Status des tests multi-cores au CC-IN2P3

Sommaire

- Position de CMS et choix techniques
- Historique
- …et problèmes rencontrés
- Bilan (rapide) des performances
- Résumé et perspectives

Position de CMS et choix techniques

Approche de CMS pour le multi-core :

- un job « maître » qui créé des processus fils sur chaque cœur (lit le /proc/cpuinfo) ➡ *indépendance du système de batch utilisé*.
- permet le partage de la mémoire grâce aux *POSIX IPCs* (outils de communication entre processus : sémaphores, mémoire partagée et queues de message).

➡ Choix du « whole node » pour les tests mais possibilité de calcul en *milieu virtualisé ou Cloud (l'un des critères principal du choix technique)* car le nombre de cœurs peut-être spécifier au workflow multi-core (option possible au niveau du WMAgent).

Les jobs multi-core sont des *jobs de reconstruction*, la mémoire partagée entre les processus est composée du *code lui-même, de la géométrie du détecteur et des conditions de reconstruction* (alignement et calibration des détecteurs).

Historique

04/11

- Lors d'un workshop « CMS usage of T1s sites », CMS demande la création d'une queue dédiée pour des tests de jobs multi-cores.

08/11

- « Whole node » en production au CC fin août 2011, avec 2 machines réelles, et 2 virtuelles (« use-case » réel pour l'étude de l'overhead de la virtualisation)

  ⇨ machines 8 cœurs, 16 Go (15,2 Go pour MVs), derrière cccreamceli01.

10/11

- Pas d'activité en septembre et octobre (plusieurs grosses productions ont « occupé » les gens) → reprise début novembre.

11/11

- Un peu d'activité en novembre → vrai début des tests (pendant 1 mois environ, essentiellement concentrés sur quelques sites).

⇨ Peu d'activité au CC…

Problèmes rencontrés

01/12

- Jobs qui consomment jusqu'à 30 Go de mémoire, en fait 2 problèmes :
  - CMS avait demandé 3 Go/cœur (~2,2 Go par cœur, ~18 Go au total) → 9/10 jobs sont donc tués ;
  - Bug dans le job « logcollector », qui passe en 1-core, et qui consomme trop de mémoire : problème également observé sur d'autres sites, et ensuite corrigé (étude très difficile au CC car pas de monitoring temps réel pour les jobs, et impossible de contrôler la mémoire des jobs running…).

