



Performance des machines de calcul au CC-IN2P3



Yannick Perret



Préambule



Évaluation de la puissance des machines :

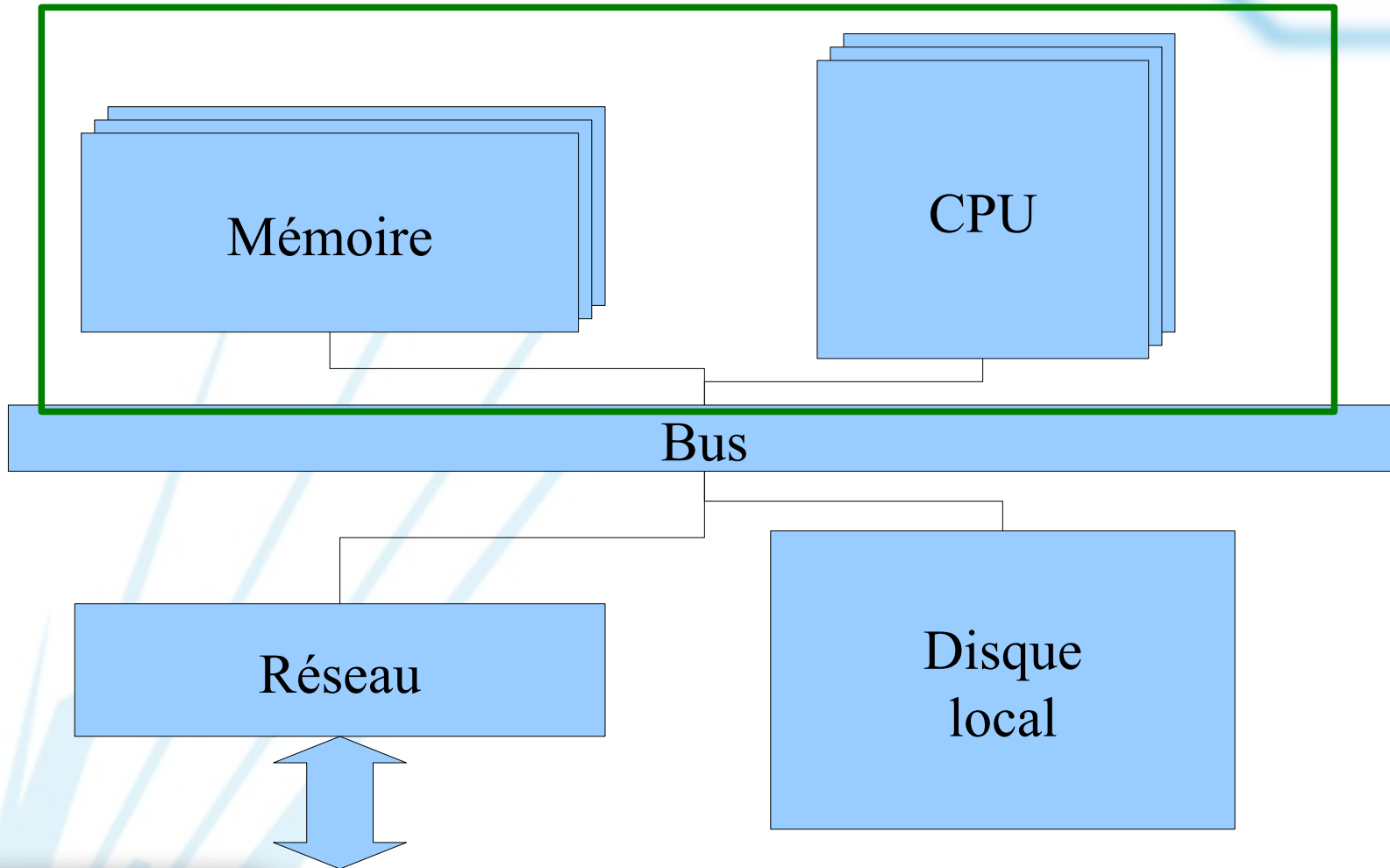
- Nombreux benches existants
- La communauté LHC : SPEC (avec certaines options)

→ SPEC-HEP06

■ Limites :

- Uniquement CPU (et un peu architecture mémoire)
- Pas de disque
- Pas de réseau
- Pas d'interaction entre les composants
- « Moyenne » de tests spécifiques

Une machine...



