ATLAS ANALYSIS PERFORMANCE ON THE GRID

monitoring and improving

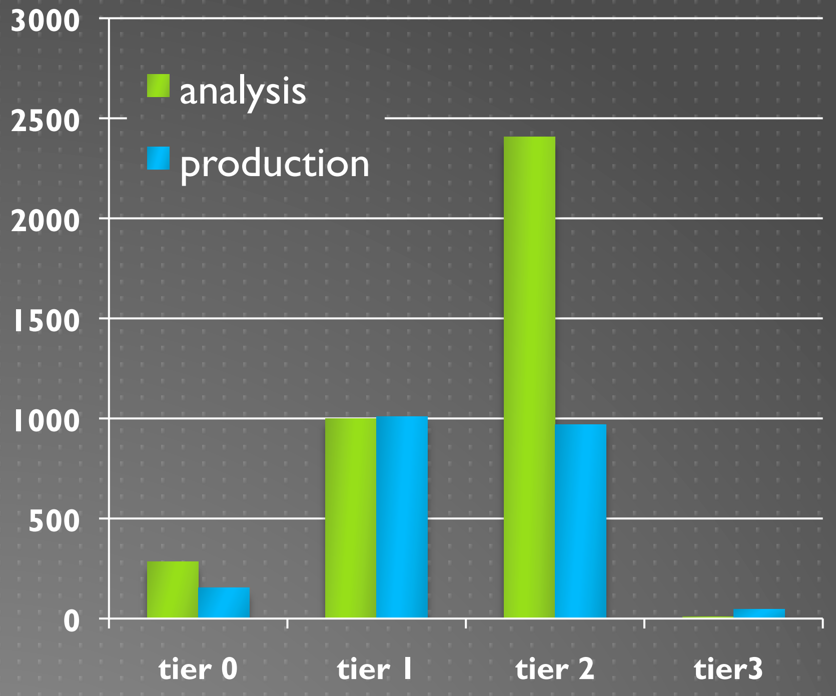
Ilija, Wahid, Doug, Jack and others on Atlas (esp. Hammercloud people)

WE WANT TO

- Know what is performance of ATLAS jobs on the grid
  - We don't have one widely used framework that we could instrument so we need to be open to any kind of jobs: root analysis scripts, athena jobs, d3pd maker
- Understand the numbers we get
- Improve
  - Our software
  - Our files
  - Way we use root
  - Middleware
  - Sites
- Way to test developments
- Have it as simple, realistic, accessible, versatile as possible
  - Running on most of the resources we have
  - Fast turn around
  - Test codes that are "recommended way to do it"
  - Web interface for most important indicators

WHY ANALYSIS JOBS ARE IMPORTANT ON ATLAS ?

