

Adaptation de micro-architectures hétérogènes via l'utilisation de gem5 et CERE

Nicolas Derumigny Pablo de Oliveira Castro

ENS Lyon

UVSQ

5 Septembre 2016





Codelet Extractor and REplayer : Définitions

CERE extrait des morceaux de code d'une application, créant ainsi une autre application autonome, reproduisant au plus près possible l'exécution de codelet, autant dans ses instructions que dans son contexte.

Codelet

- Portion d'un application.
- *In-vivo* : fonctionnement naturel à l'intérieur de son application.
- *In-vitro* : fonctionnement de l'application produite par CERE.



Fonctionnalités de CERE

- Basé sur LLVM.
- Capture de boucles.
- Capture de régions OpenMP.
- Opération au niveau Intermediate Representation (IR).
- Capture du contexte mémoire précise au pages mémoires près.
- Deux méthodes de réchauffe du cache : Workload (chargement des données utilisées dans la boucle) et Trace (chargement des derniers accès avant la boucle).
- Outil de profilage et de mesure du temps du codelet.



CERE : Un outil d'isolation de morceaux de code

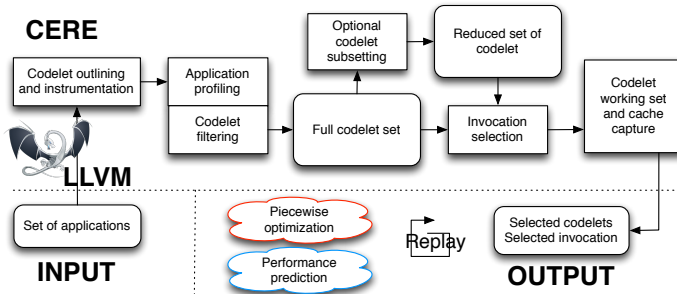


Figure – Schéma fonctionnel de CERE.

Gem5 : Un simulateur précis au cycle près



Gem5 : Le mode *syscall emulation*



Gem5 : Le mode *fullsystem*



McPAT : Un simulateur de conception et de consommation de puces

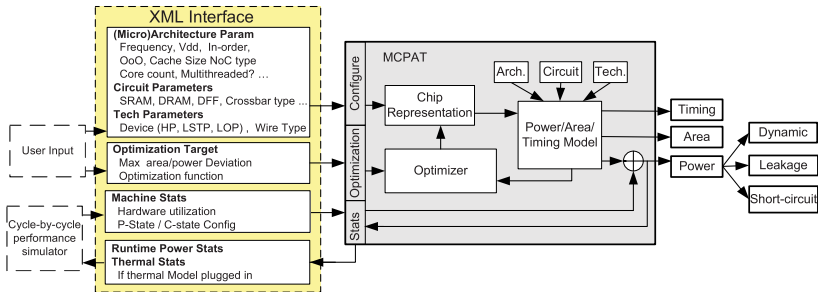


Figure – Schema fonctionnel de McPAT.

Correction de *readFunc()*



Ajout des appels système *getdents()* et *getdents64()*



Ajout des appels système *getdents()* et *getdents64()*



Simulation de quatre processeurs aux caractéristiques diverses

Nom	Fréquence	Assoc.	L2		L3
			Taille	Assoc.	
A15	1 GHz	2	1 Mo	16	Non
i5-3550	3,3-3,7 GHz	8	256 ko	8	8 Mo / 16
i5-3337U	1,8 GHz	8	256 ko	16	4 Mo / 8
Q9100	2,26 GHz	8	8 Mo	16	Non

Figure – Paramètres utilisés pour chaque CPU. Le reste de la configuration est fixe : 8 Go de RAM, processeur x86 générique, et un unique cœur simulé.



Trois codelets sur le banc d'essai



Compilation croisée depuis l'architecture x86 vers Aarch64



IS : Une première application simple

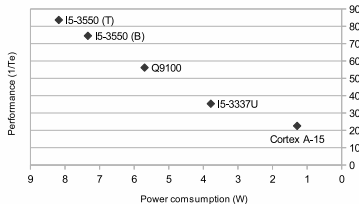


Figure – Graphique du ratio performance-consommation énergétique du codelet *IS*.

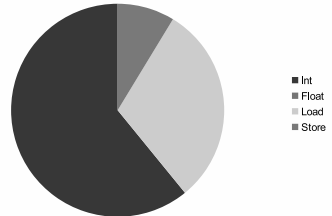


Figure – Répartition des instructions au cours de l'exécution du codelet *IS*.

Freqmine : Une application parallèle à peu de données

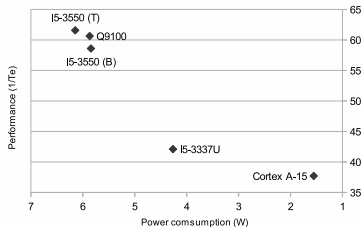


Figure – Graphique du ratio performance-consommation énergétique du codelet *Freqmine*.

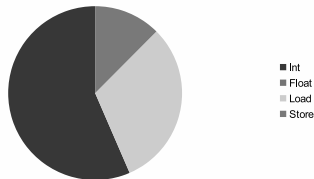


Figure – Répartition des instructions au cours de l'exécution du codelet *Freqmine*.

Blackscholes : Une application utilisant significativement les flottants

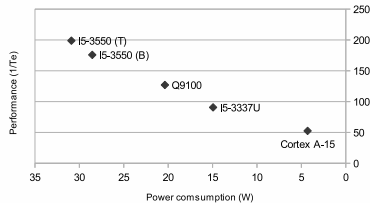


Figure – Graphique du ratio performance-consommation énergétique du codelet *Blackscholes*.

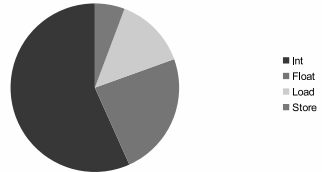


Figure – Répartition des instructions au cours de l'exécution du codelet *Blackscholes*.

Définition du ratio performance-consommation énergétique

Ratio performance-consommation énergétique

Le ratio performance-consommation énergétique I est défini par :

$$I = \frac{1}{P \cdot t_e}$$

Avec P la puissance (en W) du processeur simulé et T_e le temps d'exécution du codelet (en s).

Note : Cela correspond à l'inverse de la consommation totale du processeur.



Graphe récapitulatif

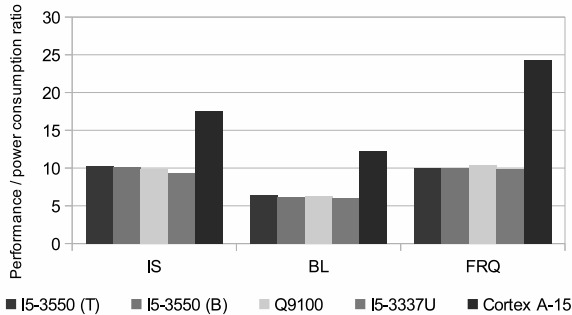


Figure – Ratio performance-consommation énergétique pour chaque codelet.



Bibliographie