Contexte déclencheur



Problèmes de latence / durée d'exécution sur certains jobs :

- Phénomènes « aléatoires »
- Symptômes « diffus »
- Pas de problèmes techniques

■ Diagnostiques difficiles :

- Flous coté utilisateurs et exploitation
- Difficilement reproductible
- Mélanges entre plusieurs problèmes distincts

→ Pointait vers un problème de perf. sur certains systèmes de fichiers

Limitations des performances ?



- Coté serveurs :
 - Réactivité des (méta-)données
 - Bande passante (réseau, disque...)
 - Efficacité logiciels serveurs

- Coté clients :
 - (limites serveurs)
 - Limites d'I/O locales (réseau, disque, bus...)
 - Paramétrages (taille cache, configuration...)

▶ Dépasser les limites



- Côté serveurs : + = mieux !
 - Plus de serveurs de (méta-)données
 - Serveurs plus performants (CPU, réseau, disque...)
 - Plus de réseau (débit, latence)

 - Côté clients : plus délicat
 - Machines plus performantes (CPU, réseau, disque...)
 - Adapter la configuration à l'utilisation
 - Augmenter le cache local, la priorité...
- effets secondaires difficiles à prévoir !

▶ Effets secondaires (clients)



Effets secondaires variés :

- Pas toujours d'utilisation « type » (pas de paramètres universels)
- Machine + puissante $\Rightarrow$ programmes + rapides (+ de sollicitations)
- Cache plus gros $\Rightarrow$ plus de traitements dans le client
- Coût : serveurs $\rightarrow$ quelques machines ; clients $\rightarrow$ +1000 machines

Et surtout :

Aucun FS n'a une réponse linéaire en fonction du nombre d'accès concurrents !

« Nouveau » : complexité algorithmique des traitements

Évolution des machines



Il y a 10 ans : 2 « processeurs »

Aujourd'hui : 24

Demain : 48, 64...

