

QA  
76.9  
C58  
G755  
2003

# Grid Computing

## Making the Global Infrastructure a Reality

**Edited by**

**FRAN BERMAN**

*University of California, San Diego & San Diego Supercomputer Center, USA*

**GEOFFREY FOX**

*Community Grids Lab, Indiana University, USA*

**TONY HEY**

*Director e-Science Core Programme & University of Southampton, UK*

**Kurt F. Wendt Library**

For more information about this document  
contact the Reference Desk at Wendt Library  
(askwendt@engr.wisc.edu) or 262-0696



**WILEY**

---

# Contents

---

	<b>Overview of the book: Grid computing – making the global infrastructure a reality</b>	<b>3</b>
	<i>Fran Berman, Geoffrey Fox, and Tony Hey</i>	
<b>1</b>	<b>The Grid: past, present, future</b>	<b>9</b>
	<i>Fran Berman, Geoffrey Fox, and Tony Hey</i>	
<b>2</b>	<b>The Grid: A new infrastructure for 21st century science</b>	<b>51</b>
	<i>Ian Foster</i>	
<b>3</b>	<b>The evolution of the Grid</b>	<b>65</b>
	<i>David De Roure, Mark A. Baker, Nicholas R. Jennings, and Nigel R. Shadbolt</i>	
<b>4</b>	<b>Software infrastructure for the I-WAY high-performance distributed computing experiment</b>	<b>101</b>
	<i>Ian Foster, Jonathan Geisler, Bill Nickless, Warren Smith, and Steven Tuecke</i>	
<b>5</b>	<b>Implementing production Grids</b>	<b>117</b>
	<i>William E. Johnston, The NASA IPG Engineering Team, and The DOE Science Grid Team</i>	
<b>6</b>	<b>The anatomy of the Grid</b>	<b>171</b>
	<i>Ian Foster, Carl Kesselman, and Steven Tuecke</i>	
<b>7</b>	<b>Rationale for choosing the Open Grid Services Architecture</b>	<b>199</b>
	<i>Malcolm Atkinson</i>	
<b>8</b>	<b>The physiology of the Grid</b>	<b>217</b>
	<i>Ian Foster, Carl Kesselman, Jeffrey M. Nick, and Steven Tuecke</i>	
<b>9</b>	<b>Grid Web services and application factories</b>	<b>251</b>
	<i>Dennis Gannon, Rachana Ananthkrishnan, Sriram Krishnan, Madhusudhan Govindaraju, Lavanya Ramakrishnan, and Aleksander Slominski</i>	

<b>10</b>	<b>From Legion to Avaki: the persistence of vision</b>	<b>265</b>
	<i>Andrew S. Grimshaw, Anand Natrajan, Marty A. Humphrey, Michael J. Lewis, Anh Nguyen-Tuong, John F. Karpovich, Mark M. Morgan, and Adam J. Ferrari</i>	
<b>11</b>	<b>Condor and the Grid</b>	<b>299</b>
	<i>Douglas Thain, Todd Tannenbaum, and Miron Livny</i>	
<b>12</b>	<b>Architecture of a commercial enterprise desktop Grid: the Entropia system</b>	<b>337</b>
	<i>Andrew A. Chien</i>	
<b>13</b>	<b>Autonomic computing and Grid</b>	<b>351</b>
	<i>Pratap Pattnaik, Kattamuri Ekanadham, and Joefon Jann</i>	
<b>14</b>	<b>Databases and the Grid</b>	<b>363</b>
	<i>Paul Watson</i>	
<b>15</b>	<b>The Open Grid Services Architecture, and Data Grids</b>	<b>385</b>
	<i>Peter Z. Kunszt and Leanne P. Guy</i>	
<b>16</b>	<b>Virtualization services for Data Grids</b>	<b>409</b>
	<i>Reagan W. Moore and Chaitan Baru</i>	
<b>17</b>	<b>The Semantic Grid: a future e-Science infrastructure</b>	<b>437</b>
	<i>David De Roure, Nicholas R. Jennings, and Nigel R. Shadbolt</i>	
<b>18</b>	<b>Peer-to-peer Grids</b>	<b>471</b>
	<i>Geoffrey Fox, Dennis Gannon, Sung-Hoon Ko, Sangmi-Lee, Shrideep Pallickara, Marlon Pierce, Xiaohong Qiu, Xi Rao, Ahmet Uyar, Minjun Wang, and Wenjun Wu</i>	
<b>19</b>	<b>Peer-to-peer Grid databases for Web service discovery</b>	<b>491</b>
	<i>Wolfgang Hoschek</i>	
<b>20</b>	<b>Overview of Grid computing environments</b>	<b>543</b>
	<i>Geoffrey Fox, Dennis Gannon, and Mary Thomas</i>	
<b>21</b>	<b>Grid programming models: current tools, issues and directions</b>	<b>555</b>
	<i>Craig Lee and Domenico Talia</i>	
<b>22</b>	<b>NaradaBrokering: an event-based infrastructure for building scalable durable peer-to-peer Grids</b>	<b>579</b>
	<i>Geoffrey Fox and Shrideep Pallickara</i>	