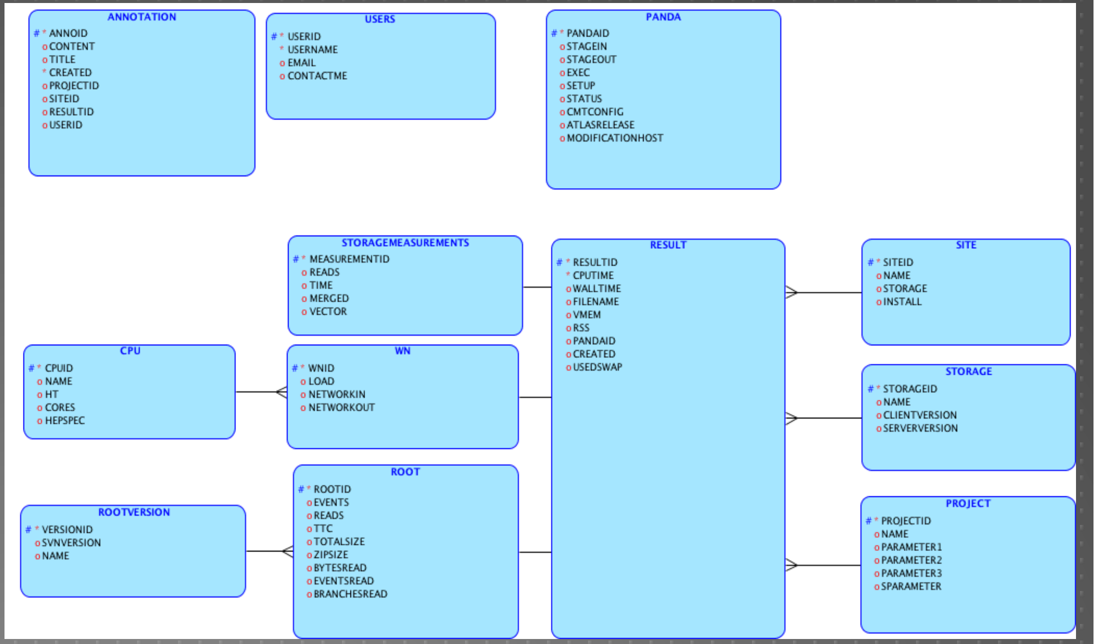
- Number of analysis jobs are increasing
- Production jobs are mostly CPU limited, well controlled, hopefully optimized and can be monitored through other already existing system
- Analysis jobs we know very little about and potentially could: be inefficient, wreck havoc at storage elements, networks.
- Results here focus on "D3PDs"
  - Plain ntuples – no POOL / custom T/P
  - "Final analysis stage" – most chaotic



