

« Elapsed Time » pertinent pour de la facturation ?

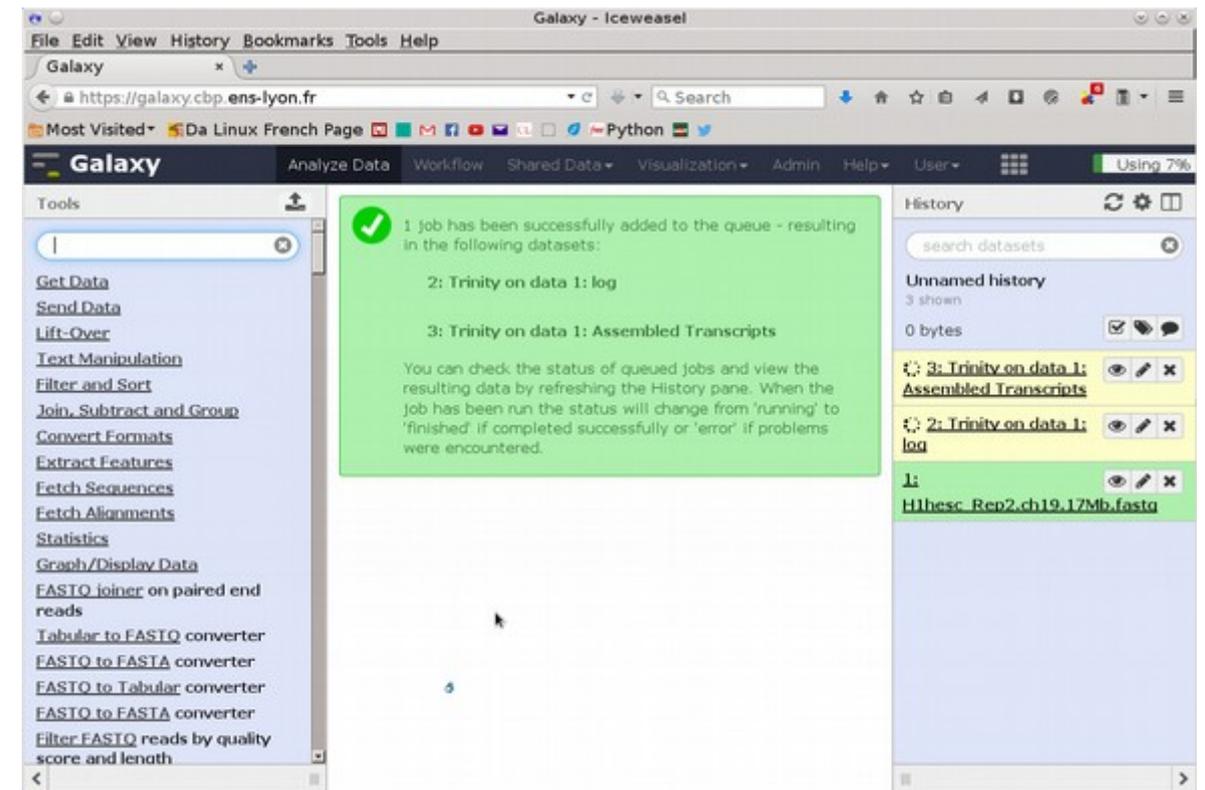
Retour sur la métrologie (sommaire) de
quelques années au PSMN & au CBP

*« Il existe trois types de mensonges : les petits mensonges,
les gros mensonges, et les statistiques ! » Mark Twain*

Emmanuel Quemener

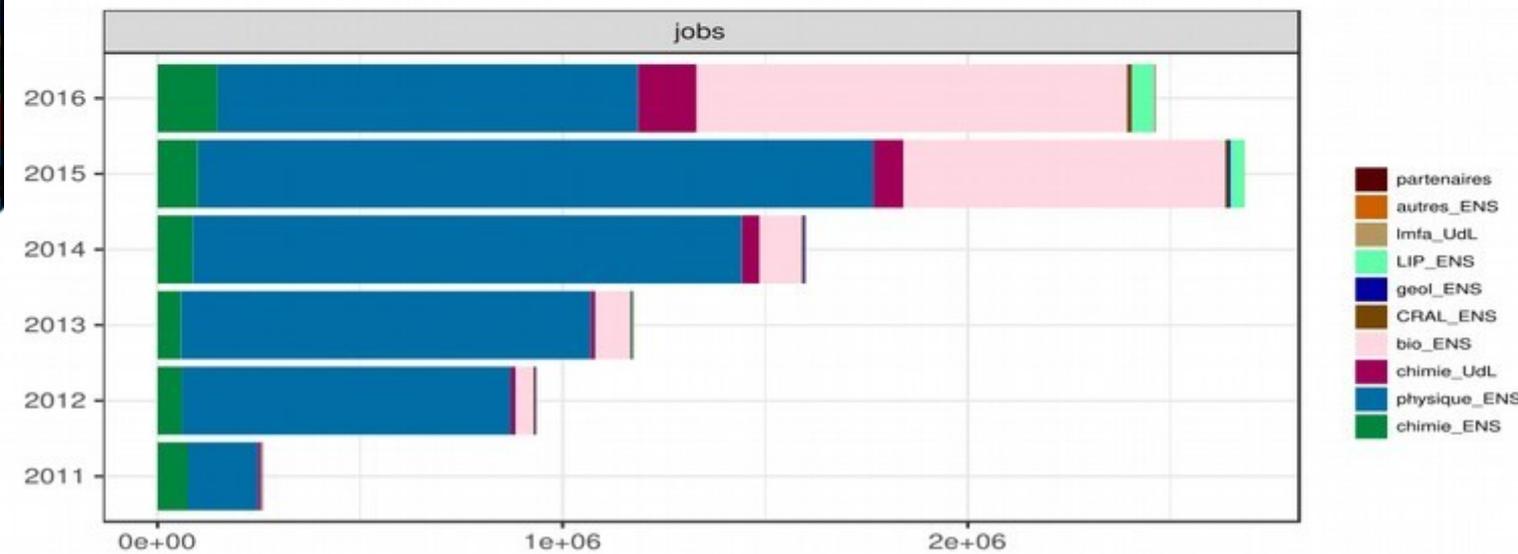
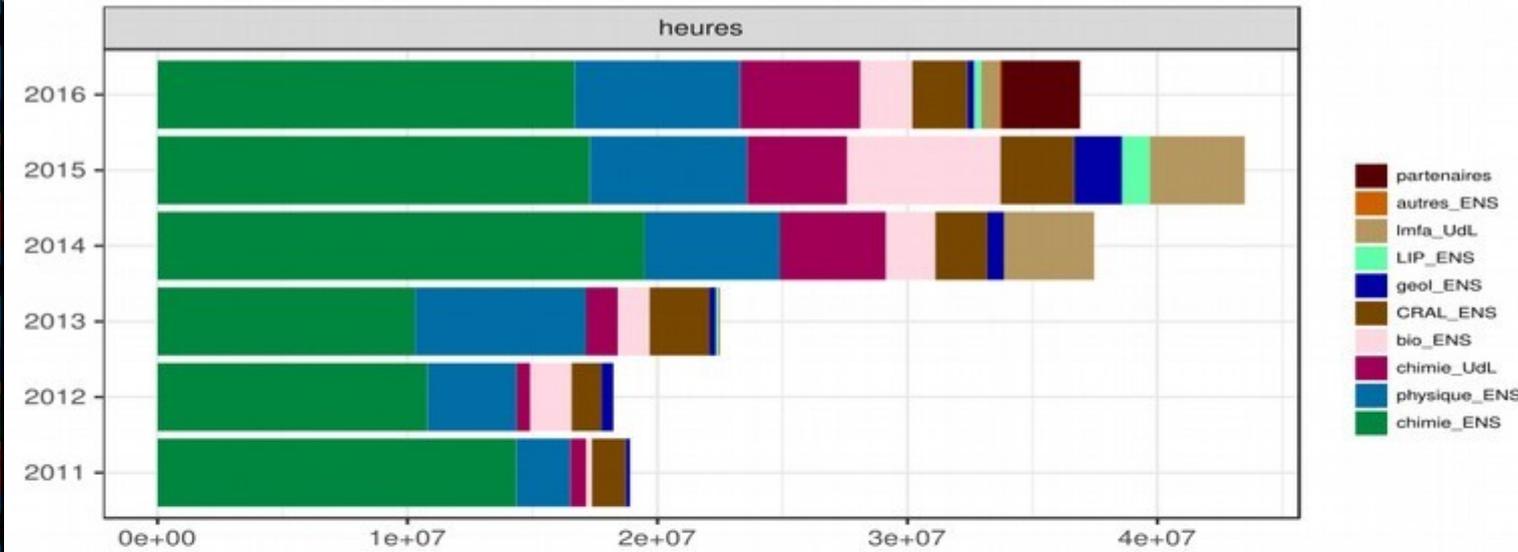
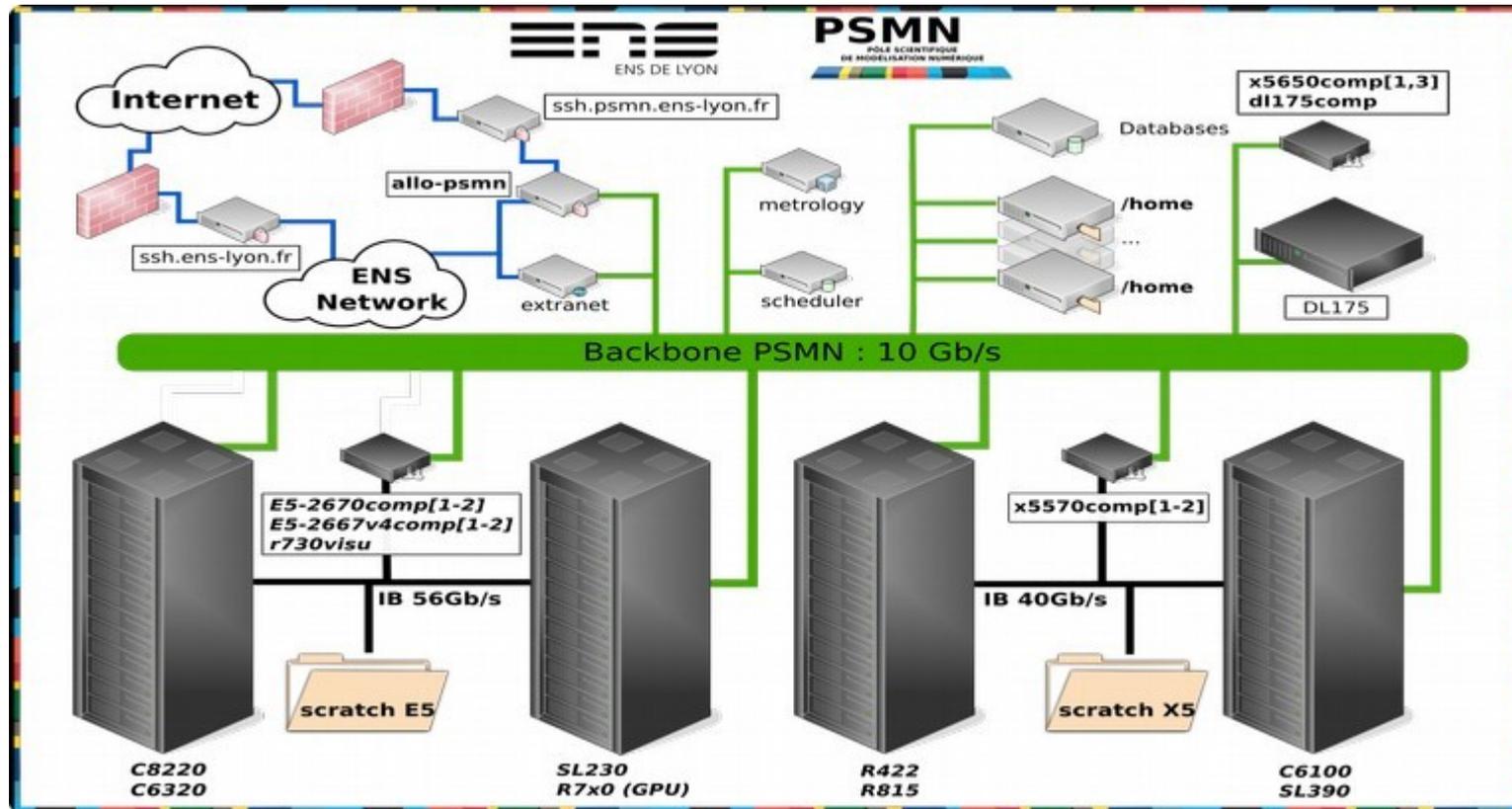
Des « paillasse numériques » permanentes ou éphémères !