---

<b>23</b>	<b>Classifying and enabling Grid applications</b>	<b>601</b>
	<i>Gabrielle Allen, Tom Goodale, Michael Russell, Edward Seidel, and John Shalf</i>	
<b>24</b>	<b>NetSolve: past, present, and future – a look at a Grid enabled server</b>	<b>615</b>
	<i>Sudesh Agrawal, Jack Dongarra, Keith Seymour, and Sathish Vadhiyar</i>	
<b>25</b>	<b>Ninf-G: a GridRPC system on the Globus toolkit</b>	<b>625</b>
	<i>Hidemoto Nakada, Yoshio Tanaka, Satoshi Matsuoka, and Satoshi Sekiguchi</i>	
<b>26</b>	<b>Commodity Grid kits – middleware for building Grid computing environments</b>	<b>639</b>
	<i>Gregor von Laszewski, Jarek Gawor, Sriram Krishnan, and Keith Jackson</i>	
<b>27</b>	<b>The Grid portal development kit</b>	<b>657</b>
	<i>Jason Novotny</i>	
<b>28</b>	<b>Building Grid computing portals: the NPACI Grid portal toolkit</b>	<b>675</b>
	<i>Mary P. Thomas and John R. Boisseau</i>	
<b>29</b>	<b>Unicore and the Open Grid Services Architecture</b>	<b>701</b>
	<i>David Snelling</i>	
<b>30</b>	<b>Distributed object-based Grid computing environments</b>	<b>713</b>
	<i>Tomasz Haupt and Marlon E. Pierce</i>	
<b>31</b>	<b>DISCOVER: a computational collaboratory for interactive Grid applications</b>	<b>729</b>
	<i>Vijay Mann and Manish Parashar</i>	
<b>32</b>	<b>Grid resource allocation and control using computational economies</b>	<b>747</b>
	<i>Rich Wolski, John Brevik, James S. Plank, and Todd Bryan</i>	
<b>33</b>	<b>Parameter sweeps on the Grid with APST</b>	<b>773</b>
	<i>Henri Casanova and Fran Berman</i>	
<b>34</b>	<b>Storage manager and file transfer Web services</b>	<b>789</b>
	<i>William A. Watson III, Ying Chen, Jie Chen, and Walt Akers</i>	
<b>35</b>	<b>Application overview for the book: Grid computing – making the global infrastructure a reality</b>	<b>805</b>
	<i>Fran Berman, Geoffrey Fox, and Tony Hey</i>	

---

<b>36</b>	<b>The data deluge: an e-Science perspective</b>	<b>809</b>
	<i>Tony Hey and Anne Trefethen</i>	
<b>37</b>	<b>Metacomputing</b>	<b>825</b>
	<i>Larry Smarr and Charles E. Catlett</i>	
<b>38</b>	<b>Grids and the virtual observatory</b>	<b>837</b>
	<i>Roy Williams</i>	
<b>39</b>	<b>Data-intensive Grids for high-energy physics</b>	<b>859</b>
	<i>Julian J. Bunn and Harvey B. Newman</i>	
<b>40</b>	<b>The new biology and the Grid</b>	<b>907</b>
	<i>Kim Baldridge and Philip E. Bourne</i>	
<b>41</b>	<b>eDiamond: a Grid-enabled federated database of annotated mammograms</b>	<b>923</b>
	<i>Michael Brady, David Gavaghan, Andrew Simpson, Miguel Mulet Parada, and Ralph Highnam</i>	
<b>42</b>	<b>Combinatorial chemistry and the Grid</b>	<b>945</b>
	<i>Jeremy G. Frey, Mark Bradley, Jonathan W. Essex, Michael B. Hursthouse, Susan M. Lewis, Michael M. Luck, Luc Moreau, Dave C. De Roure, Mike Surridge, and Alan H. Welsh</i>	
<b>43</b>	<b>Education and the enterprise with the Grid</b>	<b>963</b>
	<i>Geoffrey Fox</i>	
<b>Index</b>		<b>977</b>
	Views of the Grid	1002
	Indirect Glossary	1004
	List of Grid Projects	1007

