PMB introduction and performance overview

Thomas Kittelmann
University of Pittsburgh
ATLAS S&C week Core/PMB session 01-12-2010

Overview

- Update on performance tools + data on http://atlaspmb.cern.ch/
- VMEM/CPU status for 2011
- Per-domain results from private tests.

Provided on atlaspmb.cern.ch

- Dozer-based monitoring of RTT jobs
  - Job subtraction gives global per-domain breakdowns
  - VMEM changes in domain triggers email alert
- Long-term archive of relevant RTT job output
  - Log files, perfmon summary files, etc.
- Archive-based monitoring (NEW)
  - More flexible than Dozer-based monitoring
  - So far more robust
  - For now focus on complementing Dozer-based monitoring:
    Use per-alg perfmon numbers and transform timers rather than job-subtraction.
- Various oprofile results

Historical tag info

- Since ~may (with one disruption) NICOS stores tags used in builds in txt files on AFS
- Python script provides easy access for PMB investigations, providing answers to questions such as “why is this PMB plot showing a change in VMEM 2 weeks ago?”

```
(tkittel@lxplus438 ~)> /afs/cern.ch/atlas/project/pmb/bin/pmb-taginfo --diff=16.0.X:2010-11-09:2010-11-11
Finding tag differences between 16.0.X:2010-11-09 and 16.0.X:2010-11-11:
  Packages with different tags:
    00-01-52    00-01-53    PhysicsAnalysis/D3PDMaker/EventCommonD3PDMaker
    00-06-20    00-06-21    PhysicsAnalysis/D3PDMaker/SUSYD3PDMaker
    00-02-87    00-02-88    PhysicsAnalysis/PATJobTransforms
    00-02-53    00-02-58    PhysicsAnalysis/StandardModelPhys/PhotonAnalysisUtils
    00-05-58    00-05-59    TileCalorimeter/TileMonitoring
(tkittel@lxplus438 ~)>
```

PerfMon status
(apologies to Sebastien for inaccuracies)

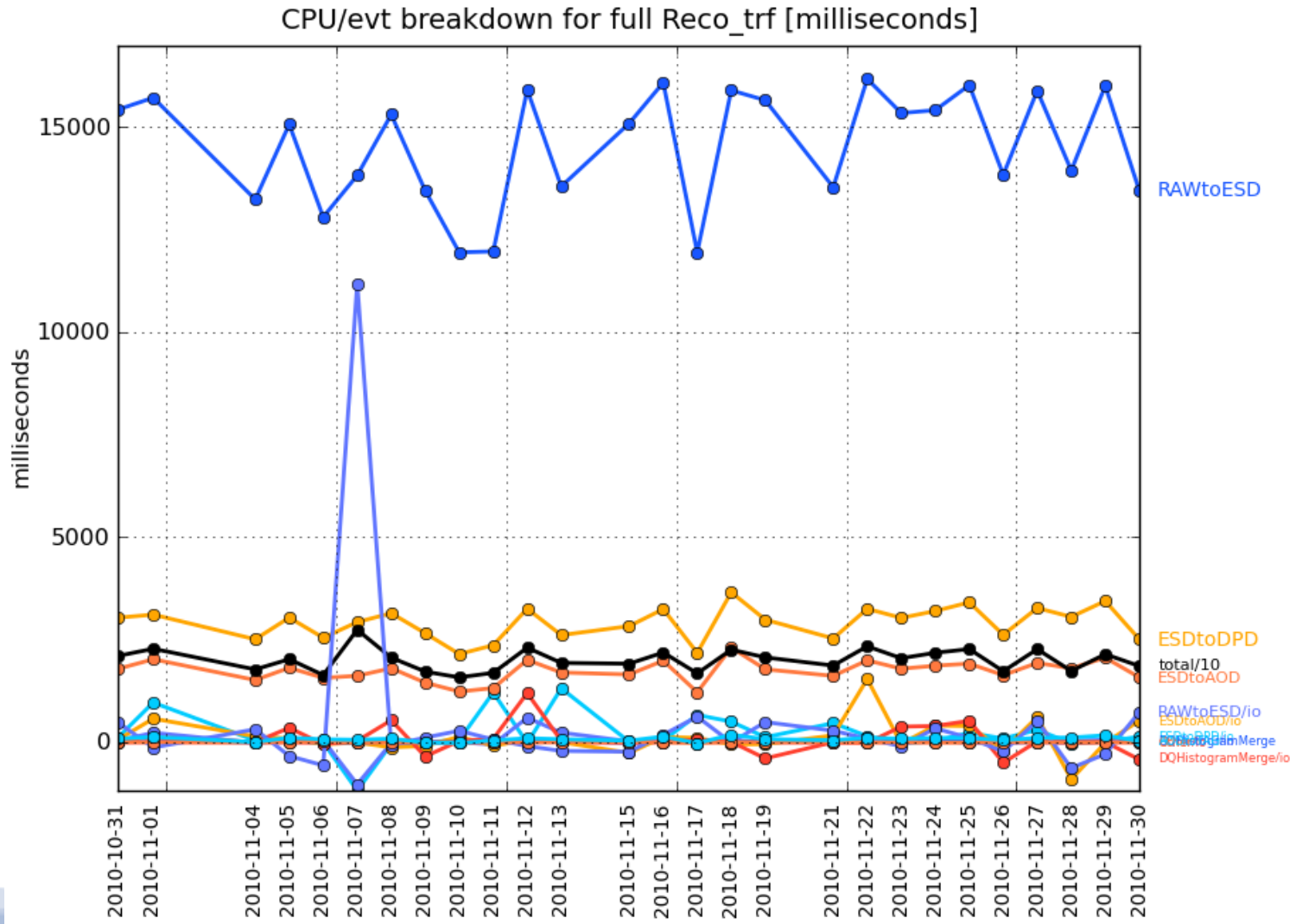
- New features (that I know of):
  - Per-alg alloc|free numbers now listed in output summary txt files.
  - Approximate mem-leak provided even in non-detailed mode.
- PerfMon doesn't work in recent nightlies (due to Gaudi bug). Hopefully fixed soon.
- Hopefully to come:
  - Try to avoid double-counting such as when one alg's ::initialize() triggers a svc initialization.
  - Use alg2domain mapping to provide approx per-domain summaries.