- Ecole des Houches : 6 éditions 2011-2016
- Humanités Numériques : 20 machines virtuelles
- Géographie : stations graphiques déportées
- Biologie :
 - Postes de traitement & visualisation
 - Serveur pour étude et + Repeat*
 - **Portail Galaxy**
 - (avec exploitation de cluster)



Mésocentre PSMN

Son utilisation en jobs & durées



- Evolutions :
 - dans le temps
 - dans les usages

Deux contextes, deux périodes

Un format mais 45 attributs...

- Contextes & périodes : ENS-Lyon
 - **PSMN** (méso-centre) : du 11/02/2010 au 31/07/2017, **2726** jours
 - **CBP** (hôtel à projets & centre d'essais) : du 30/01/2015 au 28/07/2017, **909** jours
- Un format : SGE et Grid Engine
- 45 attributs : dont 18 issus de getusage
 - qname, hostname, group, **owner**, **job_name**, job_number, account, priority, submission_time, start_time, end_time, **failed**, **exit_status**
 - **wallclock**, **utime**, **stime**, **maxrss**, **ixrss**, **ismrss**, **idrss**, **isrss**, minflt, majflt, nswap, **inblock**, **oublock**, **msgsnd**, **msgrev**, nsignals, **nvcs**, **nivcs**
 - project, department, granted_pe, **slots**, task_number, **cpu**, **mem**, **io**, category, iow, pe_taskid, **maxvmem**, arid, ar_sub_time

Comment ? Des journaux à la fouille

- Un format de logs simple : « : » *separated values*
- Tailles : commande `wc`
 - PSMN : 10839519 (lines) 38318175 (char) 4275622266 (bytes)
 - CBP : 130907 (lines) 355754 (chars) 40332932 (bytes)
- Comment fouiller ?
 - Par un shell avec des appels Rscript : un peu « overkill »
 - Par un python avec Pandas : très lourd à l'usage
 - Par une SGBD relationnelle : efficace
 - Par une approche hybride : SQLite3 + Python Pandas (+ Libreoffice)

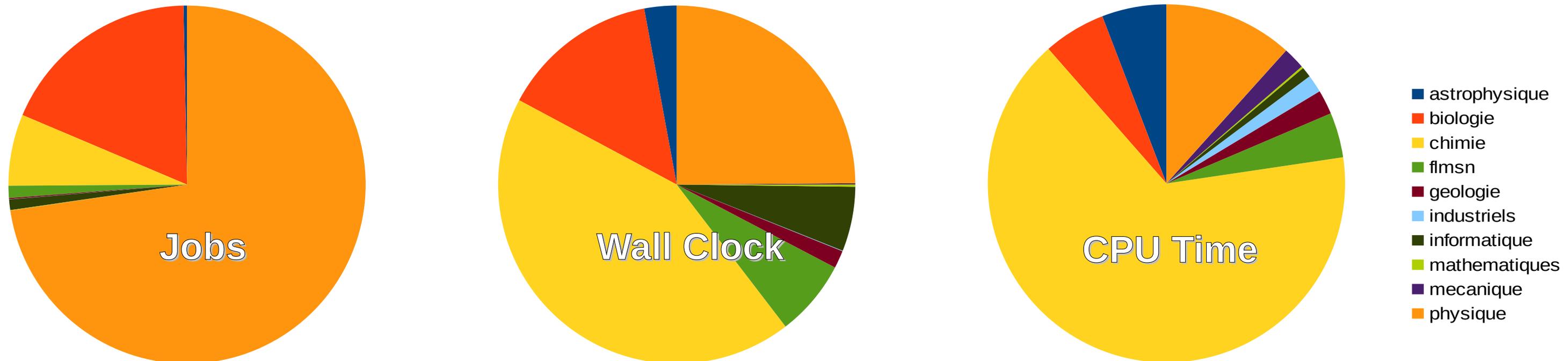
Comment ? Les étapes...

- Création d'une base sqlite3
- Importation dans BDD sqlite3
 - Des logs Grid Engine du CBP
 - Des logs Sun Grid Engine du PSMN
 - Des correspondances identifiant/laboratoire, laboratoire/activité
- Nettoyage des journaux
- Analyse selon quelques critères objectifs
- Synthèse pour isoler des usages par communauté

Une première étape : le nettoyage... Salutaire !