---

# Table of Contents

---

<b>List of Contributors</b>	<b>xxxi</b>
<b>About the Editors</b>	<b>xliii</b>
<b>Foreword</b>	<b>xliv</b>
<b>PART A Overview and motivation</b>	<b>1</b>
<b>Overview of the book: Grid computing – making the global infrastructure a reality</b>	<b>3</b>
<i>Fran Berman, Geoffrey Fox, and Tony Hey</i>	
Summary of the book	3
Part A: overview	5
Part B: architecture and technologies of the Grid	5
Part C: Grid computing environments	6
Part D: Grid applications	7
References	8
<b>1 The Grid: past, present, future</b>	<b>9</b>
<i>Fran Berman, Geoffrey Fox, and Tony Hey</i>	
1.1 The Grid	9
Box 1.1 Chapter Overview	11
1.2 Beginnings of the Grid	12
1.3 A community Grid model	15
1.4 Building blocks of the Grid	16
1.4.1 Networks	16
1.4.2 Computational ‘nodes’ on the Grid	22
1.4.3 Pulling it all together	24
1.4.4 Common infrastructure: standards	28
1.5 Grid applications and application middleware	29
1.5.1 Life science applications	29
1.5.2 Engineering-oriented applications	31
1.5.3 Data-oriented applications	33
1.5.4 Physical science applications	35
1.5.5 Trends in research: e-Science in a collaborative	36

1.5.6	Commercial Applications	38
1.5.7	Application Summary	39
1.6	Futures – Grids on the horizon	41
1.6.1	Adaptative and autonomic computing	41
1.6.2	Grid programming environments	43
1.6.3	New Technologies	44
1.6.4	Grid Policy and Grid Economies	45
1.6.5	A final note	45
	Box 1.2 Overview of Parts A and B of book	45
	References	46
<b>2</b>	<b>The Grid: A new infrastructure for 21st century science</b>	<b>51</b>
	<i>Ian Foster</i>	
2.1	Technology trends	52
2.2	Infrastructure and tools	54
2.3	Grid architecture	55
2.4	Authentication, authorization, and policy	58
2.5	Current status and future directions	60
	Acknowledgments	61
	References	62
<b>3</b>	<b>The evolution of the Grid</b>	<b>65</b>
	<i>David De Roure, Mark A. Baker, Nicholas R. Jennings, and Nigel R. Shadbolt</i>	
3.1	Introduction	65
3.2	The evolution of the Grid: the first generation	66
3.2.1	FAFNER	66
3.2.2	I-WAY	67
3.2.3	A summary of early experiences	69
3.3	The evolution of the Grid: the second generation	69
3.3.1	Requirements for the data and computation infrastructure	70
3.3.2	Second-generation core technologies	72
3.3.3	Distributed object systems	74
3.3.4	Grid resource brokers and schedulers	76
3.3.5	Grid portals	77
3.3.6	Integrated systems	79
3.3.7	Peer-to-Peer computing	82
3.3.8	A summary of experiences of the second generation	84
3.4	The evolution of the Grid: the third generation	84
3.4.1	Service-oriented architectures	85
3.4.2	Information aspects: relationship with the World Wide Web	89
3.4.3	Live information systems	91
3.5	Summary and discussion	93
3.5.1	Comparing the Web and the Grid	94
3.5.2	The Semantic Grid	94

3.5.3	Research issues	95
	References	97
<b>4</b>	<b>Software infrastructure for the I-WAY high-performance distributed computing experiment</b>	<b>101</b>
	<i>Ian Foster, Jonathan Geisler, Bill Nickless, Warren Smith, and Steven Tuecke</i>	
4.1	Introduction	101
4.2	The I-WAY experiment	103
	4.2.1 Applications	103
	4.2.2 The I-WAY network	103
4.3	I-WAY infrastructure	104
	4.3.1 Requirements	104
	4.3.2 Design overview	105
4.4	Point of presence machines	105
	4.4.1 I-POP design	106
	4.4.2 I-POP discussion	107
4.5	Scheduler	107
	4.5.1 Scheduler design	108
	4.5.2 Scheduler discussion	108
4.6	Security	109
	4.6.1 Security design	109
	4.6.2 Security discussion	110
4.7	Parallel programming tools	111
	4.7.1 Parallel tools design	111
	4.7.2 Parallel tools discussion	112
4.8	File systems	112
4.9	Related work	113
4.10	Conclusions	114
	Acknowledgments	114
	References	115
<b>5</b>	<b>Implementing production Grids</b>	<b>117</b>
	<i>William E. Johnston, The NASA IPG Engineering Team, and The DOE Science Grid Team</i>	
5.1	Introduction: lessons learned for building large-scale Grids	117
5.2	The Grid context	119
5.3	The anticipated Grid usage model will determine what gets deployed, and when	121
	5.3.1 Grid computing models	121
	5.3.2 Grid data models	125
5.4	Grid support for collaboration	128
5.5	Building an initial multisite, computational and data Grid	129
	5.5.1 The Grid building team	129
	5.5.2 Grid resources	129



