Analyse de performances Linux avec perf

Hadrien Grasland

2022-11-15

Périmètre

- Pour optimiser, il faut savoir **à quoi on passe son temps**
  - Accélération 10x sur 0,1 % du temps = gain global 0,09 %…
- Bons outils spécifiques OS ou matériel, souvent les deux
  - Sujet principal ici : analyse de l'**activité CPU** sous **Linux**
- La **compilation avant exécution** permet des outils génériques
  - Modèle standard pour C/++, Fortran, Go, Rust…
  - Quelques informations accessibles de façon universelle

Pourquoi perf ?

	(i)gprof, gperftools…	callgrind	VTune	perf
Support multi-thread	⚠	✘	✔	✔
Activité micro-architecture	✘	⚠	✔	✔
Granularité instruction ASM	✘	✔	✔	✔
Activité libc (mutex, malloc…)	⚠	⚠	⚠	✔
Instrumentation de code	✘	✘	✘	✔
Analyse activité OS	✘	✘	✘	✔
Utilisable via SSH	⚠	⚠	⚠	✔
Biais de mesure faible	✔	✘	✔	✔
Support matériel AMD, ARM…	✔	✔	✘	✔

Limites de perf

- Mieux vaut avoir un **noyau plus récent que le CPU**
  - Attention aux distributions dont les paquets ont CentANS
- Soucis possibles avec la virtualisation et les conteneurs*
- Certaines observations nécessitent des **privilèges utilisateur**
- Visualisation & documentation peu développés → Ce TP !

* Pour la virtualisation, il faut configurer l'hyperviseur pour donner accès aux PMUs.
Pour les conteneurs, il faut que la version de perf installée soit compatible avec le noyau hôte.
Dans les deux cas, l'observabilité limité de l'activité du matériel/système hôte est un biais dangereux.

Participer au TP

- Plate-forme : srv-calcul-ambulant
  - CPU mutualisé : **Utilisez srun** pour les activités coûteuses !
  - Accessible via réseau Wi-Fi local « srv-calcul-ambulant »
    - Infos de connexion affichées après la présentation
  - URL locale TPs : http://srv-calcul-ambulant/docs/user/getting-started/practical-work.html
- Format semi-encadré
  - Majorité du TP faisable en autonomie

Le reste de la famille

- Autres TPs disponibles sur le serveur :
  - CADNA, PROMISE (arithmétique stochastique)
  - E.V.E (vectorisation explicite C++20)
  - Ecriture de plugins LLVM
  - La collection de Pierre Aubert
    - Outils (GDB, Gitlab, MAQAO, Valgrind)
    - Optimisation de performances
    - Impact des flottants spéciaux (NaN, dénormaux…)

Attention, service dégradé en 2024

- Normalement, le serveur a 4 Go/hyperthread (128 Go)
- J'ai dû enlever 6/8 RAM défaillantes avant les JIs
  - Pas le temps de les remplacer
- Donc on tourne à 1 Go/hyperthread (32 Go)
  - Attention à l'empreinte RAM de vos jobs
  - TP perf peu affecté **sauf** expé working set @ perf stat

Merci de votre attention !

Version internet : https://grasland.pages.in2p3.fr/tp-perf/