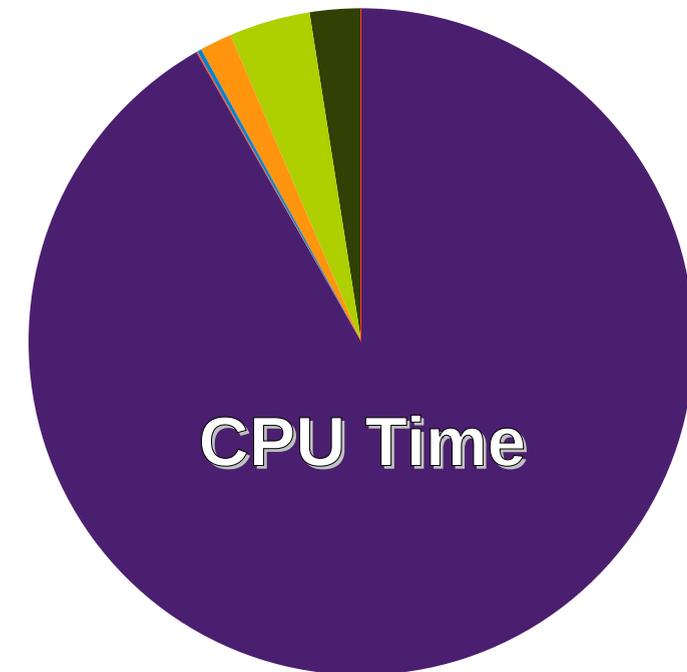
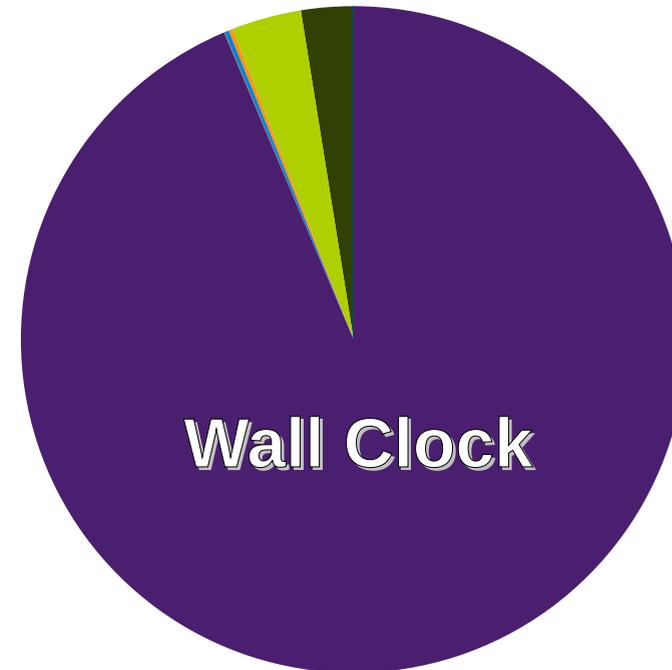
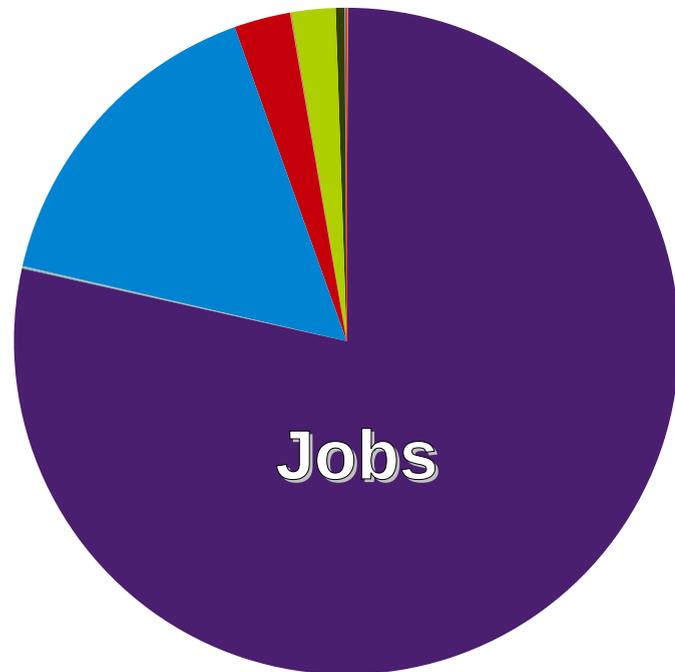
- Avant : CBP#130907 et PSMN#10839510
 - Suppression :
 - Epoch=0 : des dates sont incohérentes
 - Failed!=0 ou Exit_status!=0 : des jobs vautrés
 - Wallclock=0 ou CPU<1 : la durée totale est nulle
- Après : CBP#101929 et PSMN#9333980
 - -19 % pour CBP, -10 % pour PSMN

Classement PSMN par discipline : Jobs, WallClock, CPU Time



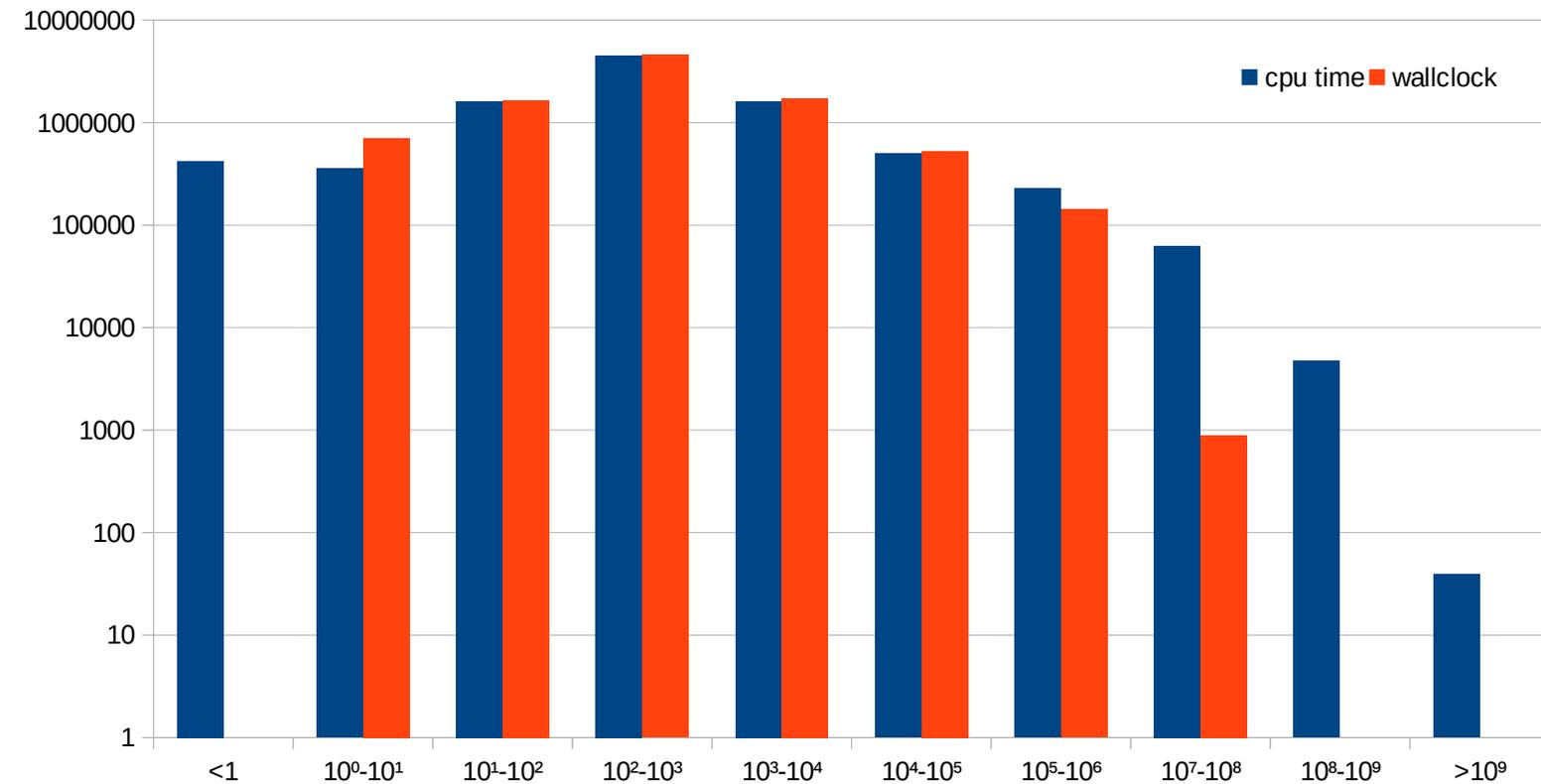
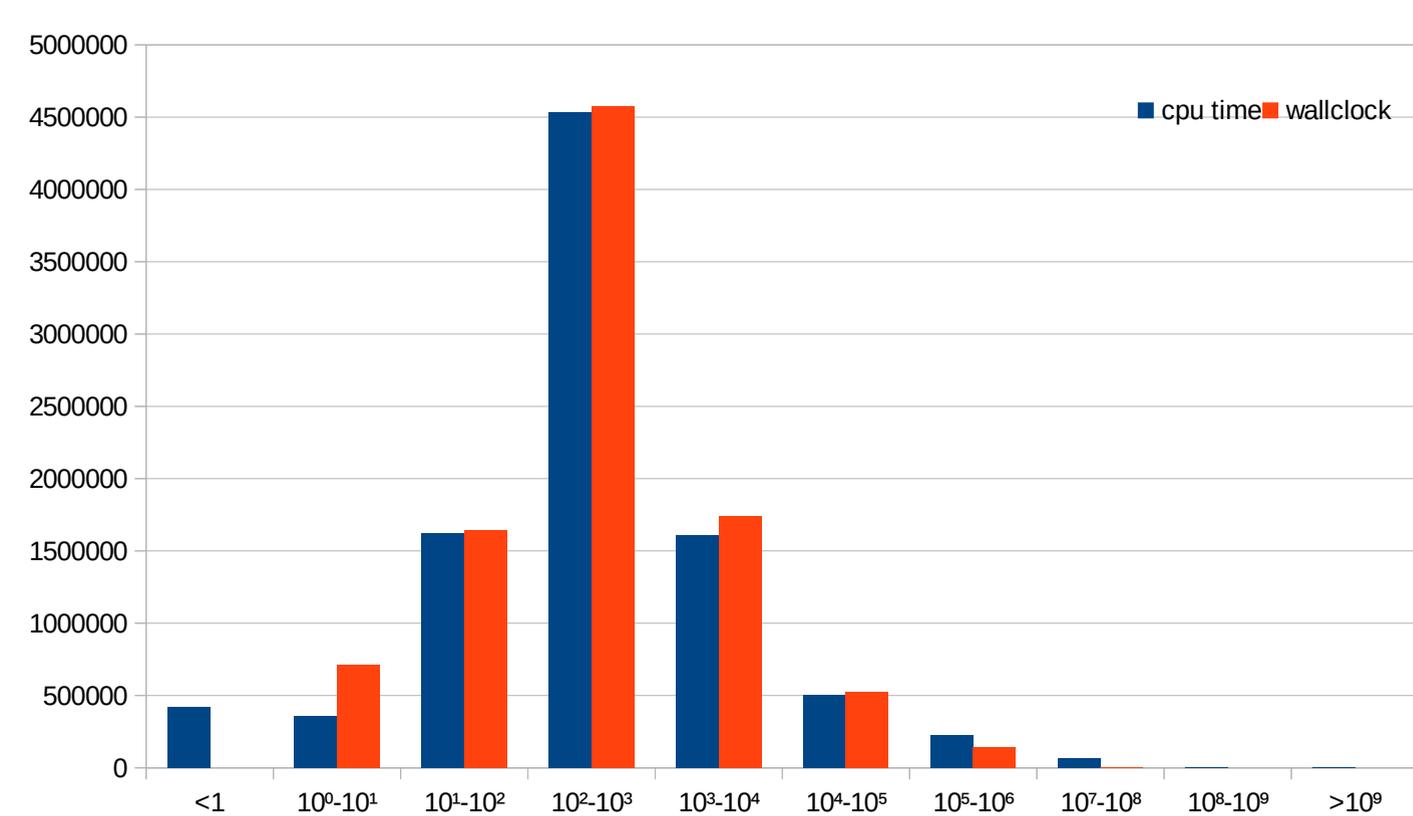
- Des diversités d'utilisation visuellement différentes :
 - La physique : beaucoup de jobs, visiblement courts
 - La chimie : beaucoup de temps CPU et temps écoulés
 - La biologie : beaucoup de jobs, pas mal de temps, moins de CPU