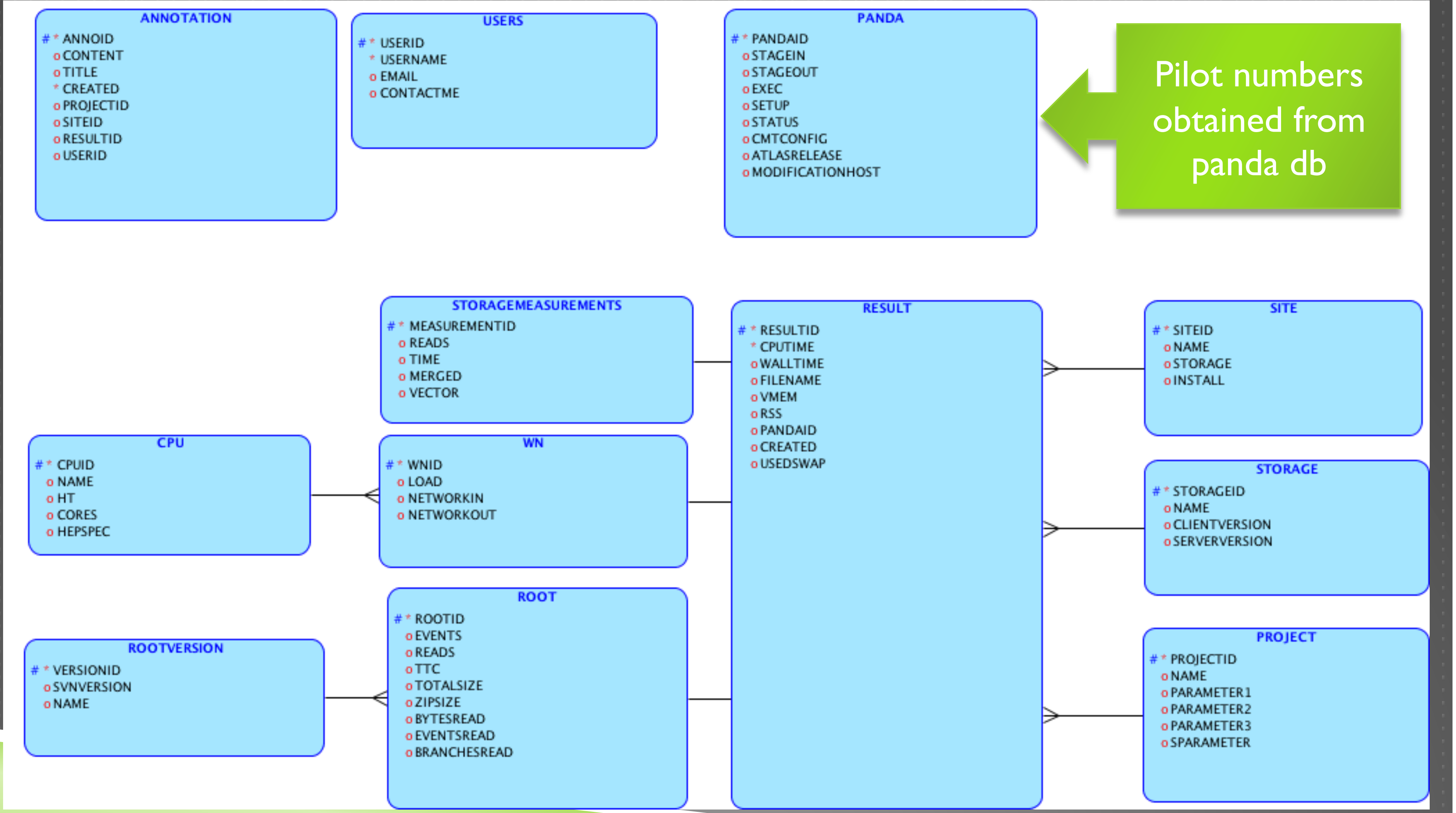
HOW ITS DONE

1. HammerCloud submits jobs
2. Jobs collects and sends info to DB

- Continuous
  - Job performance
    - Generic ROOT IO scripts
    - Realistic analysis jobs”
  - Site performance
  - Site optimization
- One-off
  - new releases (Athena, ROOT)
  - new features, fixes
- All T2D sites (currently 35 sites)
- Large number of monitored parameters
- Central database
- Wide range of visualization tools



```
ANNOTATION
# * ANNOID
  o CONTENT
  o TITLE
  * CREATED
  o PROJECTID
  o SITEID
  o RESULTID
  o USERID

USERS
# * USERID
  * USERNAME
  o EMAIL
  o CONTACTME

PANDA
# * PANDAID
  o STAGEIN
  o STAGEOUT
  o EXEC
  o SETUP
  o STATUS
  o CMTCONFIG
  o ATLASRELEASE
  o MODIFICATIONHOST

STORAGEMEASUREMENTS
# * MEASUREMENTID
  o READS
  o TIME
  o MERGED
  o VECTOR

RESULT
# * RESULTID
  * CPUTIME
  o WALLTIME
  o FILENAME
  o VMEM
  o RSS
  o PANDAID
  o CREATED
  o USEDSWAP

SITE
# * SITEID
  o NAME
  o STORAGE
  o INSTALL

CPU
# * CPUID
  o NAME
  o HT
  o CORES
  o HEPSPEC

WN
# * WNID
  o LOAD
  o NETWORKIN
  o NETWORKOUT

STORAGE
# * STORAGEID
  o NAME
  o CLIENTVERSION
  o SERVERVERSION

ROOTVERSION
# * VERSIONID
  o SVNVERSION
  o NAME

ROOT
# * ROOTID
  o EVENTS
  o READS
  o TTC
  o TOTALSIZE
  o ZIPSIZE
  o BYTESREAD
  o EVENTSREAD
  o BRANCHESREAD

PROJECT
# * PROJECTID
  o NAME
  o PARAMETER1
  o PARAMETER2
  o PARAMETER3
  o SPARAMETER
```