5.5.3	Build the initial test bed	129
5.6	Cross-site trust management	130
5.6.1	Trust	131
5.6.2	Establishing an operational CA	132
5.7	Transition to a prototype production Grid	135
5.7.1	First steps	135
5.7.2	Defining/understanding the extent of 'your' Grid	135
5.7.3	The model for the Grid Information System	135
5.7.4	Local authorization	137
5.7.5	Site security issues	138
5.7.6	High performance communications issues	138
5.7.7	Batch schedulers	139
5.7.8	Preparing for users	140
5.7.9	Moving from test bed to prototype production Grid	140
5.7.10	Grid systems administration tools	141
5.7.11	Data management and your Grid service model	141
5.7.12	Take good care of the users as early as possible	142
5.8	Conclusions	144
	Acknowledgments	144
	Notes and References	145

## **PART B Grid architecture and technologies 169**

### **6 The anatomy of the Grid 171**

*Ian Foster, Carl Kesselman, and Steven Tuecke*

6.1	Introduction	171
6.2	The emergence of virtual organizations	174
6.3	The nature of Grid architecture	176
6.4	Grid architecture description	177
6.4.1	Fabric: Interfaces to local control	178
6.4.2	Connectivity: Communicating easily and securely	180
6.4.3	Resource: Sharing single resources	181
6.4.4	Collective: Coordinating multiple resources	182
6.4.5	Applications	185
6.5	Grid architecture in practice	185
6.6	'On the Grid': the need for intergrid protocols	186
6.7	Relationships with other technologies	187
6.7.1	World Wide Web	187
6.7.2	Application and storage service providers	188
6.7.3	Enterprise computing systems	188
6.7.4	Internet and peer-to-peer computing	189
6.8	Other perspectives on Grids	189
6.9	Summary	191
	Acknowledgments	191

	Appendix: definitions	191
	References	194
<b>7</b>	<b>Rationale for choosing the Open Grid Services Architecture</b>	<b>199</b>
	<i>Malcolm Atkinson</i>	
7.1	Introduction	199
7.2	The significance of data for e-Science	200
7.3	Building on an OGSA platform	201
	7.3.1 Web services	201
	7.3.2 The Open Grid Services Architecture	202
7.4	Case for OGSA	203
7.5	The challenge of OGSA	205
7.6	Planning the United Kingdom's OGSA contributions	206
7.7	Establishing common infrastructure	206
	7.7.1 Grid services hosting environment	207
	7.7.2 Standard types	207
7.8	Baseline database access	208
	7.8.1 An overview of the OGSA-DAI architecture	208
7.9	Baseline logging infrastructure	211
7.10	Summary and conclusions	212
	Acknowledgments	213
	References	213
<b>8</b>	<b>The physiology of the Grid</b>	<b>217</b>
	<i>Ian Foster, Carl Kesselman, Jeffrey M. Nick, and Steven Tuecke</i>	
8.1	Introduction	217
8.2	The need for Grid technologies	220
	8.2.1 The evolution of enterprise computing	220
	8.2.2 Service providers and business-to-business computing	222
8.3	Background	222
	8.3.1 The Globus Toolkit	222
	8.3.2 Web services	224
8.4	An Open Grid Services Architecture	225
	8.4.1 Service orientation and virtualization	226
	8.4.2 Service semantics: The Grid service	228
	8.4.3 The role of hosting environments	231
	8.4.4 Using OGSA mechanisms to build VO structures	232
8.5	Application example	234
8.6	Technical details	235
	8.6.1 The OGSA service model	235
	8.6.2 Creating transient services: Factories	237
	8.6.3 Service lifetime management	237
	8.6.4 Managing handles and references	239
	8.6.5 Service data and service discovery	240