Classement CBP par utilisateur : Jobs, WallClock, CPU Time



- Des diversités d'utilisation encore plus marquées
 - Un « goinfre » de calcul séquentiel
 - Un « galaxy » significatif en nombre de jobs, imperceptible ailleurs
 - Un usage CPU plus massif que Wallclock pour un utilisateur

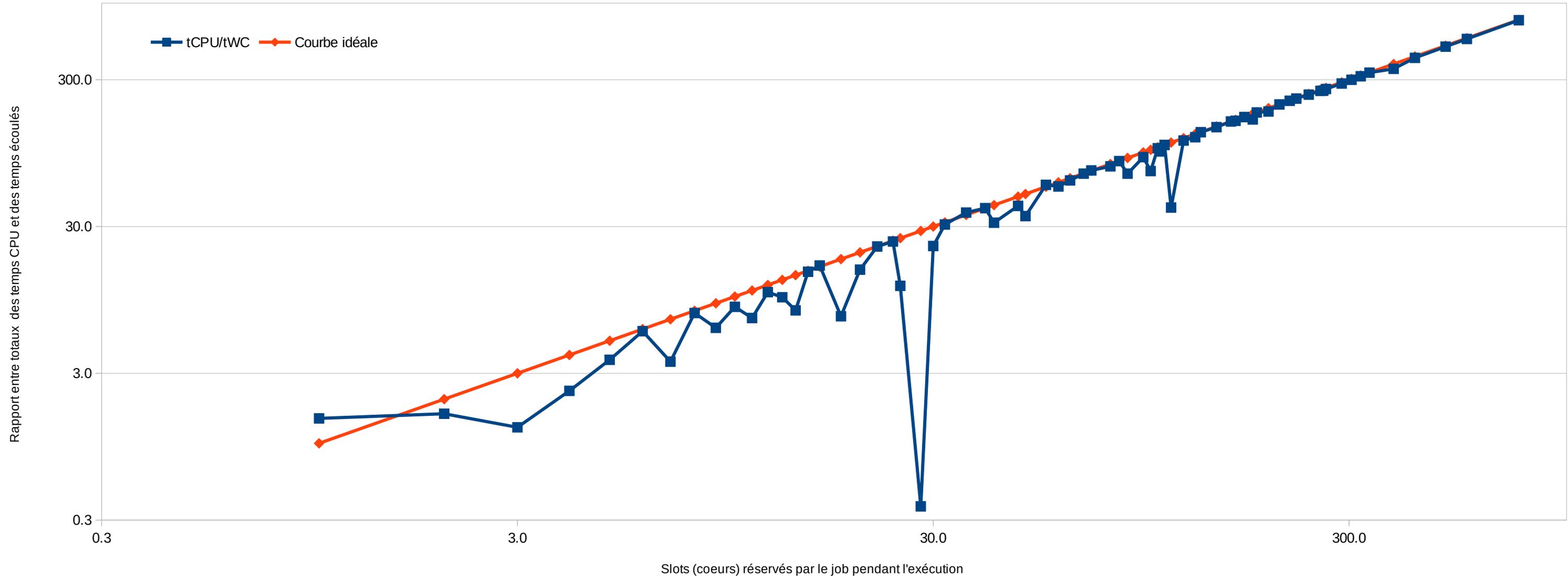
Distribution des jobs en durées au PSMN : WallClock*Slots & CPUtime



- Une immense majorité de jobs entre 100 & 1000 s
- Quelques jobs au dessus de l'année
- Quelques centaines de milliers sous la seconde...

Analyse globale des slots réservés

Sont-ils bien occupés ? Oui, mais...



Je consomme « moins » que ce que je réserve, sauf slots=1 !

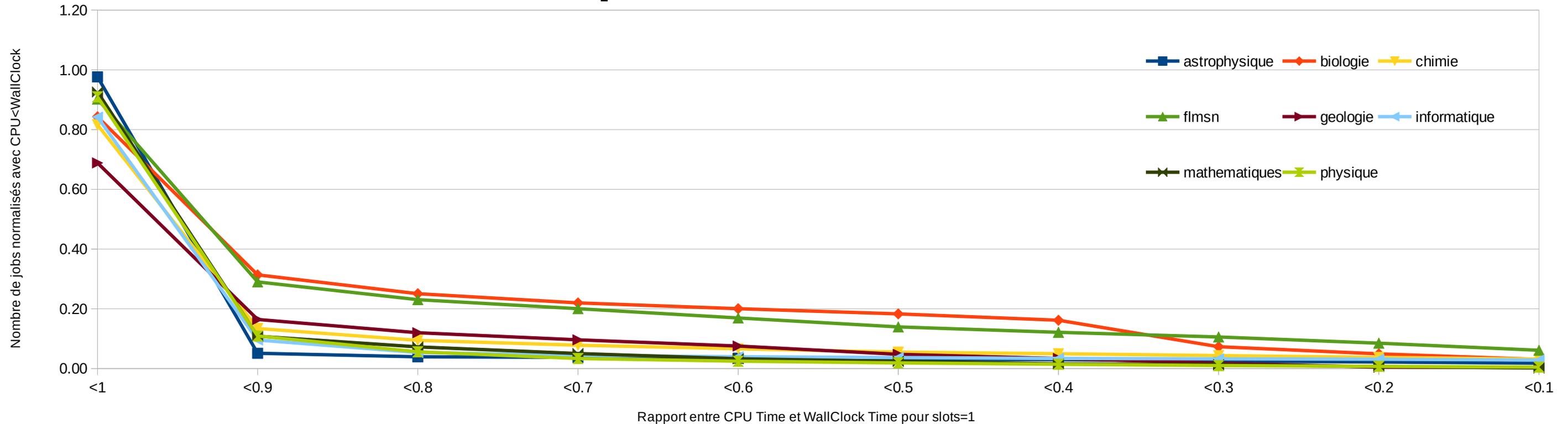
C'est dans les limites que la vérité se dévoile...