VMEM/CPU status

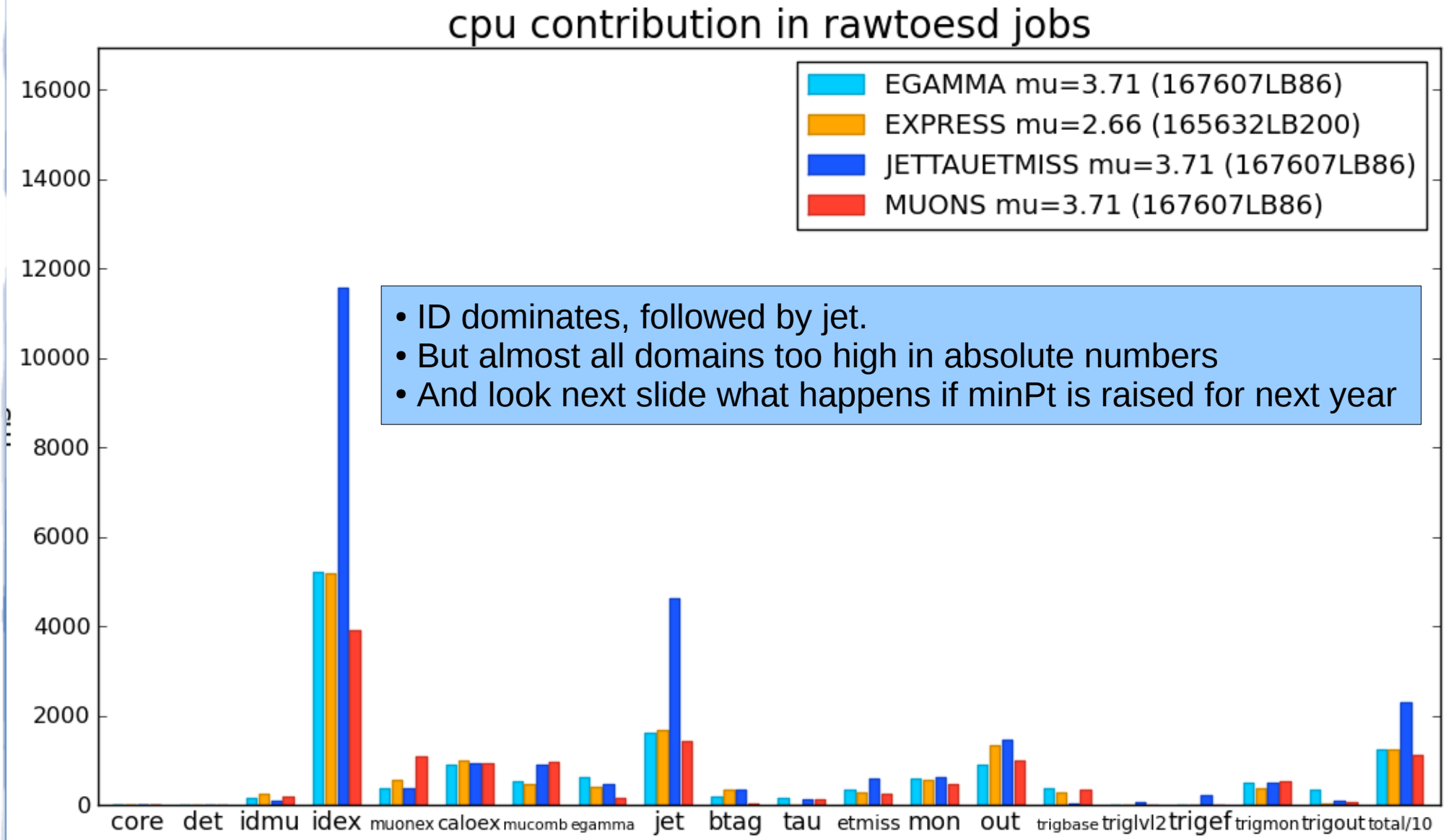
- VMEM remains a problem ~2.5GB in pp rawtoesd. This is the limit where crashes start! (situation in HI even worse). Goodbye multi-core.
- CPU usage an increasing concern:
  - Currently we use ~20 seconds/evt depending on stream and conditions. Used to be lower, but events get more and more interesting all the time with higher lumi.
  - Physics push for evt rates in 2011 of ~400Hz (even 600Hz) compared to nominal 200Hz.
  - Note that data-taking time will go from small fraction in 2010 (~10%?) to most of the time (>~50%?) in 2011 => can no longer hide the problem.
  - Can expect (slightly) higher pileup + perhaps 75ns bunch spacing.
  - Part of the “solution” will be to raise the ID tracking pT threshold, but it likely won't be enough.
  - CPU spent in many places => no golden bullet.
- If VMEM was 500MB lower we would gain 14% CPU (7->8 jobs/node @ T0)
- Need to improve BOTH CPU and VMEM across the board to serve the needs of physics communities!!!

Overview of Reco_trf time usage

- With low tracking pT threshold, rawtoesd dominate again
- ESDtoDPD can supposedly be fixed to save 0.5-1.0 s/evt (Karsten looking at it)

