

RAPPORT DE TP

Évaluation des performances d'un programme et optimisations de code par compilation

Réalisé par
Francois Flandin

Encadré par
Pr Sid Touati

Contents

1	Mesures des performances et profilage d'un programme	2
1.1	Temps d'exécution des programmes	2
1.1.1	Multiplication de matrices	2
1.1.2	Multiplication d'un vecteur par un scalaire	2
1.1.3	Addition de deux vecteurs	2
1.2	Comment calculer l'IPC, le CPI et le GFLOPS	3
1.2.1	Calculer l'IPC	3
1.2.2	Calculer le CPI	3
1.2.3	Calculer le GFLOPS	3
2	Optimisations de code avec le compilateur gcc	3
2.1	-fprofile-generate	3
3	Optimisations de code avec le compilateur d'Intel icc ou icx	3
3.1	Multiplication de matrices	3
3.2	Multiplication d'un vecteur par un scalaire	4
3.3	Addition de vecteurs	4

1 Mesures des performances et profilage d'un programme

1.1 Temps d'exécution des programmes

1.1.1 Multiplication de matrices

version statique

1	23.05	user	0.02	system	0:23.11	elapsed
2	23.33	user	0.02	system	0:23.39	elapsed
3	23.45	user	0.01	system	0:23.51	elapsed
4	24.09	user	0.02	system	0:24.19	elapsed

On remarque que le temps d'exécution reste stable. Presque 99.5% du temps d'exécution sert pour le code utilisateur, tandis que 0.001% du temps sert pour les appels système. Le reste correspond aux "pauses" du processeur.

version dynamique

1	27.01	user	0.02	system	0:27.16	elapsed
2	27.59	user	0.04	system	0:27.81	elapsed
3	27.72	user	0.03	system	0:27.89	elapsed
4	28.28	user	0.02	system	0:28.46	elapsed

1.1.2 Multiplication d'un vecteur par un scalaire

version statique

1	0.45	user	0.10	system	0:00.56	elapsed
2	0.43	user	0.11	system	0:00.55	elapsed
3	0.41	user	0.10	system	0:00.52	elapsed
4	0.43	user	0.08	system	0:00.52	elapsed

version dynamique

1	0.41	user	0.09	system	0:00.51	elapsed
2	0.39	user	0.10	system	0:00.49	elapsed
3	0.40	user	0.08	system	0:00.49	elapsed
4	0.40	user	0.08	system	0:00.49	elapsed

1.1.3 Addition de deux vecteurs

version statique

1	0.84	user	0.55	system	0:01.40	elapsed
2	0.82	user	0.58	system	0:01.41	elapsed
3	0.85	user	0.55	system	0:01.41	elapsed
4	0.81	user	0.53	system	0:01.35	elapsed

version dynamique

1	0.91	user	0.58	system	0:01.51	elapsed
2	0.89	user	0.54	system	0:01.44	elapsed
3	0.86	user	0.55	system	0:01.42	elapsed
4	0.86	user	0.53	system	0:01.40	elapsed

1.2 Comment calculer l'IPC, le CPI et le GFLOPS

1.2.1 Calculer l'IPC

On peut utiliser la commande `perf stat nom_prog` qui nous donne ceci avec `mat_mult_STATIC`:

```

1      94,108,593,606      cycles:u      #      1.594 GHz
2      264,116,293,661     instructions:u  #      2.81  insn per cycle
```

1.2.2 Calculer le CPI

La formule pour calculer le CPI est la suivante :

$$CPI = \frac{\text{nombre_de_cycles}}{\text{nombre_d'instructions}}$$

Avec la commande precedente, on a le nombre de cycles et le nombre d'instructions de l'execution d'un programme, on a donc

$$\frac{94,108,593,606}{264,116,293,661} \approx 0.35$$

1.2.3 Calculer le GFLOPS

La formule pour calculer le GFLOPS est la suivante:

$$GFLOPS = \frac{\text{Nombre_de_calculs_flottants}}{\text{Temps_d'execution}}$$

2 Optimisations de code avec le compilateur gcc

2.1 -fprofile-generate

Voici ce qu'en dit le manuel de gcc:

```

1  Enable options usually used for instrumenting application to produce
   profile useful for later recompilation with profile feedback based
   optimization. You must use -fprofile-generate both when compiling
   and when linking your program.
```

3 Optimisations de code avec le compilateur d'Intel icc ou icx

3.1 Multiplication de matrices

version statique

```

1      13.84 user      0.02 system      0:13.88 elapsed
2      13.52 user      0.01 system      0:13.57 elapsed
3      15.97 user      0.02 system      0:16.02 elapsed
4      15.98 user      0.01 system      0:16.03 elapsed
```

version dynamique

1	15.95	user	0.02	system	0:16.07	elapsed
2	13.64	user	0.02	system	0:13.77	elapsed
3	14.68	user	0.03	system	0:14.87	elapsed
4	15.58	user	0.04	system	0:15.80	elapsed

3.2 Multiplication d'un vecteur par un scalaire

version statique

1	0.00	user	0.11	system	0:00.12	elapsed
2	0.01	user	0.10	system	0:00.12	elapsed
3	0.02	user	0.10	system	0:00.12	elapsed
4	0.01	user	0.09	system	0:00.11	elapsed

version dynamique

1	0.01	user	0.08	system	0:00.10	elapsed
2	0.02	user	0.08	system	0:00.10	elapsed
3	0.01	user	0.09	system	0:00.11	elapsed
4	0.01	user	0.10	system	0:00.11	elapsed

3.3 Addition de vecteurs

version statique

1	0.10	user	0.58	system	0:00.69	elapsed
2	0.10	user	0.56	system	0:00.68	elapsed
3	0.11	user	0.60	system	0:00.72	elapsed
4	0.09	user	0.59	system	0:00.70	elapsed

version dynamique

1	0.12	user	0.58	system	0:00.71	elapsed
2	0.13	user	0.61	system	0:00.74	elapsed
3	0.11	user	0.65	system	0:00.77	elapsed
4	0.12	user	0.62	system	0:00.75	elapsed