

# 目次

<b>第1章</b>	<b>はじめに</b>	<b>1</b>
1.1	どういう計算が並列にできるか . . . . .	2
1.2	どのような方針で並列計算を組み立てるか . . . . .	3
1.3	どのような並列計算の方法があるか . . . . .	4
1.4	並列計算の性能の評価 . . . . .	5
1.5	1990年代までの並列計算の動向 . . . . .	7
1.6	1990年代後半以降の動向 . . . . .	11
1.7	近年の日本のスパコン . . . . .	17
1.8	分散コンピューティング, グリッドコンピューティング . . . . .	18
<b>第2章</b>	<b>ベクトル並列</b>	<b>21</b>
2.1	ベクトル処理 . . . . .	21
2.1.1	演算パイプライン処理 . . . . .	21
2.1.2	ベクトル計算機の基本構成 . . . . .	23
2.1.3	ベクトル計算機の性能 . . . . .	25
2.1.4	ベクトル化率 . . . . .	29
2.1.5	do ループとベクトル処理 . . . . .	30
2.1.6	ベクトル化の条件 . . . . .	33
2.1.7	ベクトル演算の高速化 . . . . .	37
2.2	ベクトル並列の方法と具体例 . . . . .	41
2.2.1	ベクトル並列計算機と並列化インターフェース . . . . .	41
2.2.2	並列処理の実行イメージ . . . . .	43

2.2.3	グローバル変数とローカル変数 . . . . .	45
2.2.4	並列処理プログラムの記述法 . . . . .	46
2.2.5	並列処理プログラム例 1 . . . . .	46
2.2.6	配列の分割 . . . . .	50
2.2.7	並列処理プログラム例 2 . . . . .	54
2.2.8	グローバル変数の resident 指定 . . . . .	58
2.2.9	拡張 spread do 構文 . . . . .	60
2.2.10	配列式の並列化 . . . . .	61
2.2.11	プログラムの分割 . . . . .	61
2.2.12	グローバル変数・ローカル変数間のデータ転送 . . . . .	62
2.2.13	重複ローカル配列の転送 . . . . .	66
2.2.14	グローバル関数 . . . . .	67
2.2.15	ブロードキャスト . . . . .	70
2.2.16	入出力 . . . . .	71
<b>第 3 章</b>	<b>スレッド並列</b>	<b>73</b>
3.1	共有メモリ型スカラ並列計算機とスレッド並列 . . . . .	73
3.2	並列化率 . . . . .	75
3.3	do ループとスレッド並列 . . . . .	75
3.4	スレッド並列プログラムのチューニング . . . . .	79
3.4.1	キャッシュ競合 . . . . .	79
3.4.2	プロファイラ . . . . .	80
3.4.3	パディング . . . . .	82
3.4.4	配列コピーの回避 . . . . .	84
<b>第 4 章</b>	<b>プロセス並列 ——MPI による並列プログラミング——</b>	<b>95</b>
4.1	MPI とは . . . . .	95
4.2	MPI プログラミングの基礎知識 . . . . .	97
4.2.1	MPI プログラムの基本的構成 . . . . .	97
4.2.2	主な MPI の関数 . . . . .	100