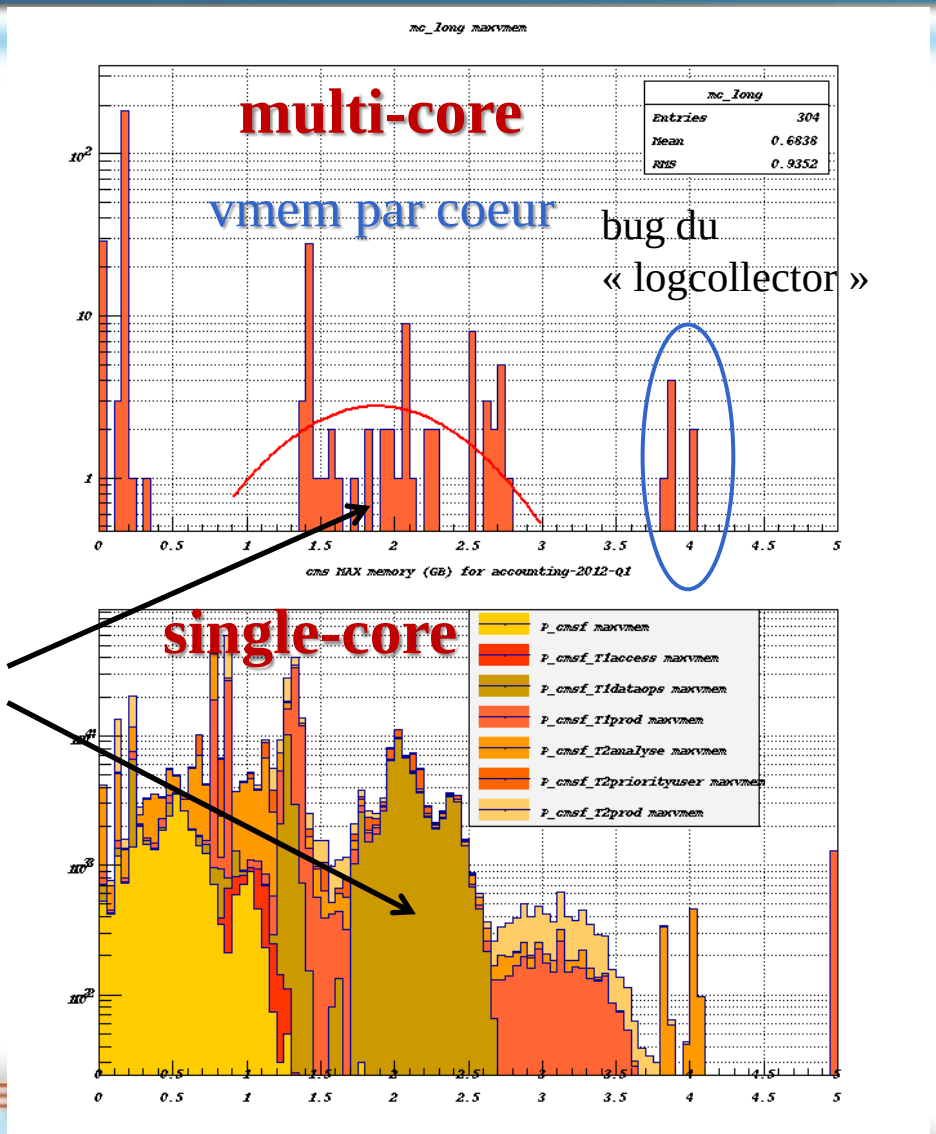
02/12

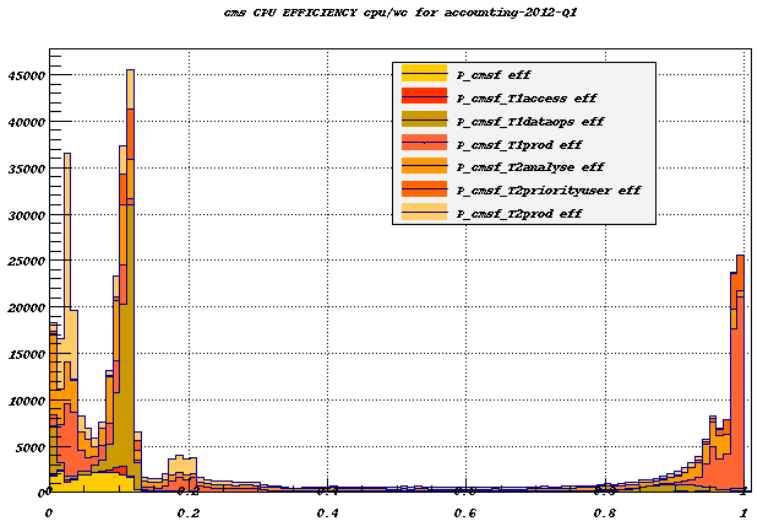
- Nouvelle ferme : 16 cœurs, 3 Go/cœur, 3 physiques et 3 MVs. Problèmes :
  - Installation et utilisation du code CMS
  - Staging impossible suite à la mise-à-jour de dCache (2nde « golden » release, arrêt du 7 février), résolu par un staging manuel (puis par Yvan, voir ticket GGUS #79729).

03/12

- Les jobs multi-core s'exécute enfin correctement (presque !)…

Bilan (succinct) des performances





Note : rien à dire sur les VMs

- pas de VMs en production en ce moment
- Profil smurf très étrange des VMs lors de leur utilisation en janvier/février (impossible de s'y connecter, les jobs tournaient très très longtemps sur ces machines)...