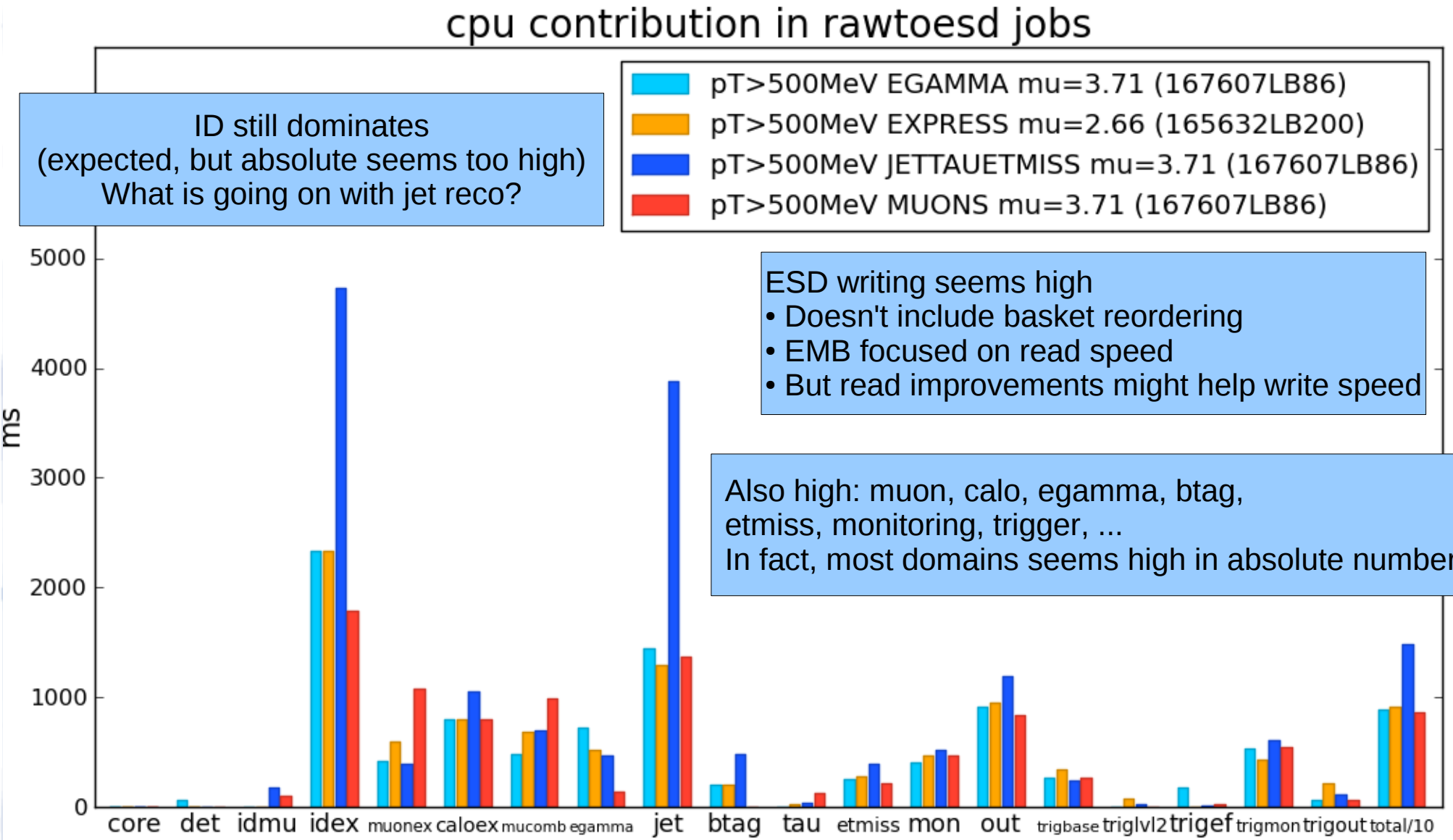


Domain breakdown (minPt=100MeV like at T0)



- ID dominates, followed by jet.
- But almost all domains too high in absolute numbers
- And look next slide what happens if minPt is raised for next year

Domain breakdown (minPt=500MeV, probable scenario for next year)



