CMP Performance Analysis with gem5

Javier Andres Tarazona Jimenez
Ingénieur Degree Programme
STIC
ENSTA Paris
Paris, France
javier-andres.tarazona@ensta-paris.fr

Jair Anderson Vasquez Torres
Ingénieur Degree Programme
STIC
ENSTA Paris
Paris, France
jair-anderson.vasquez@ensta-paris.fr

Maeva Noukoua
Ingénieur Degree Programme
STIC
ENSTA Paris
Paris, France
maeva-sandy.noukoua@ensta-paris.fr

Carlos adrian Meneses Gamboa
Ingénieur Degree Programme
STIC
ENSTA Paris
Paris, France
carlos-adrian.meneses@ensta.fr

***Abstract*—...**
***Index Terms*—...**

I. Architecture multicoeurs avec des processeurs superscalaires out-of-order (Cortex A15)

A. Stratégie adoptée pour traiter la Q9

Pour répondre à la Q9 (faire varier le nombre de threads et la largeur superscalaire, puis produire un graphe 3D des cycles), nous avons mis en place une chaîne reproductible en trois scripts :

- un script d'orchestration des simulations,
- un script gem5 SE adapté au modèle A15/o3,
- un script de post-traitement pour extraire les cycles et générer la visualisation.

Cette organisation permet de lancer une campagne complète, reprendre après erreur, tracer précisément chaque exécution, et générer automatiquement le CSV et la figure 3D demandés.

B. Script `run_q9_a15.sh`: orchestration de la campagne

Ce qu'il fait :

- lance toutes les combinaisons (`size`, `width`, `threads`) pour Q9 ;
- crée un répertoire de sortie par combinaison ;
- enregistre l'état d'avancement dans `state.tsv` (`PENDING`, `DONE`, `FAILED`) ;
- permet la reprise automatique après interruption/échec ;
- journalise chaque run dans un fichier de log dédié.

Comment il le fait :

- construit la commande gem5 avec `-cpu-type=detailed`, `-o3-width`, `-num-cpus`, et les arguments du benchmark ;
- exécute les runs séquentiellement et s'arrête au premier échec pour conserver un diagnostic clair ;
- relit `state.tsv` au redémarrage pour ignorer les cas déjà `DONE` ;
- supporte un mode de mitigation OpenMP (`-omp-active-wait`) via un fichier d'environnement passé à gem5.

C. Script `se_a15.py`: configuration gem5 pour A15/o3

Ce qu'il fait :

- instancie un système gem5 en mode syscall emulation (SE) ;
- configure des CPU de type `detailed` (modèle o3) ;
- applique la largeur superscalaire via `o3-width` ;
- exécute le binaire `test_omp` avec les paramètres `threads` et `size`.

Comment il le fait :

- s'appuie sur les options standard gem5 (`Options.addCommonOptions`, `Options.addSEOptions`) ;
- crée `num-cpus` cœurs simulés et fixe `issueWidth` pour chaque cœur ;
- configure hiérarchie mémoire/caches et lance la simulation avec `Simulation.run()` ;
- prend en charge un fichier `-env` pour injecter des variables OpenMP/libgomp si nécessaire.

D. Script `plot_q9_cycles.py`: extraction et visualisation

Ce qu'il fait :

- lit `state.tsv` et sélectionne les runs `DONE` valides ;
- extrait les cycles à partir des `stats.txt` de gem5 ;
- génère un CSV consolidé ;
- produit le graphe 3D demandé (threads, largeur, cycles).

Comment il le fait :

- vérifie les colonnes attendues de `state.tsv` et la présence des fichiers `stats.txt` ;
- récupère la métrique `system.cpu*.numCycles` et conserve la valeur maximale par run ;
- écrit `q9_cycles.csv` puis trace `q9_cycles_3d.png` avec Matplotlib ;
- signale explicitement les combinaisons manquantes/invalides non incluses dans la figure.

REFERENCES

[1] TP5, "Analyse de performances de configurations de microprocesseurs multicoeurs pour des applications parallèles," document de travaux pratiques du cours.

[2] O. Hammami, *Introduction à l'Architecture des Microprocesseurs*, ENSTA ParisTech, 828 Bvd des Maréchaux 91762 Palaiseau cedex, https://www.ensta-paristech.fr, ENSTA PARIS - Cours ES201, Année 2022.