8.6.6	Notification	241
8.6.7	Change management	242
8.6.8	Other interfaces	242
8.7	Network protocol bindings	242
8.8	Higher-level services	243
8.9	Related work	244
8.10	Summary	245
	Acknowledgments	246
	References	246
<b>9</b>	<b>Grid Web services and application factories</b>	<b>251</b>
	<i>Dennis Gannon, Rachana Ananthakrishnan, Sriram Krishnan, Madhusudhan Govindaraju, Lavanya Ramakrishnan, and Aleksander Slominski</i>	
9.1	Introduction	251
9.1.1	An overview of the application factory service	253
9.2	XCAT and Web services	254
9.2.1	The OGSA Grid services port and standard CCA ports	259
9.3	The application factory service	261
9.4	Conclusions	263
	References	263
<b>10</b>	<b>From Legion to Avaki: the persistence of vision</b>	<b>265</b>
	<i>Andrew S. Grimshaw, Anand Natrajan, Marty A. Humphrey, Michael J. Lewis, Anh Nguyen-Tuong, John F. Karpovich, Mark M. Morgan, and Adam J. Ferrari</i>	
10.1	Grids are here	265
10.2	Grid architecture requirements	266
10.3	Legion principles and philosophy	269
10.4	Using Legion in day-to-day operations	271
10.4.1	Creating and administering a Legion Grid	272
10.4.2	Legion Data Grid	274
10.4.3	Distributed processing	277
10.4.4	Security	278
10.4.5	Automatic failure detection and recovery	279
10.5	The Legion Grid architecture: under the covers	279
10.5.1	Naming with context paths, LOIDs and object addresses	280
10.5.2	Metadata	282
10.5.3	Security	282
10.6	Core Legion objects	285
10.6.1	Class objects	285
10.6.2	Hosts	286
10.6.3	Vaults	287
10.6.4	Implementation objects	287
10.6.5	Implementation caches	288

10.7	The transformation from Legion to Avaki	288
10.7.1	Avaki today	289
10.7.2	How are Grid requirements relevant to a commercial product?	290
10.7.3	What is retained, removed, reinforced?	291
10.8	Meeting the Grid requirements with Legion	292
10.9	Emerging standards	295
10.10	Summary	297
	References	296
<b>11</b>	<b>Condor and the Grid</b>	<b>299</b>
	<i>Douglas Thain, Todd Tannenbaum, and Miron Livny</i>	
11.1	Introduction	299
11.2	The philosophy of flexibility	301
11.3	The Condor project today	302
11.3.1	The Condor software: Condor and Condor-G	304
11.4	A history of computing communities	307
11.5	Planning and scheduling	315
11.5.1	Combinations of planning and scheduling	317
11.5.2	Matchmaking in practice	318
11.6	Problem solvers	319
11.6.1	Master-Worker	319
11.6.2	Directed Acyclic Graph Manager	321
11.7	Split execution	322
11.7.1	The standard universe	323
11.7.2	The Java universe	325
11.8	Case studies	327
11.8.1	Micron Technology, Inc.	327
11.8.2	C.O.R.E. Digital Pictures	328
11.8.3	NUG30 Optimization Problem	328
11.9	Conclusion	331
	Acknowledgments	331
	References	332
<b>12</b>	<b>Architecture of a commercial enterprise desktop Grid: the Entropia system</b>	<b>337</b>
	<i>Andrew A. Chien</i>	
12.1	Introduction	337
12.2	Background	339
12.3	Requirements for distributed computing	340
12.4	Entropia system architecture	341
12.5	Layered architecture	342
12.6	Programming desktop Grid applications	344
12.7	Entropia usage scenarios	345

12.8	Applications and performance	345
12.9	Summary and futures	346
	Acknowledgments	347
	References	348
<b>13</b>	<b>Autonomic computing and Grid</b>	<b>351</b>
	<i>Pratap Pattnaik, Kattamuri Ekanadham, and Joejon Jann</i>	
13.1	Introduction	351
13.2	Autonomic server components	353
13.3	Approximation with imperfect knowledge	355
	13.3.1 Self-observation	356
	13.3.2 Collective observation	357
13.4	Grid computing	359
	13.4.1 Synergy between the two approaches	359
13.5	Concluding remarks	360
	References	361
<b>14</b>	<b>Databases and the Grid</b>	<b>363</b>
	<i>Paul Watson</i>	
14.1	Introduction	363
14.2	Terminology	365
14.3	The range of uses of databases on the Grid	366
14.4	The database requirements of Grid applications	367
14.5	The Grid and databases: the current state	372
14.6	Integrating databases into the Grid	374
14.7	Federating database systems across the Grid	377
14.8	Conclusions	381
	Acknowledgments	382
	References	383
<b>15</b>	<b>The Open Grid Services Architecture, and Data Grids</b>	<b>385</b>
	<i>Peter Z. Kunszt and Leanne P. Guy</i>	
15.1	Introduction	385
	15.1.1 The vision	386
	15.1.2 Desirable features of Grids	386
	15.1.3 Virtual organizations	387
	15.1.4 Motivation for the desirable features in Data Grids	387
15.2	The OGSA approach	388
15.3	Data Grid services	390
	15.3.1 The data	391
	15.3.2 The functionality and the services	393
	15.3.3 Data Grid and OGSA	398
15.4	Issues	401
	15.4.1 Availability and robustness	402
	15.4.2 Scalability	402