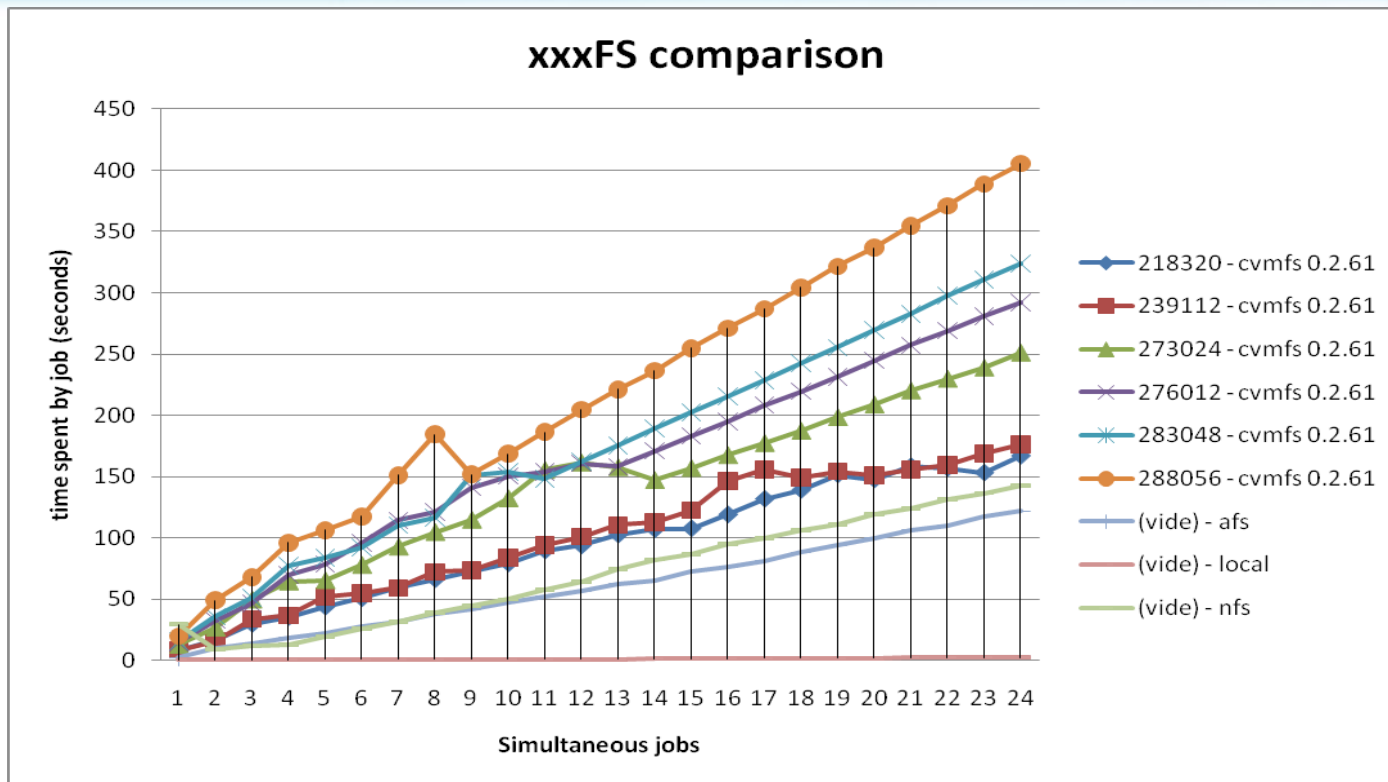
Multiplication des cœurs → capacité à générer une plus forte charge (simultanée) sur les ressources locales, y compris les FS

Réponse linéaire ou plus que linéaire selon la charge

- augmentation des temps de réponse

- on dépasse les seuils de « perception »

Quelques exemples



Mêmes comportements pour AFS, NFS, CVMFS, ...

Pistes...



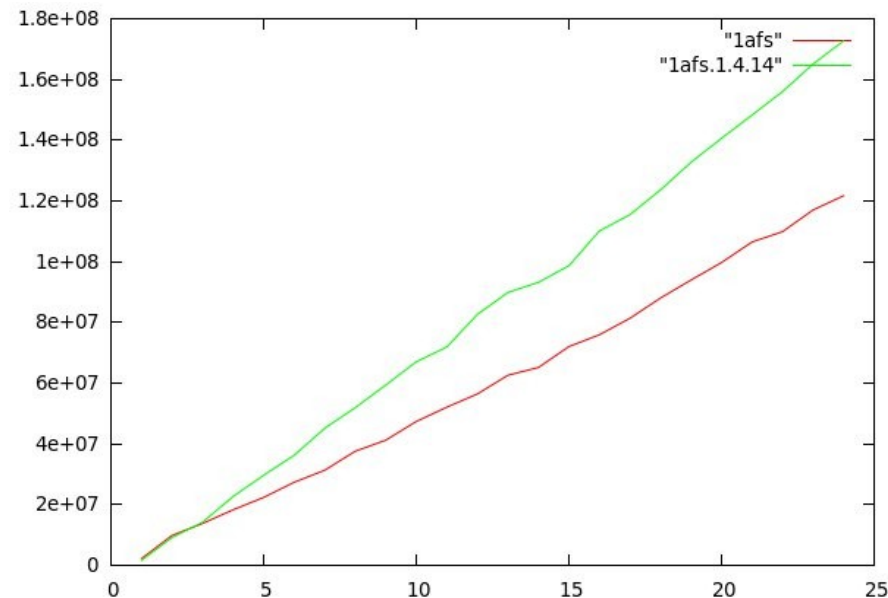
Dépend de nombreux paramètres !

