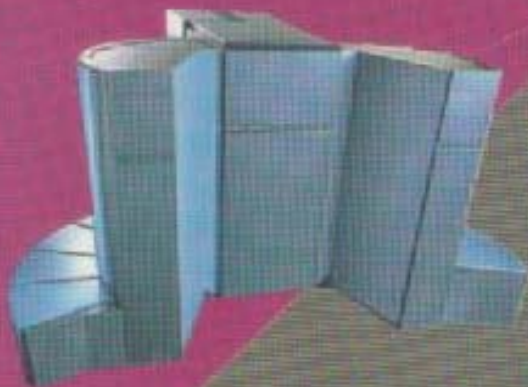


Paralations

Un modèle universel
de programmation parallèle

G.W. Sabot



DUNOD
informatique

Paralations

Un modèle universel
de programmation parallèle

Table des matières

Avant-propos	1
Remerciements	3
Chapitre 1 – Introduction	5
1. Le besoin d'un modèle parallèle	5
2. Une histoire des ordinateurs	6
3. Un besoin de rapidité insatiable	8
4. Une multitude d'architectures	10
5. L'écueil de la programmation	11
5.1. Cuisine et programmation	11
5.2. Améliorations antérieures des langages traditionnels	12
6. La solution : un jeu abstrait d'extensions	14
7. Présentation du modèle des paralations	14
7.1. Les structures de données	15
7.2. Les opérateurs	15
7.3. Stratégie	15
7.4. Etat d'implémentation actuel	16
8. Visées	17

Chapitre 2 – Description du modèle	19
1. <i>Notation</i>	19
1.1. Notation LISP	20
2. <i>Structures de données</i>	22
2.1. Propriétés des champs	24
2.2. Propriétés des paralogies	25
2.3. Détermination des index	26
2.4. Lecture des champs	26
2.5. La description de la localité est pragmatique	27
3. <i>Evaluation élémentaire</i>	27
3.1. Effets de bord	28
3.2. Synchronisation	29
3.3. Erreurs d'utilisation d'elwise	30
3.4. Désignation lexicale des variables de elwise	32
3.5. Elwise et les lambda expressions	34
4. <i>Communication par mappings</i>	35
4.1. Correspondance et mappings	36
4.2. Interaction entre sites de destination	41
5. <i>Code du programme dictionnaire</i>	42
6. <i>Dialectes et techniques de programmation</i>	43
6.1. Fonction de bibliothèque : valeur de référence (vref)	43
6.2. Fonction de bibliothèque : choisir (choose)	45
6.3. Assignation rapide pour elwise	47
7. <i>Traitement séquentiel des champs</i>	49
8. <i>Exemples de programmation</i>	50
8.1. Crible d'Eratosthène	50
8.2. Opérations sur les ensembles	51

Chapitre 3 – Complémentarité des buts du modèle 53

1. <i>Le modèle et ses buts</i>	53
2. <i>Traitement et synchronisation</i>	54

3. Communication et localité	55
4. Les distinctions améliorent la précision	56
5. Synchronisation de processus	57
6. Différentes architectures parallèles	57
6.1. Mémoire locale ou partagée	58
6.2. Nature du réseau de communications	58
6.3. Nature des processeurs	59
6.4. Modèles abstraits et matériel	59

Chapitre 4 – Paralations et programmation conventionnelle

1. Principes de programmation	61
2. Programmation structurée	62
2.1. Exemple de parallélisme imbriqué	63
2.2. Recherche A*	64
2.3. Rigidité des modèles liés à une architecture	74
2.4. Héritage de la localité	75
3. Styles de programmation	77
3.1. Exemple d'effet de bord	77
3.2. Style fonctionnel	78
3.3. Influence du langage sur le style	80
4. Principes de programmation (suite)	81
5. Complexité des programmes	88
5.1. Complexité parallèle	89
5.2. Complexité et paralations	90
6. Dilemmes classiques de programmation	93
7. Autres langages de base : le C	94
7.1 Structures de données	95
7.2. Evaluation élémentaire	96
7.3. Match	96
7.4. Move	97
7.5. Gestion de mémoire	97
7.6. Exemple du crible d'Eratosthène	97

Chapitre 5 – Facilité d'utilisation	99
1. <i>Considérations pratiques</i>	99
2. <i>Générateur de nombres premiers</i>	100
3. <i>Fonctions de bibliothèque (suite)</i>	102
3.1. Fonction de bibliothèque : collapse (concentrer)	102
3.2. Fonction de bibliothèque : collect (collecter)	103
3.2. Fonction de bibliothèque : expand (développer)	105
3.2. Fonction de bibliothèque : fork (bifurquer)	105
4. <i>Tri rapide</i>	106
4.1. Approche fonctionnelle	107
4.2. Tri rapide sur place	108
5. <i>Entrées/sorties : paralations système</i>	111
5.1. Tracé de ligne	112
6. <i>Plus court chemin dans un graphe</i>	115
6.1. Représentation d'un graphe	115
6.2. Mesurer-chemin en Paralation Lisp	116
6.3. Mappings possibles	119
 Chapitre 6 – Sémantique des paralations	 121
1. <i>Pourquoi une sémantique ?</i>	121
1.1. Différentes sémantiques	122
1.2. Sémantique de Paralation Lisp	123
2. <i>Tiny Paralation Lisp</i>	124
2.1 Structures de données	124
2.2. Fonctions utilitaires	126
2.3. Elwise	127
2.4. Match	132
2.5. <- (move)	133
3. <i>Sémantique des utilitaires</i>	136
3.1. Fonctions de bibliothèque	137
3.2. Fonctions de combinaison	138