15.4.3	Monitorability	402
15.4.4	Integration	403
15.4.5	Security	403
15.4.6	Interoperability and compatibility	403
15.4.7	Service discovery	404
15.4.8	Manageability	404
15.5	Summary	404
	Acknowledgments	405
	References	405
<b>16</b>	<b>Virtualization services for Data Grids</b>	<b>409</b>
	<i>Reagan W. Moore and Chaitan Baru</i>	
16.1	Introduction	409
16.2	Digital entities	410
	16.2.1 Long-term persistence	411
16.3	Data, information, and knowledge	412
	16.3.1 A unifying abstraction	412
	16.3.2 Virtualization and levels of data abstraction	416
	16.3.3 Data Grid infrastructure	421
	16.3.4 Data Grid projects	422
16.4	Information integration	424
	16.4.1 Data warehousing	427
	16.4.2 Database and application integration	428
	16.4.3 Semantic data integration	431
	16.4.4 Model-based integration	432
16.5	Summary	432
	Acknowledgments	433
	References	433
<b>17</b>	<b>The Semantic Grid: a future e-Science infrastructure</b>	<b>437</b>
	<i>David De Roure, Nicholas R. Jennings, and Nigel R. Shadbolt</i>	
17.1	Introduction	437
17.2	A semantic Grid scenario	440
17.3	A service-oriented view	442
	17.3.1 Justification of a service-oriented view	443
	17.3.2 Key technical challenges	448
	17.3.3 A service-oriented view of the scenario	451
17.4	The knowledge layer	455
	17.4.1 The knowledge life cycle	455
	17.4.2 Ontologies and the knowledge layer	460
	17.4.3 Knowledge layer aspects of the scenario	462
	17.4.4 Research issues	465
17.5	Conclusions	465
	References	467

<b>18</b>	<b>Peer-to-peer Grids</b>	<b>471</b>
	<i>Geoffrey Fox, Dennis Gannon, Sung-Hoon Ko, Sangmi-Lee, Shrideep Pallickara, Marlon Pierce, Xiaohong Qiu, Xi Rao, Ahmet Uyar, Minjun Wang, and Wenjun Wu</i>	
18.1	Peer-to-peer Grids	471
18.2	Key technology concepts for P2P Grids	472
18.3	Peer-to-peer Grid event service	477
18.4	Collaboration in P2P Grids	481
18.5	User Interfaces and universal access	487
	Acknowledgments	489
	References	489
<b>19</b>	<b>Peer-to-peer Grid databases for Web service discovery</b>	<b>491</b>
	<i>Wolfgang Hoschek</i>	
19.1	Introduction	491
19.2	A database for discovery of distributed content	498
	19.2.1 Content link and content provider	498
	19.2.2 Publication	500
	19.2.3 Query	502
	19.2.4 Caching	504
	19.2.5 Soft state	505
	19.2.6 Flexible freshness	506
19.3	Web service discovery architecture	507
	19.3.1 Interfaces	508
	19.3.2 Network protocol bindings and services	512
	19.3.3 Properties	513
19.4	Peer-to-peer Grid databases	514
	19.4.1 Routed versus direct response, metadata responses	515
	19.4.2 Query processing	518
	19.4.3 Static loop time-out and dynamic abort time-out	522
	19.4.4 Query scope	525
19.5	Peer database protocol	527
19.6	Related work	530
19.7	Conclusions	534
	Acknowledgments	536
	References	536
<b>PART C</b>	<b>Grid computing environments</b>	<b>541</b>
<b>20</b>	<b>Overview of Grid computing environments</b>	<b>543</b>
	<i>Geoffrey Fox, Dennis Gannon, and Mary Thomas</i>	
20.1	Introduction	543

20.2	Overall classification	544
20.3	Summary of GCE projects and features	546
20.3.1	Technology for building GCE systems	546
20.3.2	Largely problem-solving environments	547
20.3.3	Largely basic GCEShell portals	547
20.3.4	Security	548
20.3.5	Workflow	548
20.3.6	Data management	548
20.3.7	GCEShell tools	549
20.3.8	GCE computing model	549
	References	551
<b>21</b>	<b>Grid programming models: current tools, issues and directions</b>	<b>555</b>
	<i>Craig Lee and Domenico Talia</i>	
21.1	Introduction	555
21.2	Grid programming issues	556
21.2.1	Portability, interoperability, and adaptivity	557
21.2.2	Discovery	557
21.2.3	Performance	557
21.2.4	Fault tolerance	558
21.2.5	Security	558
21.2.6	Program metamodels	558
21.3	A brief survey of Grid programming tools	558
21.3.1	Shared-state models	559
21.3.2	Message-passing models	560
21.3.3	RPC and RMI models	561
21.3.4	Hybrid models	563
21.3.5	Peer-to-peer models	564
21.3.6	Frameworks, component models, and portals	565
21.3.7	Web service models	567
21.3.8	Coordination models	568
21.4	Advanced programming support	568
21.4.1	Traditional techniques	568
21.4.2	Data-driven techniques	569
21.4.3	Speculative or optimistic techniques	569
21.4.4	Distributed techniques	570
21.4.5	Grid-aware I/O	570
21.4.6	Advanced communication services	570
21.4.7	Security	572
21.4.8	Fault tolerance	572
21.4.9	Program metamodels and Grid-aware run-time systems	573
21.5	Conclusion	574
	References	575