Examinons 2 cas :

- Slots=18, $\text{Time}_{\text{cpu}} = 7 * \text{Time}_{\text{WallClock}}$ au lieu de 18...
 - Quelques jobs avec une machine « surchargée » : 8 cœurs, 18 slots
- Slots=1, $\text{Time}_{\text{cpu}} = 1.46 * \text{Time}_{\text{WallClock}}$ au lieu de 1...
 - $\text{Time}_{\text{cpu}} > \text{Time}_{\text{WallClock}}$ pour 905185 jobs, soit ~10 %
 - $\text{Time}_{\text{cpu}} < \text{Time}_{\text{WallClock}}$ pour 7672391 jobs, soit ~90 %
 - $\text{Time}_{\text{cpu}} > 10 \text{ Time}_{\text{WallClock}}$ pour 1591 jobs, soit ~0.02 %
 - **$\text{Time}_{\text{cpu}} < 10 \text{ Time}_{\text{WallClock}}$ pour 100142 jobs, soit ~1 %**
 - Des « comportements » individuels de jobs très différents...
- Comment extraire des « lois » d'une telle diversité ?

Quelle origine de cette « perte » ?

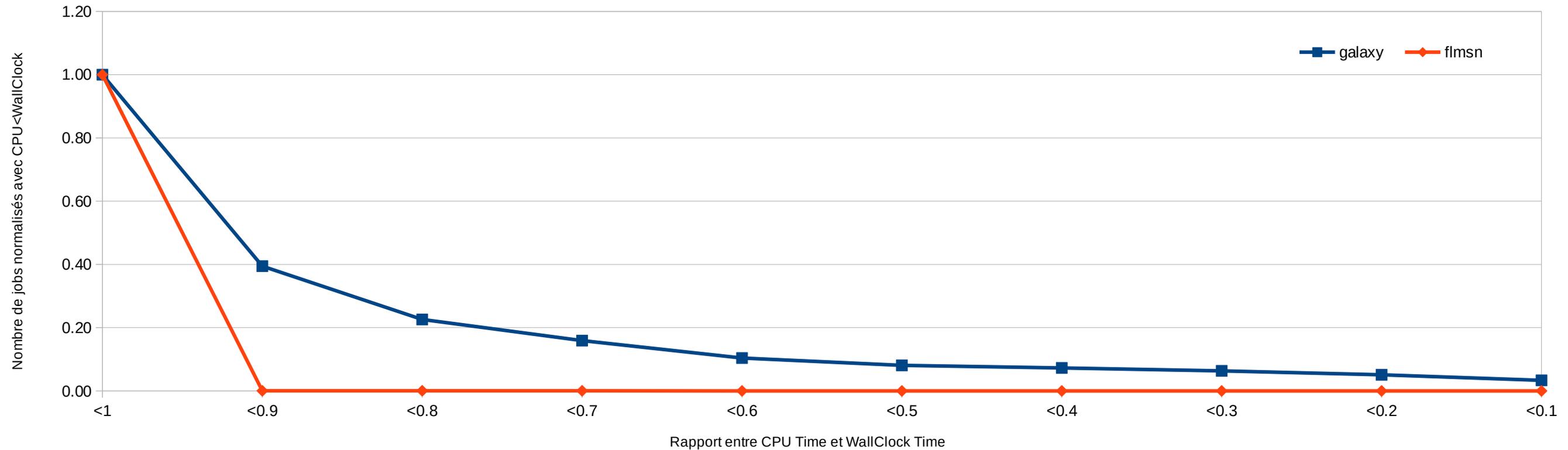
Examen pour CPU/Wallclock



- CPU~WallClock pour tous sauf biologie & flmsn
- Comportement à l'exécution manifestement différent
 - Fouille nécessaire dans les autres informations disponibles...
 - Utilisation d'une statistique plus élaborée...

Et le CBP, quel comportement ?

Examen pour CPU/Wallclock

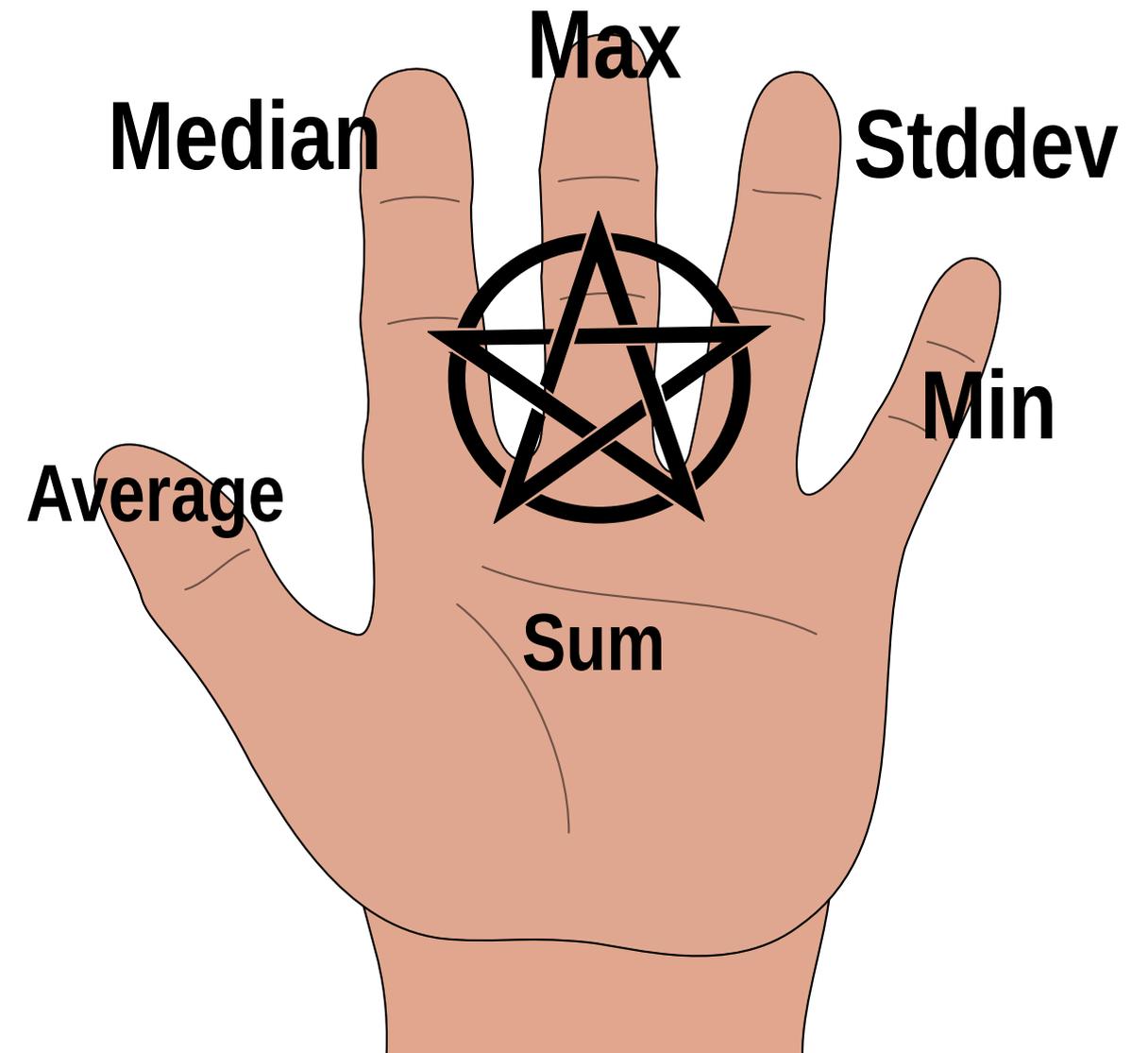


- Le critère WallClock/CPU est manifestement intéressant
- Comportement à analyser par activité...

Question de mesures...