- Type de charge (moyenne / max)
- Caractérisation des types de jobs
- Impact des changements (coûts, compatibilité...)

Parfois une « simple » mise à jour peut suffire

→ nécessite tout de même de nombreux tests et validations...

Autre exemple : NFSv4 sur SL6



Pistes... (2)



- Latence / débit réseau (dcache...)
- Performance disque : écriture transferts, lectures...
- Conflit d'accès disque scratch / caches FS (AFS, CVMFS)
- Impact de l'hyperthreading (nombreux sites sans HT)
- Jobs mal conçus (« setup », compilations à l'exécution...)
- Jobs sous-optimaux (travail dans le AFS du CERN, en 32bit, duplication des binaires/librairies...)
- Gestion des jobs (répartition par user / groupe / dans le temps...)
- Kernel linux :
 - Tests avec soft en local → toujours pas optimal
⇒ pointe vers une contention sur le VFS
 - Kernel RHEL5 « vieux » (malgré les patchs RedHat)
- Autres ?



Solutions ?



- Optimiser les codes utilisateurs (utopique ? Pas toujours...)
- Répartir les accès sur des FS différents
- Répartir les accès sur des disques différents
- Utiliser des FS locaux
- Dédier des services / machines par utilisation
- Virtualisation (diviser pour mieux régner)
- Gestion plus fine des jobs (type de ressources utilisées)