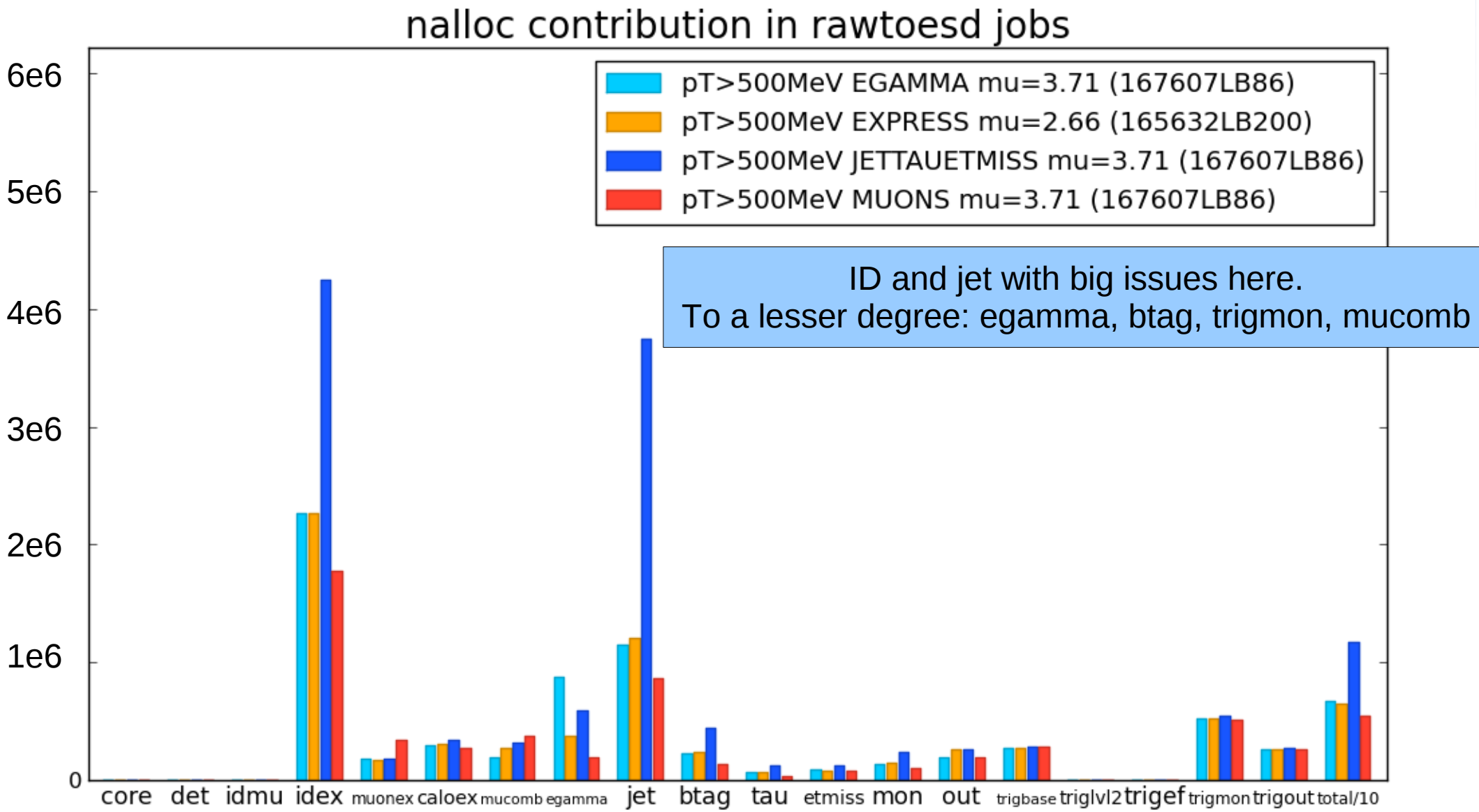
ID still dominates
(expected, but absolute seems too high)
What is going on with jet reco?

ESD writing seems high
- Doesn't include basket reordering
- EMB focused on read speed
- But read improvements might help write speed

Also high: muon, calo, egamma, btag,
etmiss, monitoring, trigger, ...
In fact, most domains seems high in absolute numbers!!

Rumour: CMS total is 2s/evt!!!