ANNOTATION
* ANNOID
o CONTENT
o TITLE
* CREATED
o PROJECTID
o SITEID
o RESULTID
o USERID

USERS
* USERID
* USERNAME
o EMAIL
o CONTACTME

PANDA
* PANDAID
o STAGEIN
o STAGEOUT
o EXEC
o SETUP
o STATUS
o CMTCONFIG
o ATLASRELEASE
o MODIFICATIONHOST

STORAGEMEASUREMENTS
* MEASUREMENTID
o READS
o TIME
o MERGED
o VECTOR

RESULT
* RESULTID
* CPUTIME
o WALLTIME
o FILENAME
o VMEM
o RSS
o PANDAID
o CREATED
o USEDSWAP

SITE
* SITEID
o NAME
o STORAGE
o INSTALL

CPU
* CPUID
o NAME
o HT
o CORES
o HEPSPEC

WN
* WNID
o LOAD
o NETWORKIN
o NETWORKOUT

STORAGE
* STORAGEID
o NAME
o CLIENTVERSION
o SERVERVERSION

ROOT
* ROOTID
o EVENTS
o READS
o TTC
o TOTALSIZE
o ZIPSIZE
o BYTESREAD
o EVENTSREAD
o BRANCHESREAD

ROOTVERSION
* VERSIONID
o SVNVERSION
o NAME

PROJECT
* PROJECTID
o NAME
o PARAMETER1
o PARAMETER2
o PARAMETER3
o SPARAMETER

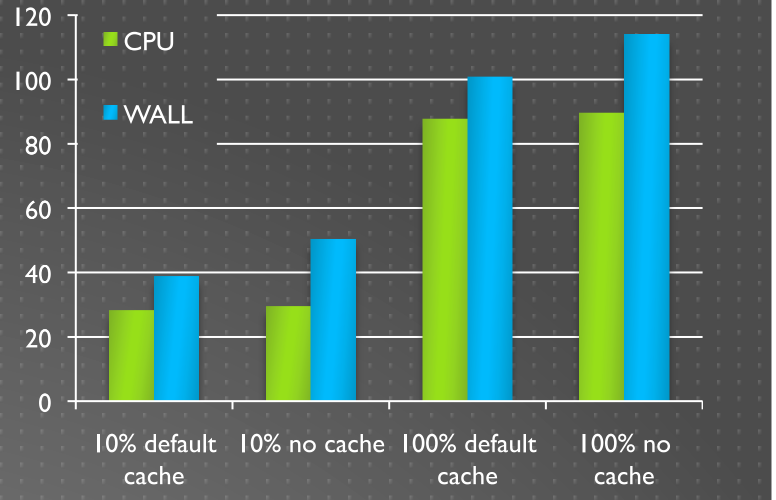
Pilot numbers obtained from panda db

MESSAGE FOR ROOT I/O GROUP

- Everybody
  - Visit http://ivukotic.web.cern.ch/ivukotic/HC/index.asp
  - Give it a spin, give us feedback and ask for features
- ROOT / CMS people
  - Give us you code/data and we do fast testing for you on all different kinds of CPUs / storage backends / protocols.
  - We'll learn something from your tests too.
  - Please give us any relevant improvements – e.g. OptmizeBaskets improvements (one of the original use-cases for this !)