<b>22</b>	<b>NaradaBrokering: an event-based infrastructure for building scalable durable peer-to-peer Grids</b>	<b>579</b>
	<i>Geoffrey Fox and Shrideep Pallickara</i>	
22.1	Introduction	579
22.2	NaradaBrokering	581
	22.2.1 Broker organization and small worlds behavior	582
	22.2.2 Dissemination of events	583
	22.2.3 Failures and recovery	584
	22.2.4 Support for dynamic topologies	584
	22.2.5 Results from the prototype	585
22.3	JMS compliance in NaradaBrokering	586
	22.3.1 Rationale for JMS compliance in NaradaBrokering	586
	22.3.2 Supporting JMS interactions	587
	22.3.3 The distributed JMS solution	588
	22.3.4 JMS performance data	589
22.4	Naradabrokering and P2P interactions	592
	22.4.1 JXTA	593
22.5	JXTA integration in NaradaBrokering	594
	22.5.1 The interaction model	595
	22.5.2 Interaction disseminations	595
	22.5.3 JXTA applications and NaradaBrokering	597
	22.5.4 NaradaBrokering-JXTA systems	597
22.6	Conclusion	598
	References	599
<b>23</b>	<b>Classifying and enabling Grid applications</b>	<b>601</b>
	<i>Gabrielle Allen, Tom Goodale, Michael Russell, Edward Seidel, and John Shalf</i>	
23.1	A new challenge for application developers	601
23.2	Applications must be the lifeblood!	602
23.3	Case study: real-world examples with the cactus computational toolkit	604
23.4	Stepping back: a simple motivation-based taxonomy for Grid applications	605
	23.4.1 Generic types of Grid applications	605
	23.4.2 Grid operations for applications	606
23.5	The way forward: Grid programming environments and their effective use	610
23.6	In conclusion	612
	Acknowledgments	613
	References	613
<b>24</b>	<b>NetSolve: past, present, and future – a look at a Grid enabled server</b>	<b>615</b>
	<i>Sudesh Agrawal, Jack Dongarra, Keith Seymour, and Sathish Vadhiyar</i>	
24.1	Introduction	615

24.2	How NetSolve works today	616
24.3	NetSolve in scientific applications	619
24.3.1	IPARS (integrated parallel accurate reservoir simulators)	619
24.3.2	MCell	619
24.3.3	SARA3D	620
24.3.4	Evolutionary farming	620
24.3.5	LSI-based conference organizer	620
24.4	Future work	621
24.4.1	Network address translators	621
24.4.2	Resource selection criteria	622
24.5	Conclusions	623
	References	623
<b>25</b>	<b>Ninf-G: a GridRPC system on the Globus toolkit</b>	<b>625</b>
	<i>Hidemoto Nakada, Yoshio Tanaka, Satoshi Matsuoka, and Satoshi Sekiguchi</i>	
25.1	Introduction	625
25.2	Globus toolkit	627
25.3	Design of Ninf-G	628
25.3.1	GridRPC system	628
25.3.2	Ninf-G implementation on the Globus toolkit	628
25.3.3	API of Ninf-G	628
25.3.4	Server side IDL	629
25.4	Ninf-G implementation	630
25.4.1	'Gridifying' a library or an application using GridRPC	630
25.4.2	Performing GridRPC	632
25.5	Usage scenario	633
25.5.1	Setup remote executable	633
25.5.2	Client program	633
25.6	Preliminary evaluation	635
25.6.1	Experimental setup	635
25.6.2	Results and discussion	635
25.7	Conclusion	636
	References	637
<b>26</b>	<b>Commodity Grid kits – middleware for building Grid computing environments</b>	<b>639</b>
	<i>Gregor von Laszewski, Jarek Gawor, Sriram Krishnan, and Keith Jackson</i>	
26.1	Introduction	639
26.2	Grid computing environments and portals	640
26.3	Commodity technologies	643
26.4	Overview of the Java CoG kit	645
26.5	Current work	647
26.5.1	InfoGram	647
26.5.2	Web services	647