Chapitre 7 – Stratégies d'implémentation	143
1. <i>Présentation</i>	143
1.1. Pourquoi décrire les implémentations ?	143
1.2. Choix du langage de base	144
1.3. Combien de processeurs ?	145
2. <i>Structures de données</i>	146
2.1. Champs imbriqués	147
2.2. Héritage de localité	149
2.3. Représentation du mapping	151
2.4. Nature des optimisations	151
3. <i>Evaluation élémentaire</i>	151
3.1. Elwise pour architectures SIMD	152
3.2. Elwise pour architectures MIMD	153
3.3. Implémentation par messages	153
3.4. Implémentation par mémoire partagée	155
3.5. Interférences entre sites	156
4. <i>Match</i>	157
4.1. Recherche parallèle dans une table	159
5. <i>Move</i>	162
5.1. Un move séquentiel à mappings canoniques efficace	162
5.2. Move parallèle	164
6. <i>Fonctions de bibliothèque</i>	165
6.1. Constructions idiomatiques	166
7. <i>Implémentations partielles</i>	167
8. <i>La machine cible idéale</i>	168
9. <i>Implémentations actuelles</i>	169

Chapitre 8 – Forme	171
---------------------------------	-----

1. <i>Qu'entend-on par forme ?</i>	171
1.1. Localité par la forme	173
1.2. Accès par la forme	174

<i>2. Forme en Paralation Lisp</i>	176
2.1. Forme et localité en Paralation Lisp	176
2.2. Forme et accès en Paralation Lisp	178
2.3. Biblioyhèque de formes : réseau à maillage carré	179
<i>3. Autres approches du parallélisme</i>	182
3.1. Extraction de nombres premiers et matrices	183
3.2. Ensembles de données non structurés	184

Chapitre 9 – Puissance des paralations

<i>1. Modèles équivalents</i>	187
<i>2. Universalité</i>	187
2.1. Emulation du PRAM	188
2.2. Efficacité de la simulation PRAM	191
<i>3. Irréductibilité</i>	191
<i>4. Modèles des paralations alternatifs</i>	193
4.1. Une alternative	194

Chapitre 10 – Comparaison avec d'autres modèles

<i>1. Comparaison de langages</i>	197
<i>2. APL : un pionnier des modèles explicitement parallèles</i>	198
2.1. APL	198
<i>3. Le modèle PRAM/MIMD</i>	201
3.1. Imperfections du modèle PRAM	202
<i>4. Modèles de programmation SIMD</i>	203
4.1. Inconvénients des modèles SIMD	205
<i>5. Modèles implicitement parallèles</i>	210
5.1. Précision, prévision des coûts et transparence	213
<i>6. Langages orientés objet</i>	215
<i>7. C*</i>	216

8. <i>Connection Machine Lisp</i>	219
8.1. Notation alpha	221
8.2. Séparation floue entre traitement et communication	223
9. <i>MultiLisp</i>	225
10. <i>Extensions proposées pour Fortran 8x</i>	227
10.1. Motivations	227
10.2. Description des extensions	228
10.3. Résumé des extensions 8x	230
11. <i>Langages de quatrième génération</i>	230
11.1. Les paralations ne sont pas des relations	231
11.2. Paralations visibles par l'utilisateur	231
11.3. Allocation dynamique	232
11.4. Généralité des paralations	233
11.5. Concepts propres aux bases de données	234
12. <i>Parallélisme et conception de langages</i>	234
13. <i>Tableaux de comparaison</i>	236
13.1. Caractéristiques générales	236
13.2. Structures de données	237
13.2. Traitement	238
13.4. Communication	239
 Chapitre 11 – Conclusion	 241
1. <i>Synthèse</i>	241
2. <i>Contributions de cette étude</i>	243
2.1. Imperfections	243
2.2. Succès	244
 Appendice A – Intégration à Lisp	 247
1. <i>Commodités syntaxiques</i>	250

Appendice B – Programmes Lisp	253
-------------------------------------	-----

1. Match canonisant	253
---------------------------	-----

2. Préfixe parallèle	253
----------------------------	-----

Appendice C – Glossaire	257
-------------------------------	-----

Bibliographie	269
---------------------	-----

Index	281
-------------	-----