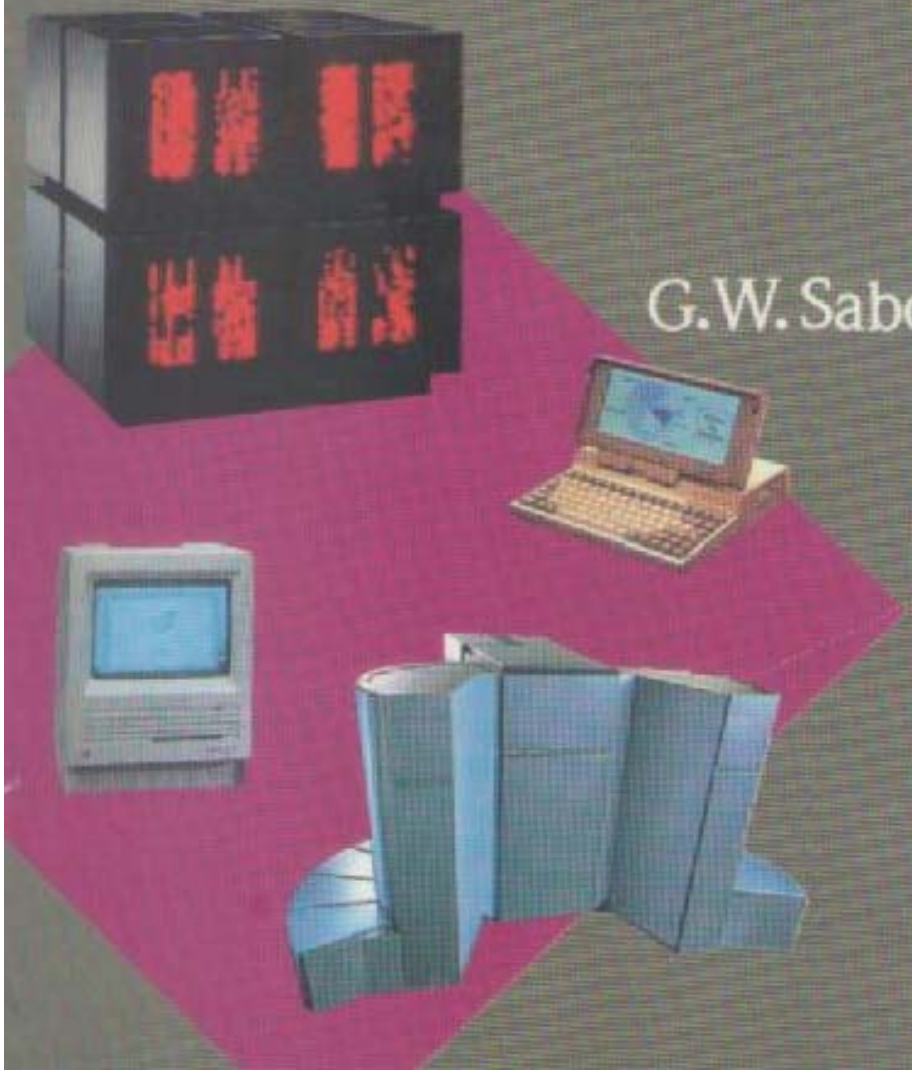
Paralations

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. Notation 19
1.1. Notation LISP 20
2. Structures de données 22
2.1. Propriétés des champs 24
2.2. Propriétés des paralations 25
2.3. Détermination des index 26
2.4. Lecture des champs 26
2.5. La description de la localité est pragmatique 27
3. Evaluation élémentaire 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. Communication par mappings 35
4.1. Correspondance et mappings 36
4.2. Interaction entre sites de destination 41
5. Code du programme dictionnaire 42
6. Dialectes et techniques de programmation 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. Traitement séquentiel des champs 49
8. Exemples de programmation 50
8.1. Crible d'Eratosthène 50
8.2. Opérations sur les ensembles 51

Chapitre 3 – Complémentarité des buts du modèle 53

1. Le modèle et ses buts 53
2. Traitement et synchronisation 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 61

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. Considérations pratiques 99
2. Générateur de nombres premiers 100
3. Fonctions de bibliothèque (suite) 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. Tri rapide 106
4.1. Approche fonctionnelle 107
4.2. Tri rapide sur place 108
5. Entrées/sorties : paralations système 111
5.1. Tracé de ligne 112
6. Plus court chemin dans un graphe 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. Pourquoi une sémantique ? 121
1.1. Différentes sémantiques 122
1.2. Sémantique de Paralation Lisp 123
2. Tiny Paralation Lisp 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. Sémantique des utilitaires 136
3.1. Fonctions de bibliothèque 137
3.2. Fonctions de combinaison 138

Chapitre 7 – Stratégies d'implémentation 143

1. Présentation 143
1.1. Pourquoi décrire les implémentations ? 143
1.2. Choix du langage de base 144
1.3. Combien de processeurs ? 145
2. Structures de données 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. Evaluation élémentaire 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. Match 157
4.1. Recherche parallèle dans une table 159
5. Move 162
5.1. Un move séquentiel à mappings canoniques efficace 162
5.2. Move parallèle 164
6. Fonctions de bibliothèque 165
6.1. Constructions idiomatiques 166
7. Implémentations partielles 167
8. La machine cible idéale 168
9. Implémentations actuelles 169

Chapitre 8 – Forme 171

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

2. Forme en Paralation Lisp 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
3. Autres approches du parallélisme 182
3.1. Extraction de nombres premiers et matrices 183
3.2. Ensembles de données non structurés 184

Chapitre 9 – Puissance des paralations 187

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

Chapitre 10 – Comparaison avec d'autres modèles 197

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

8. Connection Machine Lisp 219
8.1. Notation alpha 221
8.2. Séparation floue entre traitement et communication 223
9. MultiLisp 225
10. Extensions proposées pour Fortran 8x 227
10.1. Motivations 227
10.2. Description des extensions 228
10.3. Résumé des extensions 8x 230
11. Langages de quatrième génération 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. Parallélisme et conception de langages 234
13. Tableaux de comparaison 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. Synthèse 241
2. Contributions de cette étude 243
2.1. Imperfections 243
2.2. Succès 244

Appendice A – Intégration à Lisp 247

1. Commodités syntaxiques 250

Appendice B – Programmes Lisp 253

1. Match canonisant 253

2. Préfixe parallèle 253

Appendice C – Glossaire 257

Bibliographie 269

Index 281