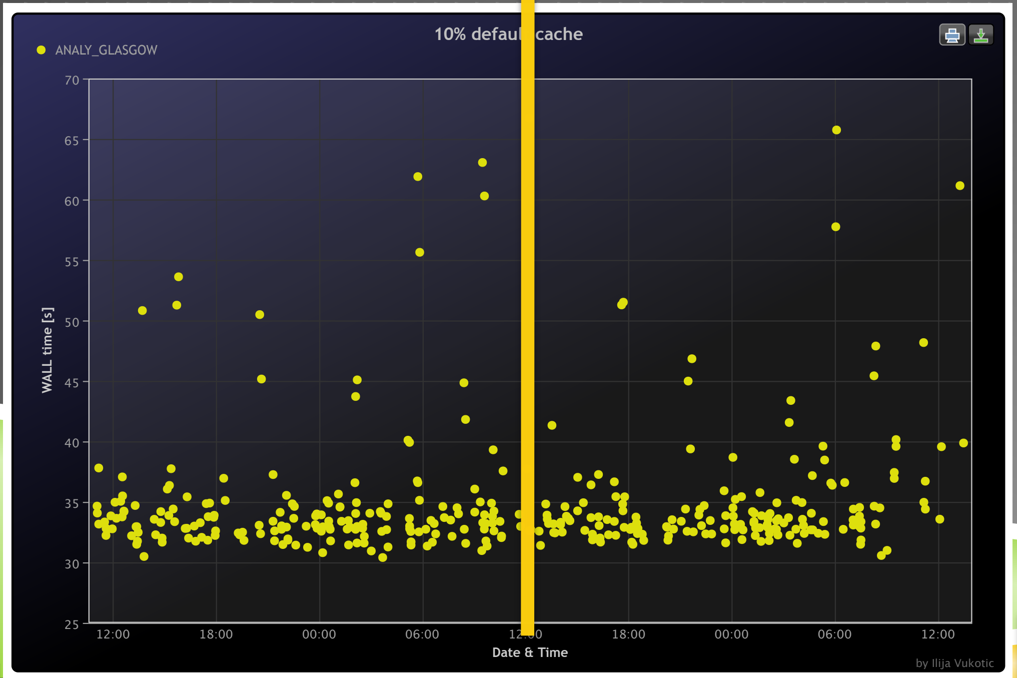
RESULT – EFFICIENCY

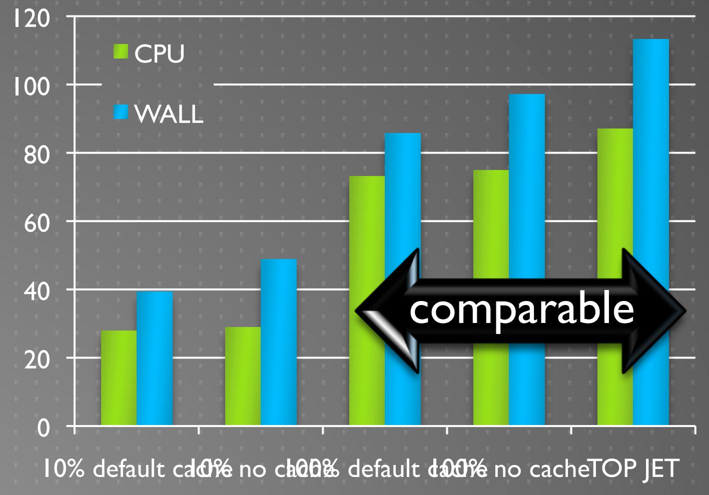
- Average results over all sites using 17.0.4 (ROOT 5.28)
- 77% Event loop CPU efficiency
- Since updated to 17.1.4 then 17.2.0 (ROOT 5.30)



ROOT 5.28 (Athena 17.0.4)

ROOT 5.30 (Athena 17.1.4)