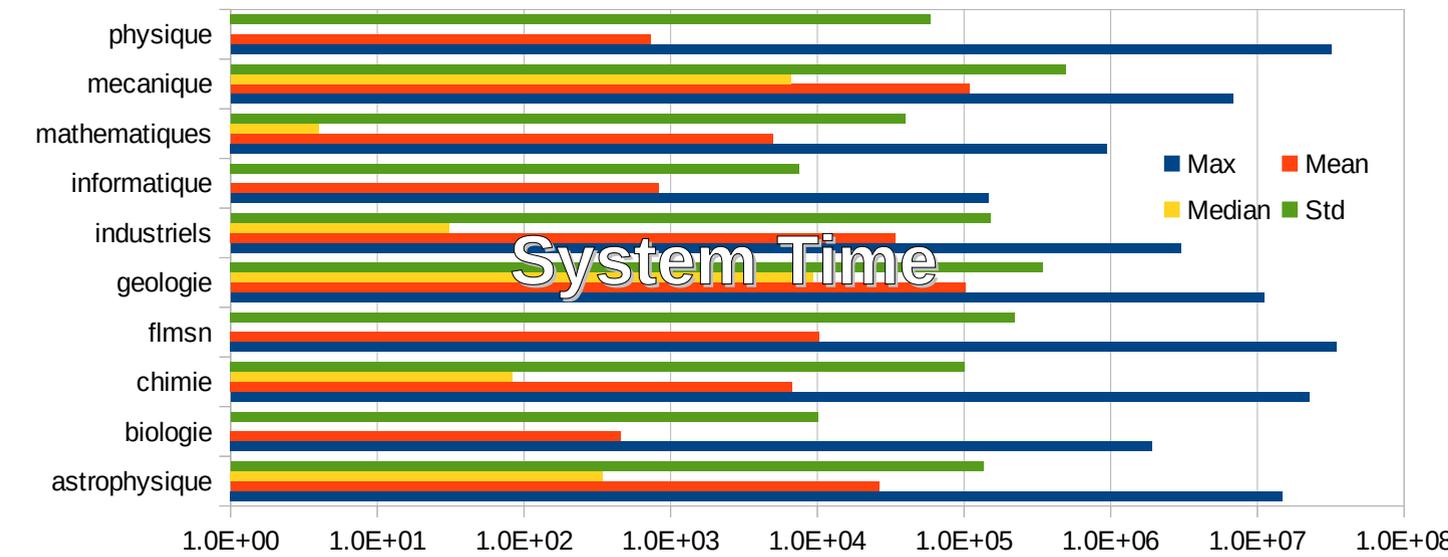
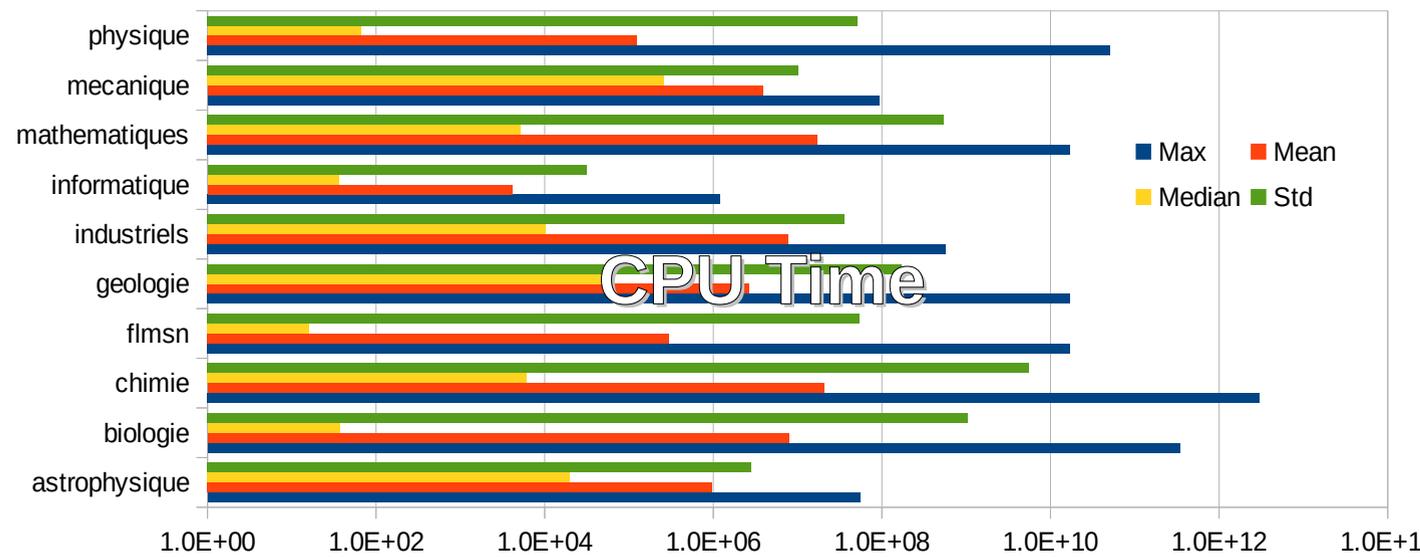
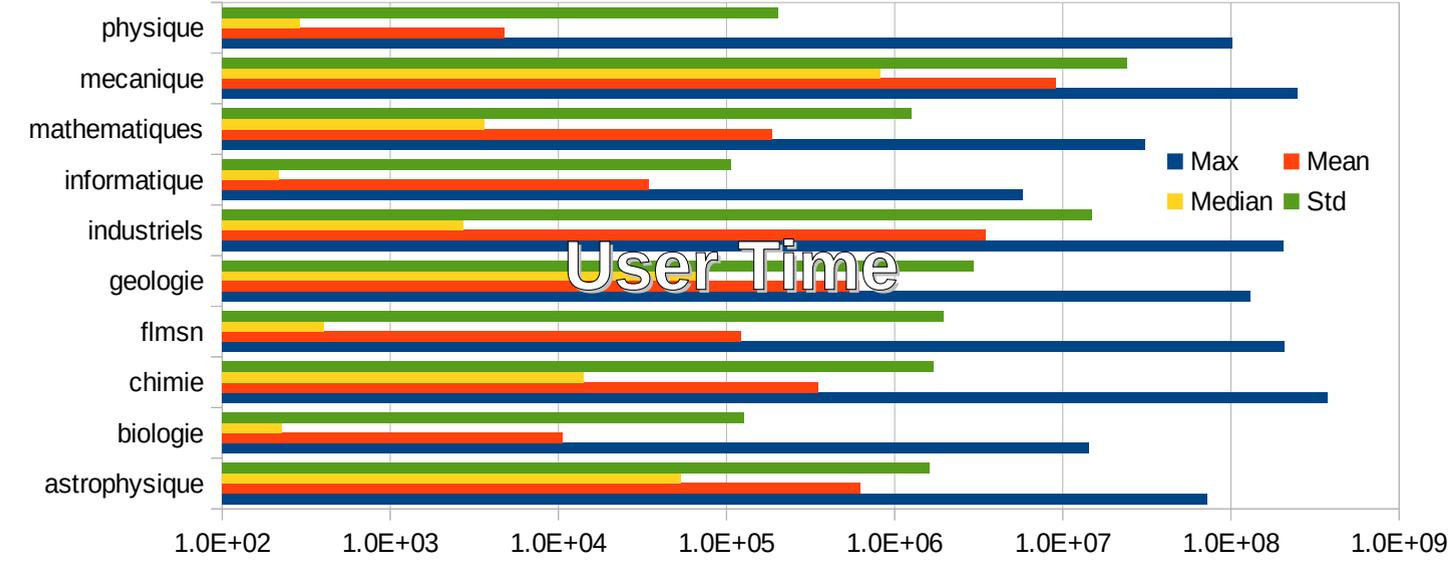
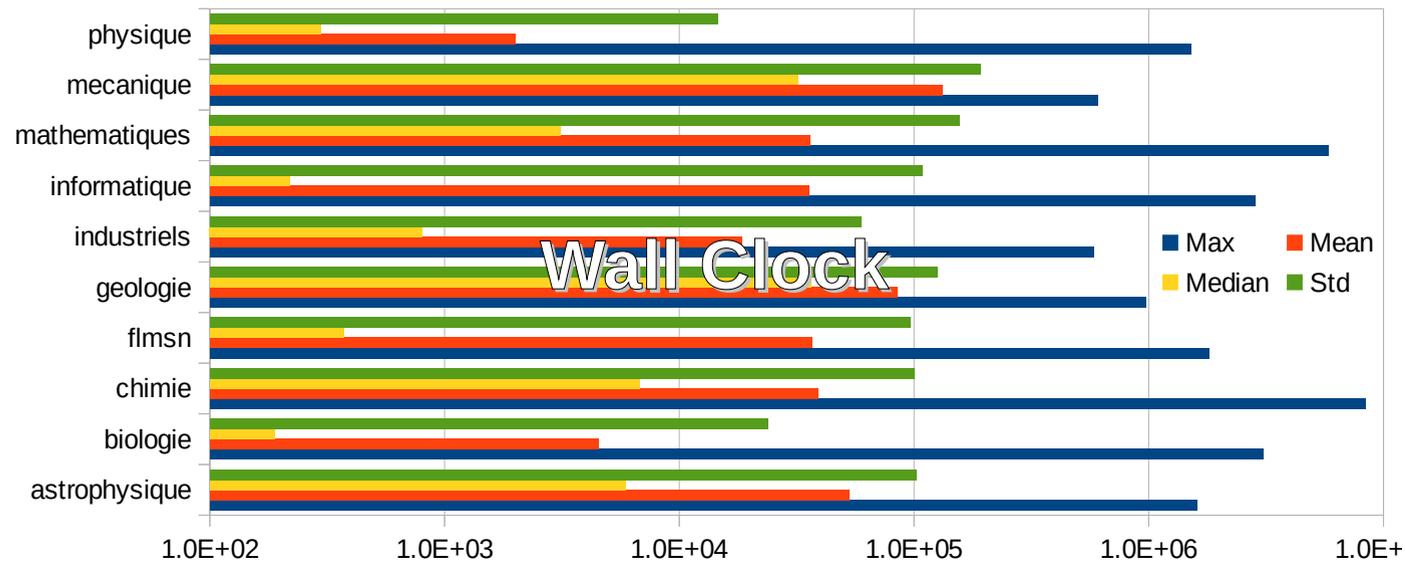
Le pen(s)tacle de la statistique