4.2.3	MPI 基本データ型と MPI 定義済み演算 . . . . .	100
4.2.4	初歩的な MPI プログラム . . . . .	102
4.2.5	ブロッキング通信とノンブロッキング通信 . . . . .	104
4.2.6	送信モード . . . . .	105
4.2.7	配列の分割 . . . . .	106
4.2.8	時間計測 . . . . .	107
4.2.9	よく利用する MPI の関数 . . . . .	107
4.3	並列化の実際例 . . . . .	108
4.3.1	2 次元目でブロック分割する場合 . . . . .	108
4.3.2	1 次元目でブロック分割する場合 . . . . .	110
4.3.3	1, 2 次元目でブロック分割する場合 . . . . .	113
4.3.4	プロセッサのメモリを最大限に活用する場合 . . . . .	116
4.4	派生データ型とトポロジー . . . . .	118
4.4.1	派生データ型 . . . . .	118
4.4.2	トポロジー . . . . .	123
<b>第 5 章</b>	<b>並列計算機のアーキテクチャ</b>	<b>129</b>
5.1	概要 . . . . .	129
5.2	ベクトルアーキテクチャ . . . . .	133
5.2.1	ベクトルプロセッサの構成 . . . . .	133
5.2.2	ベクトル演算パイプライン . . . . .	135
5.2.3	連結処理 . . . . .	136
5.2.4	ベクトルマスク機能 . . . . .	137
5.2.5	ベクトルプロセッサのメモリシステム . . . . .	138
5.3	共有メモリ型並列アーキテクチャ . . . . .	139
5.3.1	スヌープキャッシュ . . . . .	140
5.3.2	ライトスルー型 . . . . .	142
5.3.3	ライトバック・インバリデート型 . . . . .	142
5.3.4	ライトバック・アップデート型 . . . . .	145
5.3.5	同期方式 . . . . .	147

5.4	分散メモリ型並列アーキテクチャ	148
5.4.1	相互結合網のトポロジ	148
5.4.2	メッセージのルーティング	151
5.4.3	転送方式	153
5.4.4	デッドロック	154
5.5	PC クラスタ	156
5.5.1	Beowulf 型 PC クラスタ	157
5.5.2	SCore 型クラスタ	157
<b>第 6 章</b>	<b>グリッドコンピューティング</b>	<b>159</b>
6.1	グリッドコンピューティングとは	159
6.1.1	グリッドの定義	159
6.1.2	グリッドの歴史的背景	161
6.2	グリッドの展開	162
6.2.1	米国のグリッド (Teragrid から XSEDE へ)	162
6.2.2	ヨーロッパのグリッド	164
6.2.3	日本のグリッド関連プロジェクト	167
6.3	グリッドコンピューティングの利用形態	167
6.3.1	分散並列スーパーコンピューティング (メタコンピューティング)	168
6.3.2	高スループットコンピューティング	168
6.3.3	Data-Intensive Computing (データグリッド)	169
6.3.4	協調グリッド	170
6.3.5	サーバーグリッド	170
6.4	グリッドのしくみ	170
6.4.1	グリッドの階層構造	170
6.4.2	Globus Toolkit の概略	172
6.5	NAREGI プロジェクトの概要 (2003~2008 年)	173
6.5.1	プロジェクトの目的	173
6.5.2	プロジェクト参加機関	174

6.5.3	グリッド研究開発拠点の研究テーマ . . . . .	174
6.5.4	NAREGI の主要な機能 . . . . .	176
6.5.5	最終実証環境の構築とソフトウェアの公開 . . . . .	178
6.6	e-サイエンスへの展開と事例 . . . . .	180
6.6.1	e-サイエンスとは . . . . .	180
6.6.2	Cactus . . . . .	180
6.6.3	テラグリッドの Science Gateway . . . . .	182
6.6.4	Application Hosting Environment (AHE) . . . . .	182
6.6.5	RENKEI プロジェクト概要 . . . . .	183
6.7	最近の動向 . . . . .	187
6.7.1	国際標準・国際互換 . . . . .	187
6.7.2	グリッドのインフラ化・基盤化 . . . . .	187
6.7.3	階層的かつシームレスな計算研究環境 . . . . .	188
6.8	クラウドコンピューティングと HPC . . . . .	190
6.8.1	クラウドとは . . . . .	190
6.8.2	クラウドのサービスの形態 . . . . .	191
6.8.3	クラウドの HPC 分野への適用 . . . . .	193
6.8.4	グリッドとクラウドの連携 . . . . .	196
6.9	おわりに . . . . .	196
<b>付録</b>	<b>MPI でよく利用する基本的な関数</b>	<b>205</b>
<b>索 引</b>		<b>217</b>