Analyses en cours

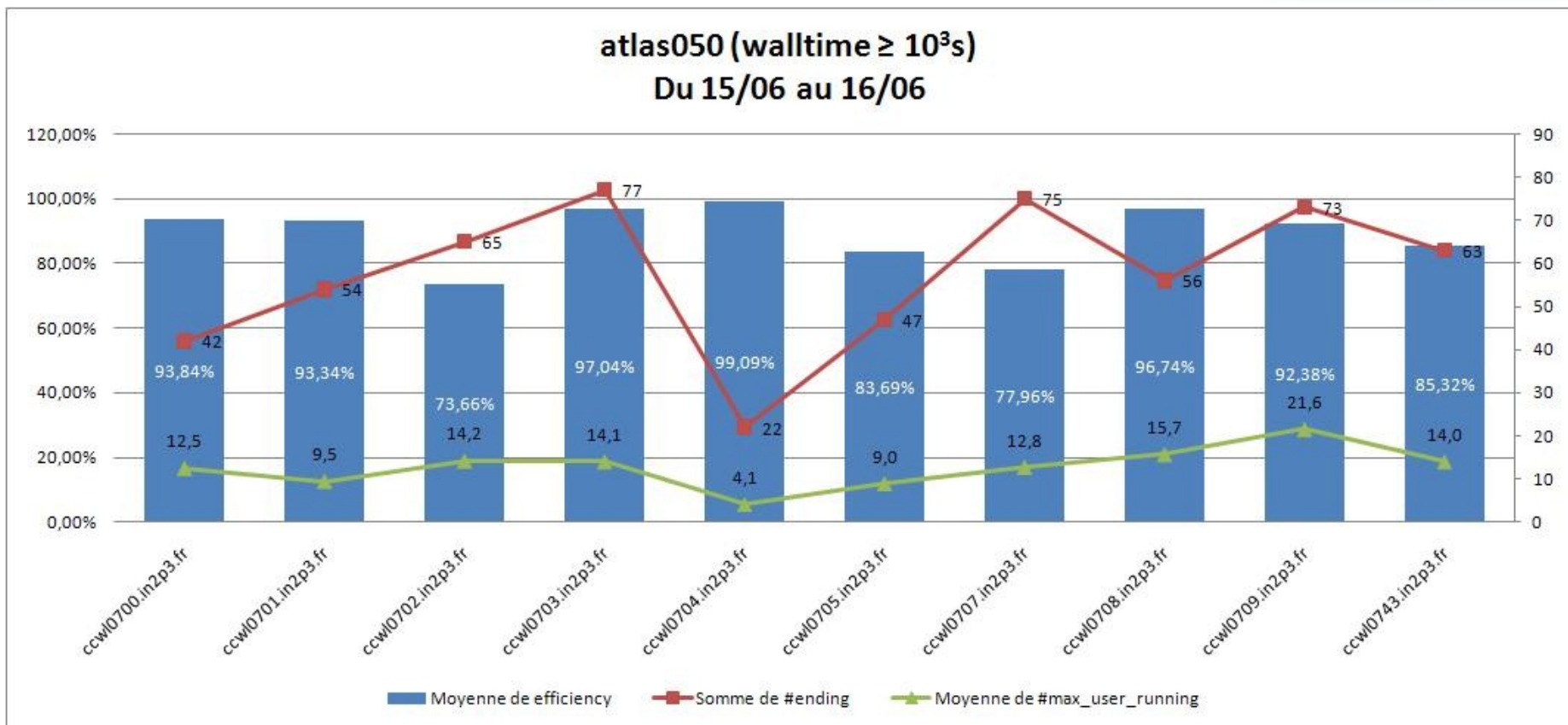


- Caractériser les différentes pistes, dans des conditions similaires :
 - Tests des soft-area en local (synchronisation !)
 - Tests de /scratch séparés (disque en plus, plus rapide, ...)
 - Tests de kernels plus récents

Difficultés :

- Tests longs (statistiques valides)
- Représentativité des tests (jobs identiques...)
- Ne changer qu'une chose à la fois

Quelques résultats (tous chauds)



0-std+17+dd ; 1-std+12 ; 2-std+17+* ; 3-new+17+stripe ; 4-std+5 ; 5-std+12+noHT ; 6-std+17 ; 7-new+17 ; 8-new+23 ; 9-std+17+cvmfs