- Pourquoi accumuler des statistiques ?
 - Parce qu'on pratique des sciences
- Le pentacle de la statistique :
 - Moyenne (avg) : le premier auquel on pense
 - Mais très sensible à l'initialisation & cas atypiques
 - Médiane : moins connu, mais plus pertinent
 - Maximum (max) : le pire ou le meilleur
 - Minimum (min) : le meilleur ou le pire
 - Ecart type (std) : indicateur de variabilité
- Variabilité : ratio Stddev/Median



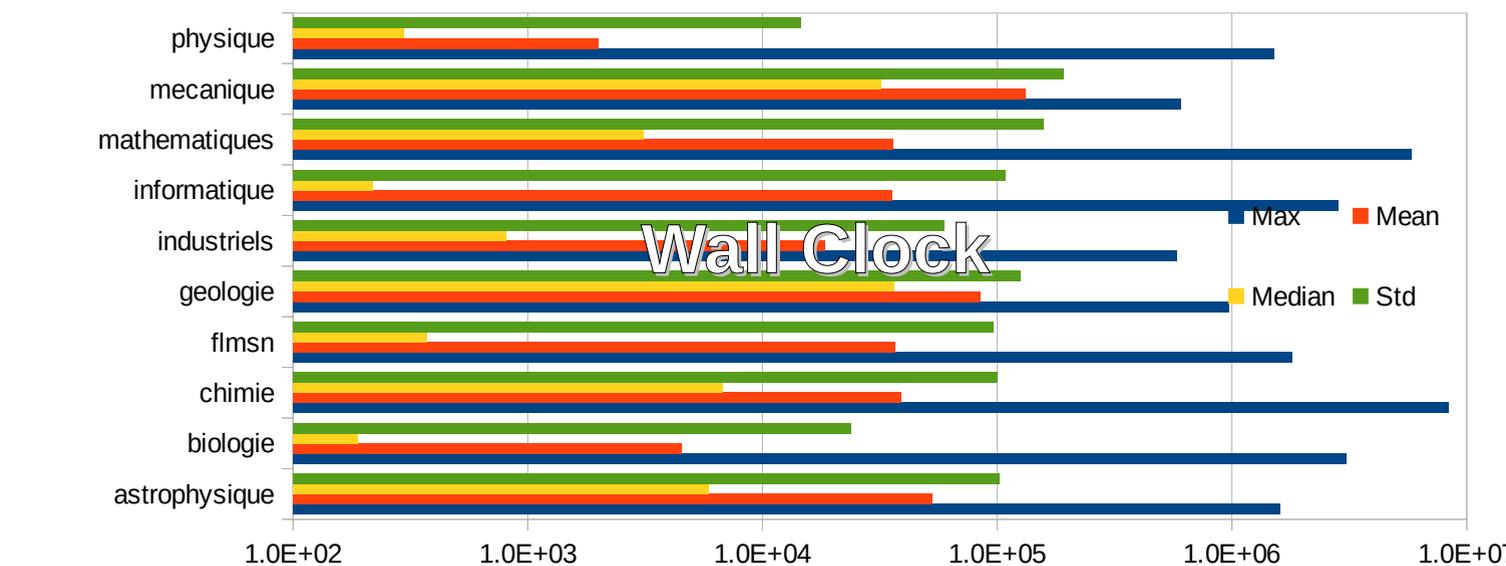
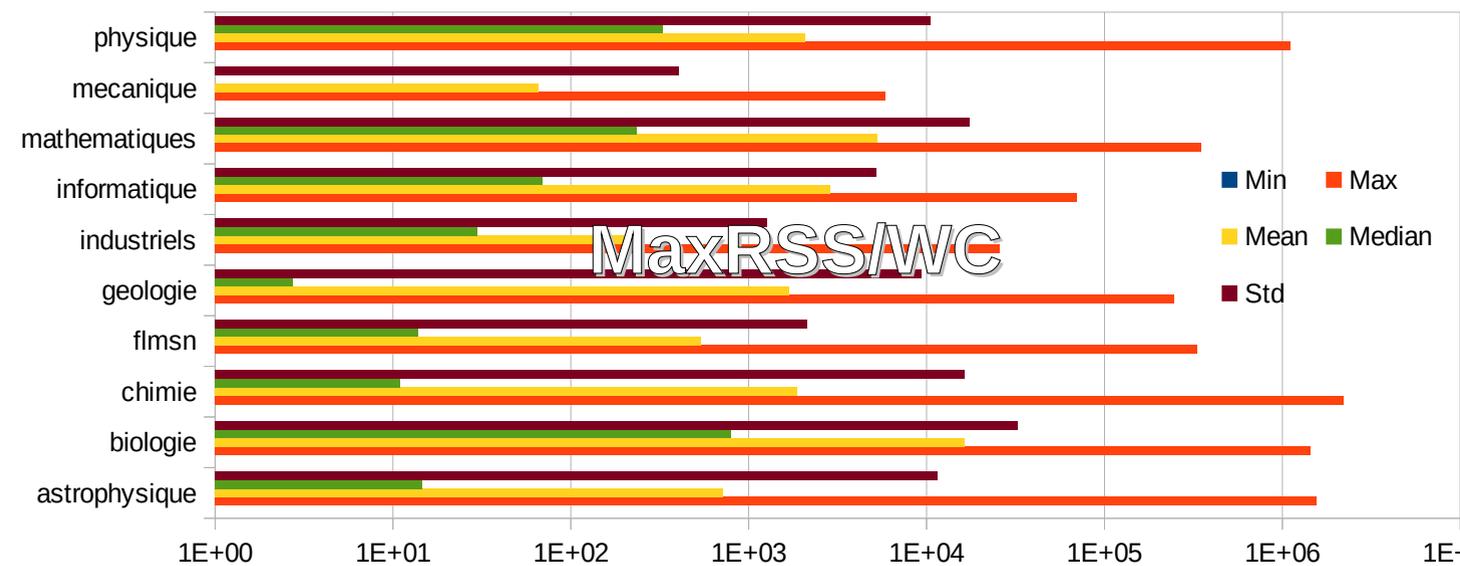
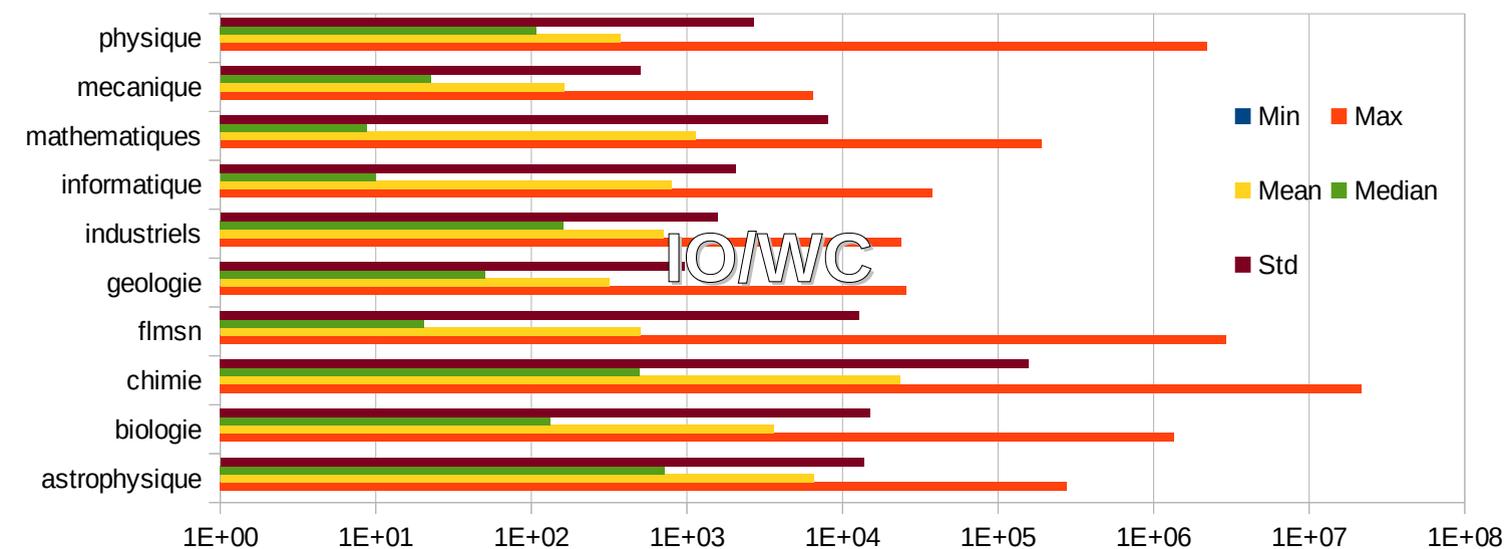
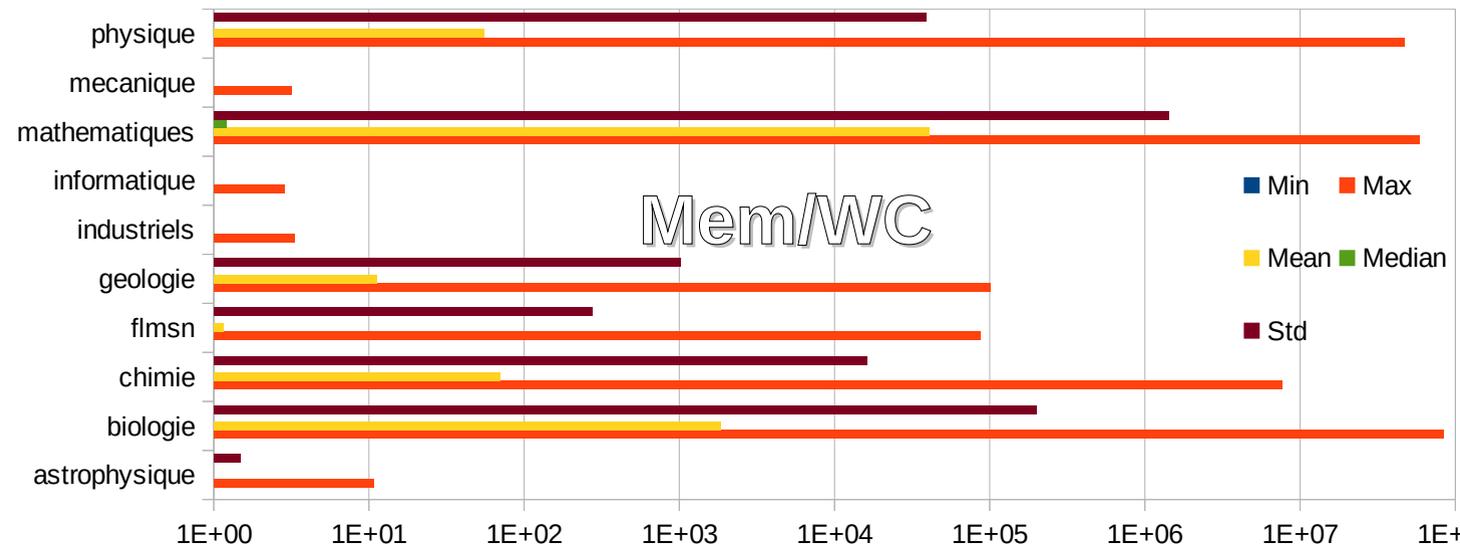
Statistiques sur les durées des jobs