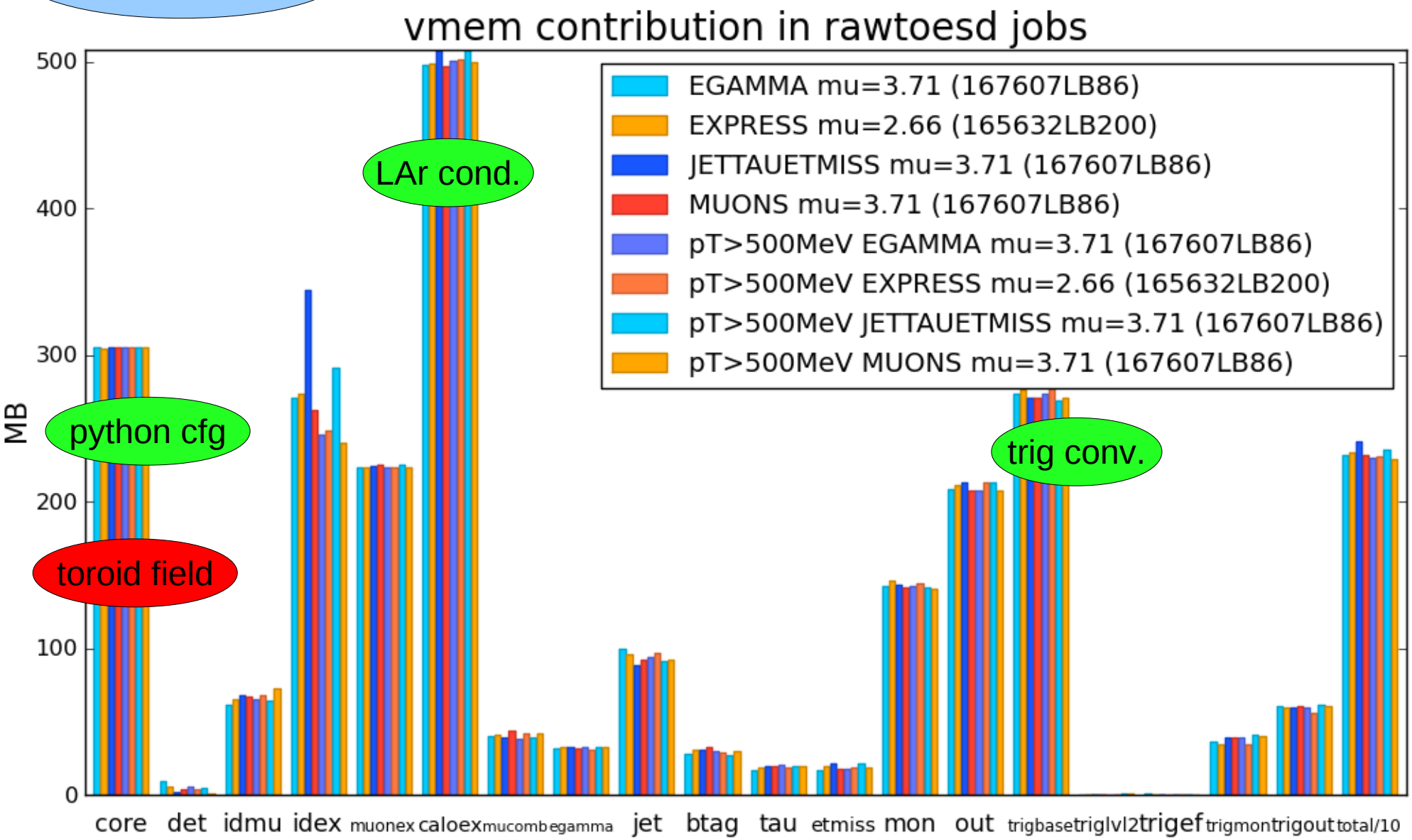
#allocations each event



ID and jet with big issues here.
To a lesser degree: egamma, btag, trigmon, mucomb

VMEM contributions

Incoming big changes
Indicated in red/green



Only ID seems to depend on stream or pT threshold

Incoming large VMEM changes

- +165mb: Move to asymmetric toroid field map, due to demand from muon reco.
- -150-200mb(?): Move to smaller Lar conditions (assume detector timings known => less cond. needed per channel)
  - What is holding this back?
- Reduce memory impact from python cfg step:
  - Two slightly different approaches (separate process vs. memory cleanup) from Seb./Wim
  - I would hope to have one in prod. in rel 17 => would be good to have both available as option soon.
- Stop preloading 150mb(?) of trigger libs.
  - Seb. did footwork, need trigger to adopt patches.
- ~90MB of potential gains in ID HI reco yet to be adopted
  - Halted by lack of ID effort + SGDeleteAlg issue with IDCs
- Overall looks OK with LAr cond., but target is -500MB so need more.

Conclusions

- CPU and VMEM continues to be an issue for ATLAS reconstruction
  - First price to pay for this will be the raising of the ID tracking pT threshold (100->500 MeV).
  - Next price will be the limit on data taking rate in 2011.
- Let us try to avoid this! But can only happen if more focus is brought on the task.
  - It seems that there is a mindset everywhere that using a lot of resources is not so bad as long as someone else is using more. This is not really helpful...
- After such stern words comes the domain reports from the people for whom they were not intended! :-)