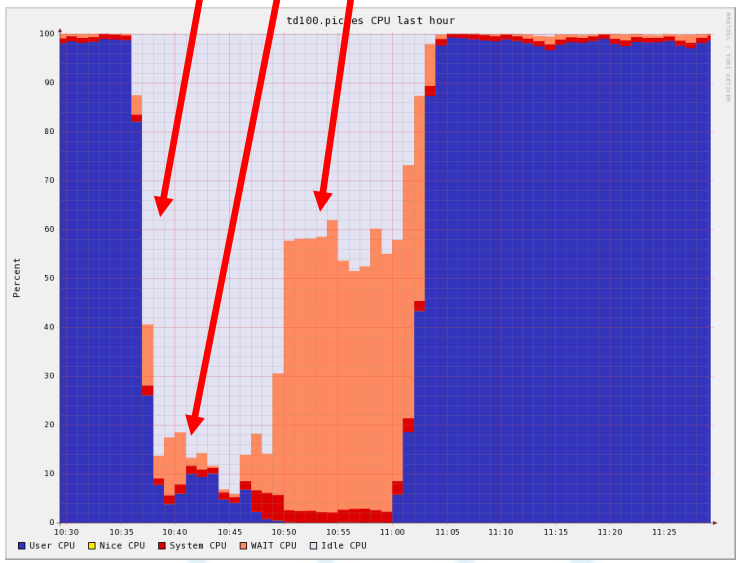
➡ Difficile de comparer les performances…

Bilan (succinct) des performances

End of event processing

merging

Lazy download



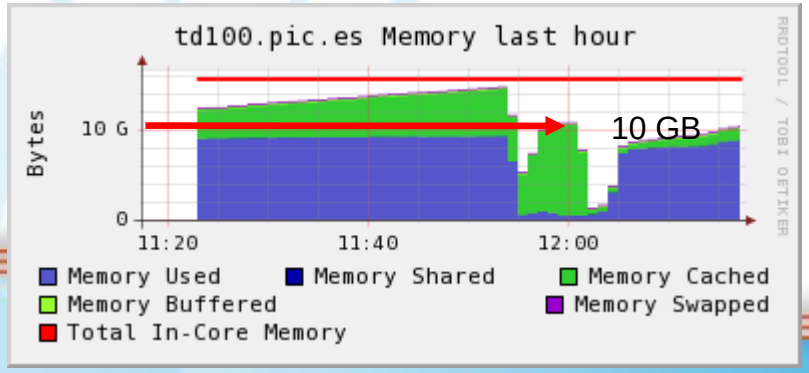
Tests de novembre 2011

- Logiciel CMSSW_4_4_2_patch2
- Reconstruction complète : AOD, RECO, DQM

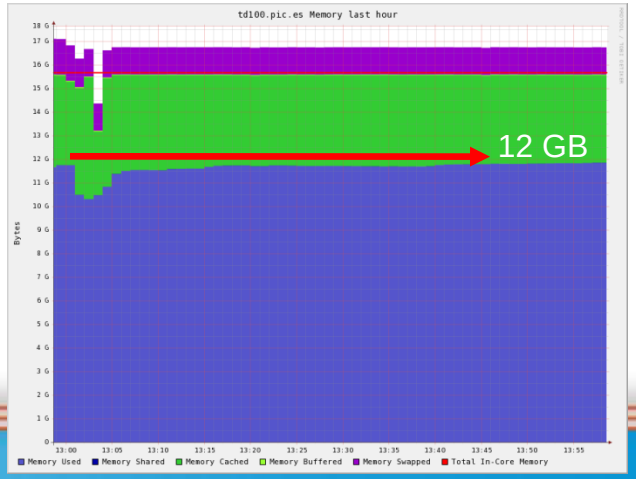
Quelques résultats:

- Inefficacité CPU due au merging, et au download des fichiers d'input
- Gain en terme de mémoire de près de 20% (RSS=1,5Go par enfant)

1 job multi-core 8 cores



8 jobs single-core



Résumé & perspectives

- CMS a montré *un gain notable* dans l'utilisation de la mémoire dans leur approche du multi-core (*près de 20% de gain*).
- Au niveau du CC, les performances *restent floues (manque de statistique),* mais les jobs multi-core s'exécute enfin correctement ! Mais ce n'est que le début de cette aventure…
- Pour suivre les recommandations des TEGs, CMS veut modifier son approche du multi-core pour pouvoir *l'exécuter dans les fermes de calcul classique* (i.e. sans « whole-node ») : CMS spécifie *le nombre de cœurs à utiliser*, et le système de batch doit lui réserver les ressources demandées.
- Ce qui devrait être relativement facile de mise en place au CC (GE gère déjà ce type de queue mc), création de queue à nombre de cœurs fixe, ou un autre mécanisme pour passer le bon nombre de cœurs à GE ?