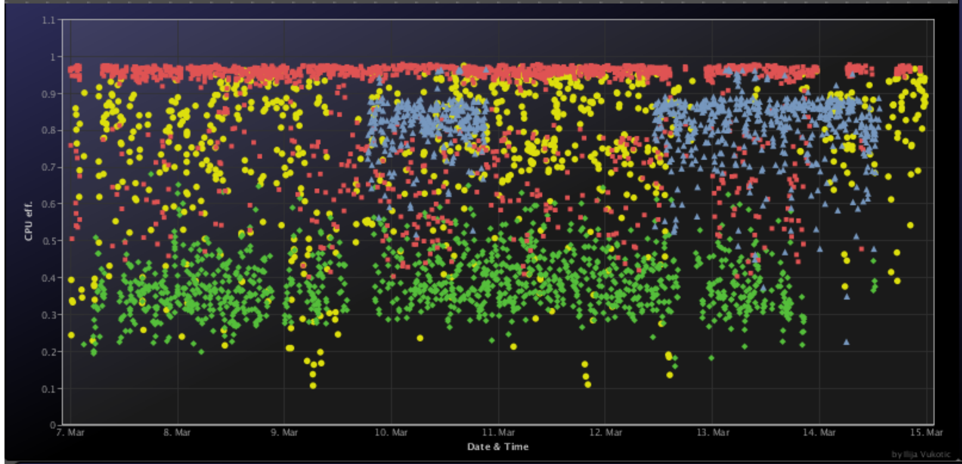




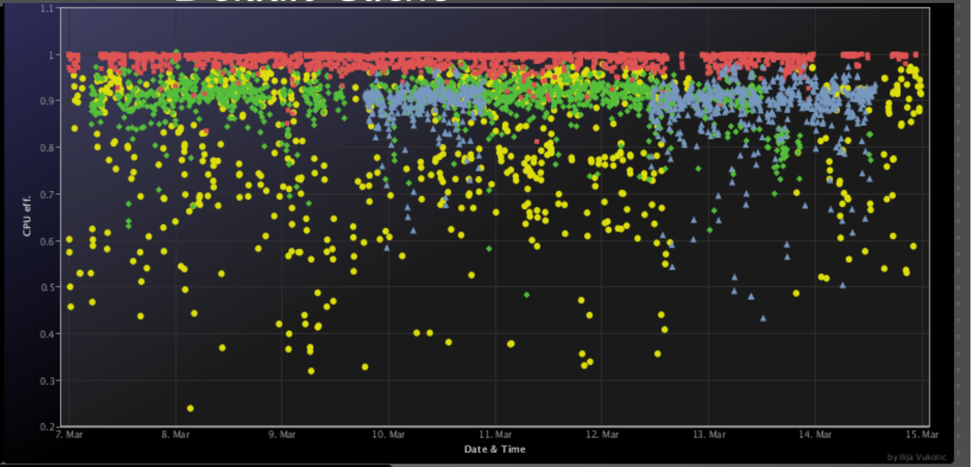
Realistic analysis

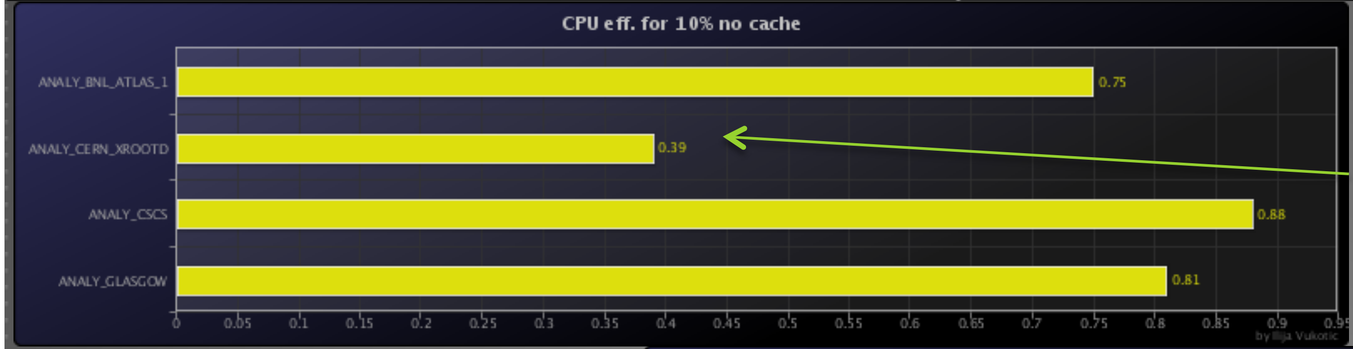
RESULT – EFFICIENCY OF TTC CPU EFF. 10% OF EVENTS READ