Conclusions

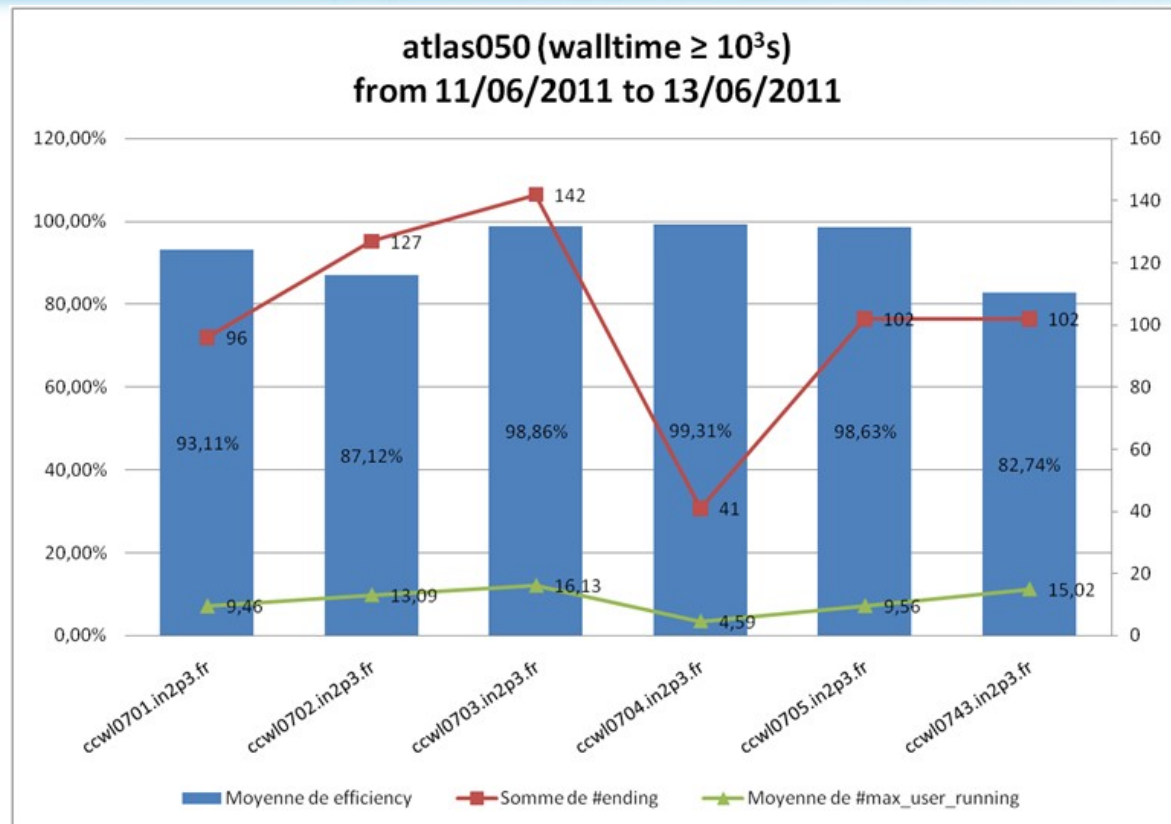


Pas de conclusion !

Il reste à faire :

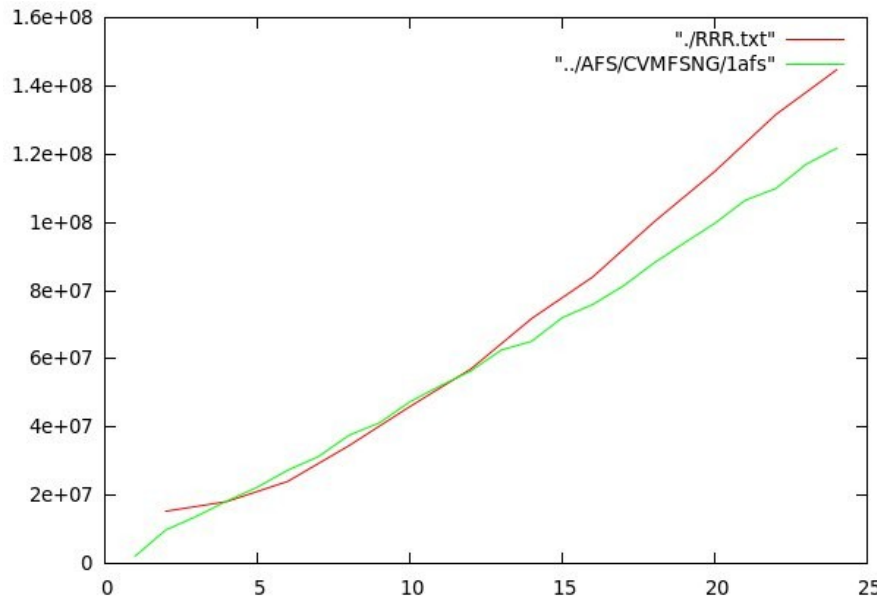
- Renforcer les statistiques
- Valider leur pertinence (type de job...)
- Combiner les différentes possibilités
- Valider l'impact sur les autres jobs
- Estimer les conséquences (coût, production...) de l'amélioration de l'efficacité

Quelques résultats



Dans l'ordre : 12wp atlas ; standard (tous) ; 17wp kernel+scratch ;
5wp atlas ; 12wp atlas (sans HT) ; 17wp atlas + CVMFS

Autres tests



VM / machine physique

AFS / NFS v4.0 / local