WallClock, User Time, System Time, CPU Time



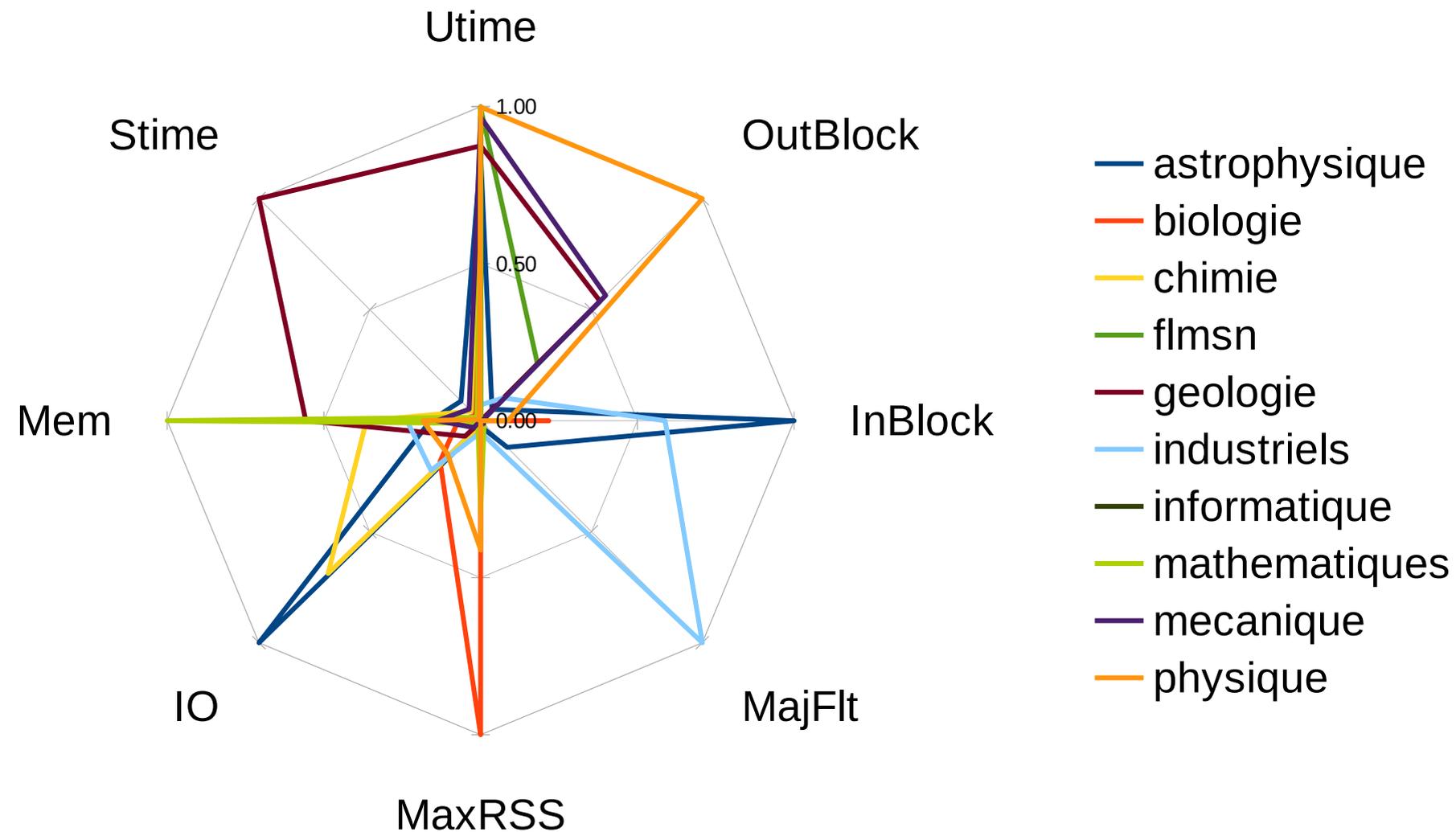
Seul point de concordance : Ecart Type > Moyenne !

Et les autres métriques... Intéressantes ? Comparables (mais inexploitable!) :-/



Toujours de grosses dynamiques, mais MaxRSS et IO

Le salut vient de la médiane...



Un comportement discriminant plus « visuel » :
moins de couleurs superposées !

Du « stress » au « burn-out » des systèmes, c'est quoi ?

- Les indicateurs du « stress » système
 - La mémoire utilisée : mem, maxrss
 - Les entrées/sorties : io, inblock, outblock
 - Le « temps système » : stime
 - Les changements de contextes : nvcsw, nivcsw
- Le « stress » : le système est utilisé au-delà sa capacité
 - Un ratio avec le Wall Clock très inférieur à 1
 - Le maxrss relatif au Wall Clock
- Le « burn-out » : le système semble inutilisé mais jobs en cours
 - Un ratio avec le Wall Clock très supérieur à 1

Quelles stratégies de facturation ?

Le bonus/malus ?

- Bonus : récompenser une bonne utilisation du système
 - Cohérence entre la réservation et l'utilisation : WallClock*slots ~ CPU
- Malus : taxer une mauvaise utilisation du système
 - Ventilation des opérations « lourdes » en I/O : le « tant que je gagne, je joue ! »
 - En fait, quand ça rame, « Je ne suis pas dans le trafic, je SUIS le trafic ! »
 - Utilisation excessive de l'OS : les codes hybrides mal lancés 16cHT : 256 processus
 - Symétrie I/O : applications de biologie
 - Redoutable pour les systèmes de fichiers distribués, le cache ne fait pas tout !
- Dans tous les cas, mais mieux instrumenter (systèmes, codes, ...)
 - Ad mortem avec /usr/bin/time, ou au fil de l'exécution avec des équivalents à dstat...

Appel aux dons : donnez vos vieilles machines !

- Plus c'est vieux, mieux c'est !
- Même si c'est cassé ça peut servir !
- Et les périphériques (et leur câblerie), c'est important...
- Exploitation pour les ateliers 3IP
 - Introduction Inductive à l'Informatique et au Parallélisme
- Conservation dans la comphéthèque du CBP
 - Maintien en condition opérationnelle des machines
- Exemple d'un K6 démarrant sous SIDUS sur Youtube