No Cache

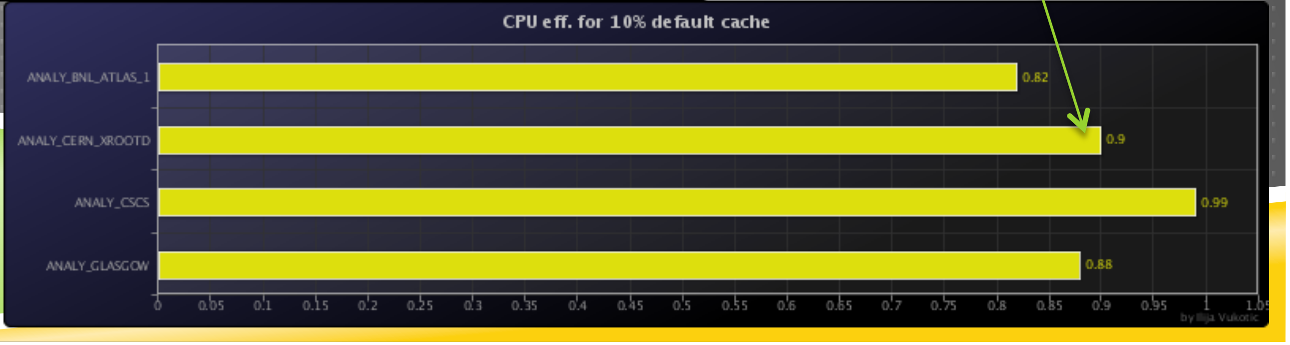


Default Cache



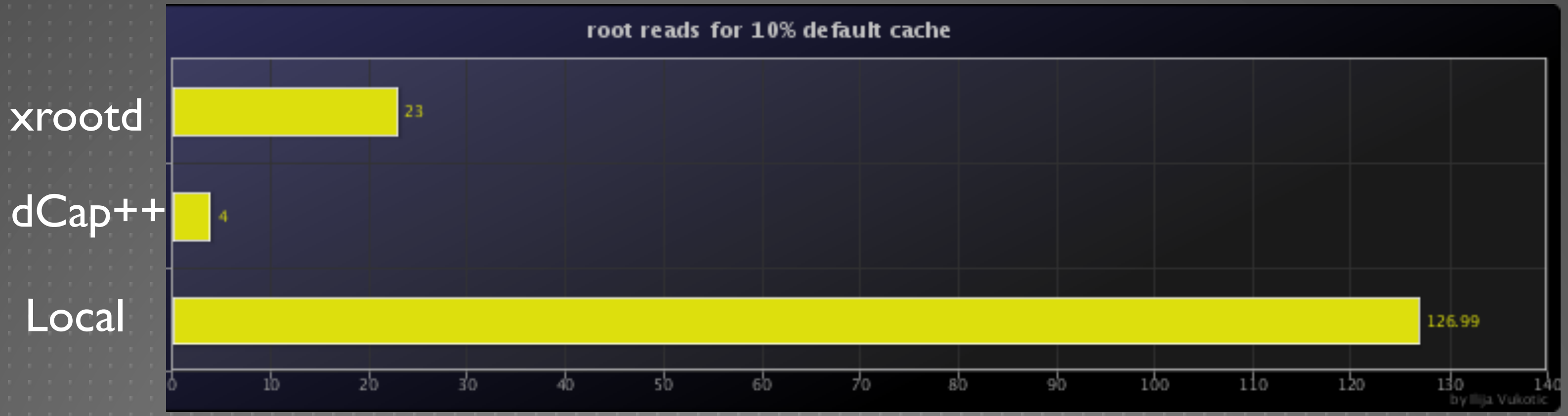


- For EOS (xrootd) it is essential



RESULT – EFFICIENCY OF TTC

- TTC more pronounced over WAN – we have seen this already and now adding WAN tests to framework
- Quite different number of reads depending on storage system / protocol



ONGOING WORK

- Stress tests
- WAN tests
- Optimizations for sites – example: is it better to pre-stage input files?
- Performance of different storages/protocols
- Optimal compression levels
- Optimal autoflush / TTC settings?
- Performance of 5.32 ROOT version (not yet used in atlas)
- And please give us some other things to test !!