



Projet CPA

Vectorisation de boucle

Pierre Aubert, Salim Nahi, Puchen Liu

Tester si une boucle est vectorisable

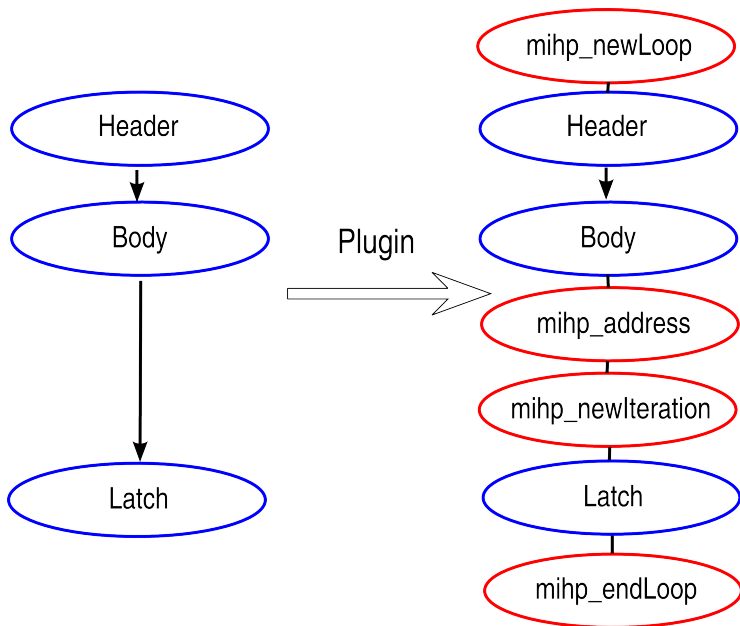
- **Plugin GCC** : insérer des sondes dans le code source
- **Bibliothèque** : analyser les accès mémoires / dire si les boucles sont vectorisables

Le plugin

Analyse

- Liste de fonction à analyser (pragma)
- Analyse des boucles les plus profondes
- Insertion d'appels de fonction à la librairie

Le plugin



C	Appels
$c[i] = a[i] + b[i];$	<code>mihp_address(&b[i], sizeof(b[i]), MIHP_READ);</code>
	<code>mihp_address(&a[i], sizeof(a[i]), MIHP_READ);</code>
	<code>mihp_address(&c[i], sizeof(c[i]), MIHP_WRITE);</code>

Gimple

D.2741 = (long unsigned int) i;

D.2742 = D.2741 * 4;

D.2743 = c + D.2742;

D.2744 = (long unsigned int) i;

D.2745 = D.2744 * 4;

D.2746 = a + D.2745;

D.2747 = *D.2746;

D.2748 = (long unsigned int) i;

D.2749 = D.2748 * 4;

D.2750 = b + D.2749;

D.2751 = *D.2750;

D.2752 = D.2747 + D.2751;

*D.2743 = D.2752;

Traduction

=> &c[i]

=> &a[i]

=> a[i]

=> &b[i]

=> b[i]

=> tmp = a[i] + b[i]

=> c[i] = tmp

Détermination de la tailles des accès mémoire

L'opérande est un pointeur

On récupère la taille du type pointé

L'opérande n'est pas un pointeur

On récupère la taille du type de la variable

Taille de l'opérande

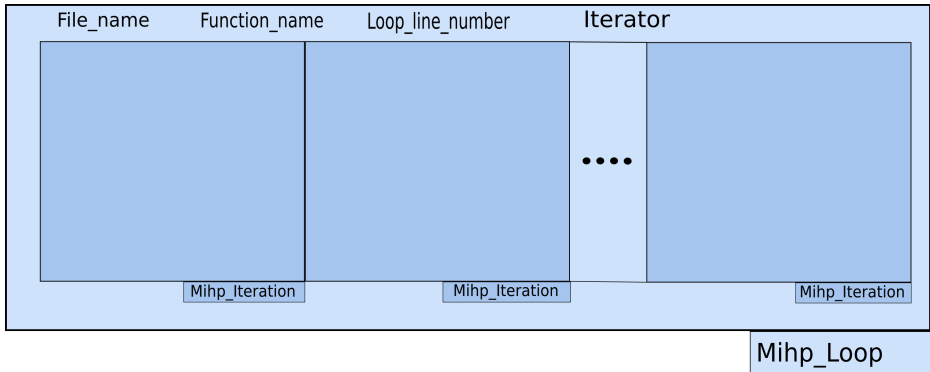
Utilisation de la fonction : `int_size_in_bytes`

La librairie d'analyse

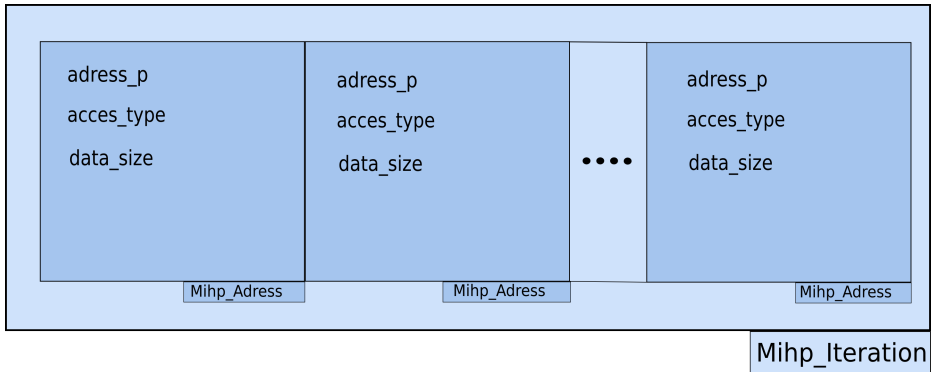
Les choses à analyser

- Lecture après lecture
- Lecture après écriture
- Écriture après lecture
- Écriture après écriture

La classe Mihp_Loop



La classe Mihp_Iteration



Conditions nécessaires

- Pas d'appels de fonction autre que sin, cos, ...
- Nombre d'itérations connu à l'avance

Merci de votre
attention