

Large-Scale Distributed Computing System

Johnny Nguyen¹

Abstract—This electronic document permits me to synthesis the third course of data mining.

I. INTRODUCTION

La plupart des utilisateurs sont des scientifiques. Les différents domaines sont :

- Astronomie,
- Nucléaire,
- Observation terrestre,
- Sciences biomédical,
- Physique,
- Chimie,
- Sciences humaines,
- Systèmes complexes.

Il y a aussi des usages dans la société.

- Diffusion,
- Prévention de catastrophe,
- Energies renouvelable.

Le but est de développer une infrastructure nécessaire à nos besoins. La fréquence du CPU est directement relié à la puissance de celui-ci. Un CPU est composé d'une mémoire cache permettant de décoder des instructions avec des registres. On peut améliorer ce processeur en effectuant du pipelining, du cache to speed up RAM et la prédiction de branche. Actuellement la tendance est de multiplier le nombre de processeur pour augmenter la puissance. Les transistors n'ont plus d'influence sur la performance d'un processeur alors que la façon de coder a une grande importance. Nous savons que SIMD est très efficace pour les GPU. La mémoire partagée n'est pas la meilleure solution tout le temps et tous les 21 mois le débit internet est censé doubler.

Dupliquer la puissance des ordinateurs permettent d'obtenir de meilleure performance.

Nous ordonnons les technologies de la plus ancienne à la plus récente :

- Optimisation d'architecture,
- Hyperthreading,
- Coeurs,
- Super ordinateurs,
- Cluster computing,
- Grid cloud.

II. PROGRAMMATION CONCURRENTE

En théorie, la programmation parallèle est plus rapide que la programmation séquentielle. La programmation parallèle se fait avec openMP et la programmation distribuée avec MPI. La programmation parallèle se fait au niveau du processeur.

¹This work was not supported by any organization

III. PROGRAMMATION PARALLÈLE

La meilleure des manières est de diviser équitablement chaque tâche. La métrique de performance au fil des années du parallélisme a une allure logarithmique. D'après la loi d'Amdahl, les communications peuvent faire une énorme différence sur nos résultats. La mémoire peut être gérée en utilisant le Non Uniform Access Memory (NUMA) qui consiste à accéder très rapidement une zone mémoire. L'unité pour le calcul des opérations des super ordinateurs est le flops. (floating operation per second). Un super ordinateur de 1994 est équivalent à nos pc en 2018. Mais la tendance semble fausser maintenant.

IV. PROGRAMMATION DISTRIBUÉE

Cette méthode permet de partager les ressources à plusieurs utilisateurs en même temps. Le but est d'optimiser l'utilisation de ces ressources en organisant les dépendances. High Performance Computing (HPC) permet d'avoir une réponse extrêmement rapide et High Throughput Computing (HTC) permet d'avoir plusieurs résultats produits. L'organisation de la distribution des données avec du parallélisme pour chaque tâche est primordiale. Les clusters personnels sont bon marché mais doivent être bien entretenus. La programmation distribuée permet d'exploiter les ressources qui ne sont pas utilisées. À terme, les clusters deviendront des grilles puis des clouds.

V. LARGE-SCALE COMPUTING

Le framework Map-reduce résume bien ce terme. Single Process Multiple Data (SPMD), c'est composé de pipeline et de workflow comme Airflow, le framework utilisé par Airbnb. Le parallélisme y est en conséquence. Plus d'un milliard d'unités sont connectées à l'heure d'aujourd'hui sur internet.

L'objectif de map-reduce est d'associer un index pour chaque page web pour le moteur de recherche en utilisant HTC avec les avantages de la location.

VI. CONCLUSION

Hadoop doit être utilisé.

ACKNOWLEDGMENT

Thanks to Johan Montagnat for his work.

REFERENCES

- [1] <http://www.i3s.unice.fr/~johan/>