WAVE : Un langage parallèle et topologique

Éric Violard

Eric.Violard@inria.fr

Équipe CAMUS
Laboratoire ICPS/ICube

7ièmes journées nationales de la compilation

4 décembre 2013

Plan

Pourquoi ce langage ?
- Une vision topologique des programmes

Concepts de base du langage
- Collection
- Position et chemin

Perspectives

Des structures de données topologiques

- Des données séquentielles ou parallèles

Exemple : une liste, un arbre, un tableau (muni d'un ordre [partiel] de parcours), etc.

Des structures de contrôle topologiques

- Des instructions totalement ou partiellement ordonnées

Exemple : boucles `for` (ou `xfor`), directives `openMP`, etc.

Définition de « paralléliser »

Réorganiser données et instructions pour améliorer la localité spatiale et temporelle :

- réutiliser les données et instructions proches (en mémoire ou dans le temps),
- rapprocher les instructions des données ou l'inverse.

$\Rightarrow$ Un langage où la position des données ou des instructions est explicite.

Comment définir une position ?

- Dans un espace topologique
- Via un référentiel
- De manière absolue (par un système de coordonnées) : la position doit être quantifiée.
- De manière relative (par une notion de voisinage) : il n'est pas nécessaire de quantifier.

Un langage expérimental

- Pour exprimer des calculs (données + instructions) et où données ou instructions ont une position explicite.
- Un minimum de constructions syntaxiques.
- Pas de variables (seulement quelques mot-clés).
- Des macros pour les raccourcis.
- Un web-interpréteur (`TryWAVE` `http://wave.gforge.inria.fr`)

Objets de première classe

- Tout est *collection*.
- Une collection, disons C, est :
  - soit une valeur atomique (une constante d'un type de base : entier, booléen, caractère, etc.)
  - soit $C_1 ; C_2$ ("C_1 se trouve avant C_2")
  - soit $C_1 \,||\, C_2$ ("C_1 et C_2 se trouvent à des endroits différents")
  - soit (C) ("C est un élément" d'une collection englobante).

Ex :

```
1;2;3
1||2||3
1;(2||3)
```

Opérateur I

- Un opérateur *n*-aire note une valeur résultant d'une opération sur les *n* valeurs situées juste avant dans la collection.
  Ex : l'opérateur binaire +

`0;1;+`	$\Rightarrow$	`0;1;1`
`0;1;+;+`	$\Rightarrow$	`0;1;1;2`
`0;1;+;+;+`	$\Rightarrow$	`0;1;1;2;3`
`0;1;+;+;+;+`	$\Rightarrow$	`0;1;1;2;3;5`

(une suite logique)

Opérateur II

- Un opérateur est distributif par rapport à `||`.
  Ex :

```
(1||2);(3||4);+    ⇒   (1||2);(3||4);(4||6)
(1;3;+)||(2;4;+)   ⇒   (1;3;4)||(2;4;6)
```

(data-parallélisme versus parallélisme de contrôle)

Désigner une collection

- Par sa position relativement à une autre
- En suivant une direction (ou un chemin) :
  `p` ($\leftarrow$), `s` ($\rightarrow$), `u` ($\uparrow$), `d` ($\downarrow$).
  Ex :

`0;1;2;@p`	$\Rightarrow$	`0;1;2;2`
`0;1;2;@p p`	$\Rightarrow$	`0;1;2;1`
`0;1;2;@p 2`	$\Rightarrow$	`0;1;2;1`
`0;1;2;@p*`	$\Rightarrow$	`0;1;2;0`

Retrouver son chemin (les cailloux blancs du petit Poucet)

- Mémoriser un chemin (`[` et `]`), puis
  suivre le chemin dans l'autre sens (`r` rewind).
  Ex :

```
(1;2;3);(1;2;@[p*] u p d r)  ⇒  (1;2;3);(1;2;3)
```

Répétition

- Notation { } inspirée de la notation BNF. Ex :

`0;1{;+}`	$\Rightarrow$	`0;1;1;2;3;5;8;...`
`0;1{;+}2;3`	$\Rightarrow$	`0;1;1;2;3`
`0;1{;+}|[p*]|;3`	$\Rightarrow$	`0;1;1;2;3`

Une collection peut être interrompue par ?
(selon la valeur d'un booléen).

Exemple : pgcd (Euclide)

```
48;180;gcd ⇒ ...;12

Let gcd be (@u p;@u p 2{;0;=;@p 4;?;@p 5;%}).
```

Exemple : somme préfixe

```
(1||2||3||4||5||6||7||8);prefixsum
 ⇒ ...;(1||3||6||10||15||21||28||36)

Let prefixsum1 be
 (0{||@[p*]u p d r p}7);+;
 (@u p d{||@[p*]u p d r 2}3);@p p;
 (@u p d s{||@[p*]u p d s r 2}3).
```

Perspectives

- Algorithmique parallèle
- Compilation : quantifier la position (modèle polyédrique)
- Réorganisation dynamique

Références

- LISP (J. McCarthy)
- MGS (J-L. Giavitto et al.)
- xfor/multifor (P. Clauss et al.)
- X10 (IBM Research) (contributors : P. Feautrier et al.)
- PEI (E. Violard et al.)
- OCaml (X. Leroy et al.)

Merci de votre attention.

Des questions à (super)poser ?