

# Projet en architecture des processeurs hautes performances et programmation efficace

## Master d'informatique, UNS

Pr Sid TOUATI

### Instructions

Le projet à rendre doit être le fruit d'un travail personnel. Vous êtes autorisés à communiquer entre vous, à vous échanger des idées et des astuces. En revanche il est interdit de soumettre un projet qui n'est pas le fruit d'un travail personnel.

## 1 Sujet du projet

Le projet consiste à faire les exercices suivants extraits du TP nommé « La hiérarchie mémoire » :

1. Exercice 1 : Tester les performances de la hiérarchie mémoire d'un CPU avec un micro-benchmark séquentiel.
2. Exercice 2 : Évaluation de la bande passante entre le processeur et la mémoire centrale.
3. Exercice 5 : L'outil Calibrator.

Vous devrez effectuer les exercices 1 et 2 et 5 sur votre machine Linux native (pas sur une machine virtuelle) de façon rigoureuse, et rendre les résultats dans un fichier archive décrit dans la Section 3. Vous devrez rédiger un rapport en latex (obligatoirement), et fournir les codes sources de vos programmes et les données expérimentales récoltées.

## 2 Méthodologie expérimentale

Préparez vos codes de micro-benchmarks en vous assurant qu'ils mesurent bien ce que vous souhaitez mesurer. Les expériences doivent s'effectuer sur une machine Linux à faible charge (*i.e.* très peu de processus en activité pour ne pas perturber les mesures). L'idéal serait que vous puissiez booter Linux avec un profile de boot qui allège la machine au maximum (pas d'interface graphique, désactivation des services de l'OS qui ne sont pas utiles pour le projet). Assurez vous de désactiver le service de modulation de fréquence d'horloge (appelé parfois DVS, qui veut dire Dynamic Voltage Scaling) pendant vos expériences : soit vous désactivez cette option via le bios, soit vous désactivez le service Linux DVS. Désactivez également le mécanisme de préchargement de données (*data preloading*) si cela est possible. Vos expériences devraient être effectuées sur une console de shell sur une machine à très faible charge de travail.

Une fois les expériences terminées et les données à récoltées dans des fichiers, vous pouvez rebooter à nouveau votre station de travail en mode graphique.

### 3 Documents à rendre strictement via jalon.unice.fr

Vous devrez rendre votre projet via jalon (via une boîte de dépôts), ne m'envoyez pas votre projet par mel. Votre archive devra s'appeler **Projet-Nom-Prenom.tar.gz**. Elle doit contenir :

1. Le source latex de votre rapport, et le pdf généré.
2. Pour chaque exercice 1 et 2 et 5 du TP (créez un répertoire distinct pour chaque exercice de TP)
  - (a) Les codes C de vos micro-benchmarks.
  - (b) Les données expérimentales générées par vos micro-benchmarks
  - (c) Les figures (en pdf) générées avec gnuplot + script gnuplot. Chaque figure doit avoir une légende, et un titre pour l'axe des X et l'axe des Y.

La taille du rapport n'a pas d'importance (rapport court ou détaillé, c'est la qualité qui compte, et votre honnêteté et crédibilité scientifique). Le rapport ne doit pas contenir des rappels de cours, ni les codes C ni les données récoltées (les codes C et les données seront fournis dans l'archive). Le rapport doit contenir ce qui suit :

1. Une page de titre + votre nom et prénom + date.
2. Une section de description de l'environnement expérimental, à savoir
  - (a) une sous section de description de l'environnement matériel : décrivez votre machine Linux (caractéristiques de l'architecture et micro-architecture de votre processeur). Vous devrez détailler les caractéristiques de la hiérarchie mémoire (différents niveaux de caches, tailles des caches, tailles des lignes cache, associativité de chaque cache, etc), du pipeline du processeur, des informations sur le parallélisme d'instructions de votre processeur, etc. Appuyez vous sur le TP nommé « Comment analyser une architecture matérielle avec Linux » pour avoir des informations précises sur votre machine.
  - (b) une sous section de description de l'environnement logiciel : version de Linux, distribution Linux, quel est le compilateur utilisé (gcc ou autre), options de compilation utilisées (flags du compilateurs), quel outil de mesure des temps d'exécution.
3. Une section pour décrire la configuration de la machine pendant les expériences (*i.e.* les processus et les services OS en activité pendant les expériences), la méthodologie que vous avez suivie pour récolter les données expérimentales (si vous avez eu des difficultés expérimentales, décrivez les).
4. Une section par exercice de TP pour décrire les résultats (exercice 1 et 2 et 5) : figure + analyse des résultats.
5. Sur certains détails que vous estimez croustillants, décrivez ce que vous avez compris par vous même. Inutile de me réciter le cours, ce qui m'importe est de comprendre ce que vous avez compris par vous même, par vos efforts.
6. Soignez la conclusion. Ne la rédigez pas à la hâte, sachez que la conclusion sera probablement la seule section qui sera lue attentivement par vos correspondants et vos supérieurs. La conclusion doit comporter le message que vous voulez faire passer, ce n'est pas une section qui décrit ce que vous avez fait mais une section qui décrit ce que vous concluez après votre travail.

### 4 Comment serez vous notés ?

Ce qui m'intéresse le plus dans votre travail est la démarche expérimentale, votre analyse des résultats, votre capacité à rédiger un document clair et bien structuré,

votre capacité à suivre des instructions précises et à rendre un travail selon un cahier de charges. Il faut que votre rapport d'étude expérimentale soit honnête, dans le sens où il doit refléter fidèlement vos résultats expérimentaux et vos observations.

## **5 Qu'allez vous apprendre grâce à ce projet ?**

- Effectuer des expériences sur l'évaluation des performances des processeurs et des programmes.
- Analyser une architecture matérielle sous Linux avec des micro-benchmarks.
- Apprendre à utiliser des outils d'édition de documents scientifiques comme Latex et gnuplot.