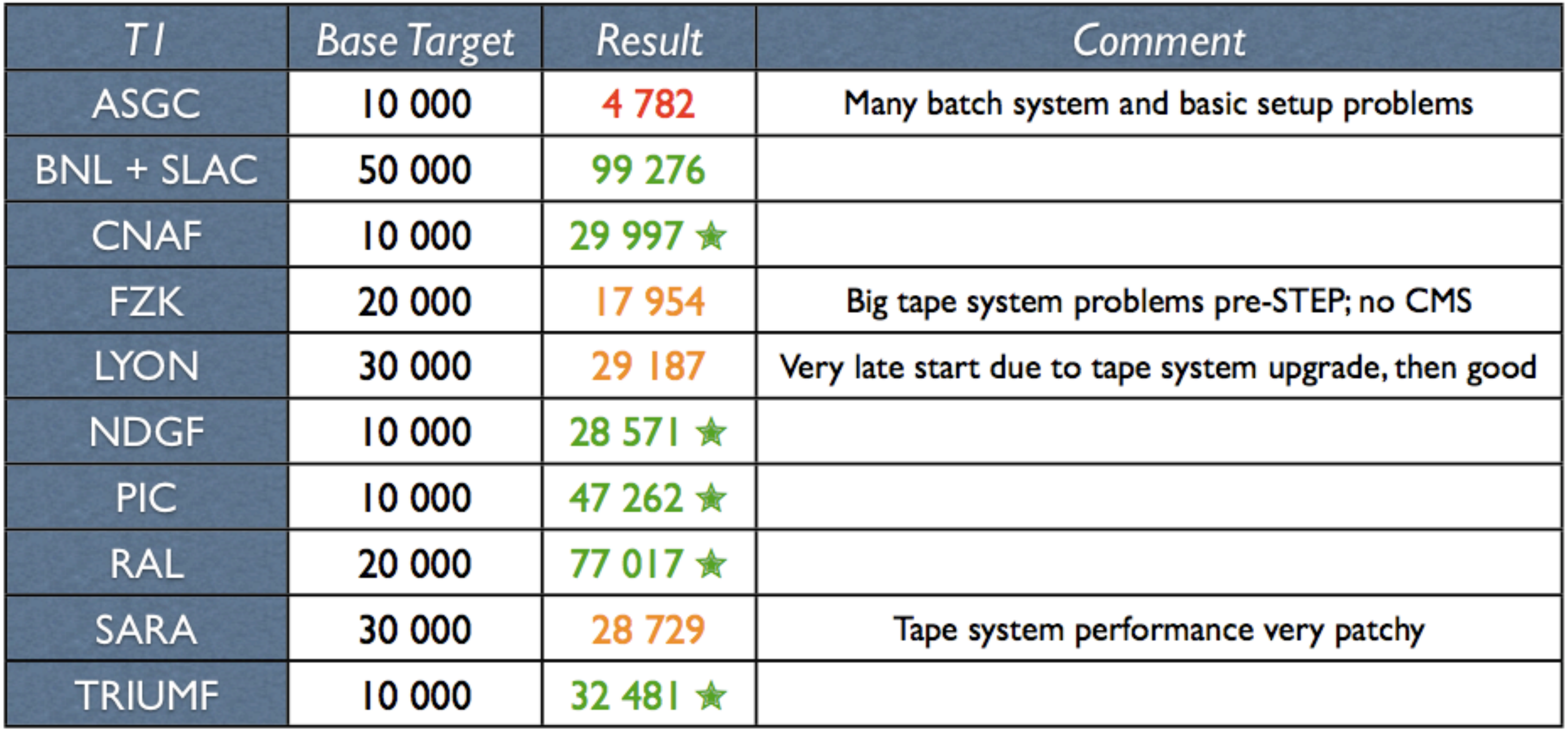
Nprestage = 1k

Nprestage = 2k

STEP09 reprocessing metrics

- ATLAS metrics for reprocessing based on files/day not throughput
  - Pseduo-repro outputs slightly lower in size
- Baseline and enhanced metrics (for the 10 repro-STEP days):
  - Nominal rate: 200Hz (1.6MB/event: 320MB/s) and 50ks/day gives:
    - 16TB/day of RAW data
      - `STEP09: baseline metric:` 400Hz (x2 nominal) :
        - 10% T1: 40Hz (1.6MB/event) => process 3.2TB/day => 2000 files/day
        - ~40 MB/s net rate (20k files over STEP09)
      - `STEP09: enhanced metric:` 1000Hz (x5 nominal):
        - 10% T1: 100Hz (1.6MB/event) => process 8/B/day => 5000 files/day
        - ~100 MB/s net rate (50k file over STEP09)
  - Above numbers account for the needed net rate between WNs and recall pools

STEP09 Results



T1	Base Target	Result	Comment
ASGC	10 000	4 782	Many batch system and basic setup problems
BNL + SLAC	50 000	99 276	
CNAF	10 000	29 997 ★	
FZK	20 000	17 954	Big tape system problems pre-STEP; no CMS
LYON	30 000	29 187	Very late start due to tape system upgrade, then good
NDGF	10 000	28 571 ★	
PIC	10 000	47 262 ★	
RAL	20 000	77 017 ★	
SARA	30 000	28 729	Tape system performance very patchy
TRIUMF	10 000	32 481 ★	

- Taken from yesterday's Graeme's:
  - http://indico.cern.ch/getFile.py/access?contribId=0&sessionId=0&resId=0&materialId=slides&confId=56580

Conclusions

- Parallel tape usage together with CMS and some LHCb activity
  - Very useful for exercising multi-VO sites
- PanDA and DDM driven workflow worked fine
  - Sites do nothing special for repro, similar workflow as the MC production
  - Bulk submission allow MSS reordering (good for robot efficiencies)
  - PanDA assigned-activated game ensure data pre-placement before job run
- Running simulation and reprocessing together can be potentially dangerous
  - Can block job slots for too long. Consider to restrict simul while reprocessing.
- dCache sites do need to bring attention to the MSS configuration
  - Avoid queued recalls
  - Tape drives - read pools balancing (MaxActive)
- STEP09 reprocessing was successful:
  - 5 out of 10 Tier-1s met enhanced metrics, 6 were validated (achieved baseline metrics)
  - 3 Tier-1s were above 90% of the target, one Tier-1 did 50%
- DDM team developing new pre-staging mechanism (file stager service)