26.6	Advanced CoG kit components	648
	26.6.1 Sample components	649
	26.6.2 Community use	650
26.7	Conclusion	652
26.8	Availability	654
	Acknowledgments	654
	References	654
<b>27</b>	<b>The Grid portal development kit</b>	<b>657</b>
	<i>Jason Novotny</i>	
27.1	Introduction	657
27.2	Overview of the Grid portal development kit	659
27.3	Grid portal architecture	660
27.4	GPDK implementation	661
27.5	GPDK services	663
	27.5.1 Security	664
	27.5.2 Job submission	665
	27.5.3 File transfer	666
	27.5.4 Information services	666
	27.5.5 GPDK user profiles	667
27.6	GPDK as a portal development environment	667
	27.6.1 Application portals based on GPDK	670
27.7	Related work	670
27.8	Conclusion and future work	671
	Acknowledgments	672
	References	672
<b>28</b>	<b>Building Grid computing portals: the NPACI Grid portal toolkit</b>	<b>675</b>
	<i>Mary P. Thomas and John R. Boisseau</i>	
28.1	Introduction	675
	28.1.1 What are Grid computing portals?	675
	28.1.2 History and motivation	676
	28.1.3 Grid portal users and developers	679
28.2	The Grid portal toolkit (GridPort)	680
	28.2.1 GridPort architecture	680
	28.2.2 GridPort capabilities	682
	28.2.3 GridPort security	683
28.3	GridPort portals	684
	28.3.1 HotPage user portal	686
	28.3.2 Laboratory for Applied Pharmacokinetics modeling portal	687
	28.3.3 Telescience portal	689
	28.3.4 NBCR computational portal environment	690
28.4	Summary of lessons learned	691
28.5	Current and future work	693
	28.5.1 GridPort GCE architecture	694

28.5.2	GridPort Grid Web services	695
28.5.3	Grid portals and applications	697
	Acknowledgments	698
	References	698
<b>29</b>	<b>Unicore and the Open Grid Services Architecture</b>	<b>701</b>
	<i>David Snelling</i>	
29.1	Introduction	701
29.1.1	Caveats	702
29.2	Implementation	702
29.2.1	Infrastructure and architecture	702
29.2.2	Supported Grid Services	705
29.3	Lessons	707
29.3.1	Are GridService extensions to WSDL really needed?	707
29.3.2	Can a single GridService support multiple portTypes?	710
29.3.3	Is the 'Push Only' notification framework sufficient?	710
29.3.4	Is security provision at binding level only adequate?	711
29.4	Conclusions and future directions	711
	References	712
<b>30</b>	<b>Distributed object-based Grid computing environments</b>	<b>713</b>
	<i>Tomasz Haupt and Marlon E. Pierce</i>	
30.1	Introduction	713
30.2	Deployment and use of computing portals	714
30.2.1	DMEFS: an application of the Mississippi Computational Web Portal	715
30.2.2	Gateway support for commodity codes	715
30.3	Computing portal services	716
30.4	Grid portal architecture	717
30.4.1	The user interface	717
30.4.2	Component-based middleware	718
30.4.3	Resource tier	722
30.5	Application descriptors	722
30.6	Sample service implementations	723
30.6.1	Batch script generation	723
30.6.2	Context management	723
30.7	Kerberos security requirements in multitiered architectures	724
30.8	Summary and future work	726
	References	727
<b>31</b>	<b>DISCOVER: a computational collaboratory for interactive Grid applications</b>	<b>729</b>
	<i>Vijay Mann and Manish Parashar</i>	
31.1	Introduction	729

31.2	The DISCOVER middleware substrate for Grid-based collaboratories	731
31.2.1	DISCOVER middleware substrate design	732
31.2.2	DISCOVER middleware substrate implementation	734
31.2.3	DISCOVER middleware operation	735
31.2.4	DISCOVER middleware substrate experimental evaluation	737
31.3	DIOS: Distributed interactive object substrate	738
31.3.1	Sensors, actuators and interaction objects	739
31.3.2	Local, global and distributed objects	740
31.3.3	DIOS control network and interaction agents	740
31.3.4	Experimental evaluation	742
31.4	The collaborative interaction and steering portal	743
31.5	Summary and current status	744
	Acknowledgments	745
	References	745
<b>32</b>	<b>Grid resource allocation and control using computational economies</b>	<b>747</b>
	<i>Rich Wolski, John Brevik, James S. Plank, and Todd Bryan</i>	
32.1	Introduction	747
32.2	Computational economies and the Grid	749
32.2.1	Price-setting mechanisms: commodities markets and auctions	751
32.2.2	Pricing functions	752
32.3	An example study: G-commerce	755
32.3.1	Producers and consumers	756
32.3.2	Commodities markets	759
32.3.3	Auctions	761
32.4	Simulations and results	762
32.4.1	Market conditions, under-demand case	763
32.4.2	Efficiency	765
32.5	Discussion	767
32.6	Conclusions	768
	Acknowledgments	769
	References	769
<b>33</b>	<b>Parameter sweeps on the Grid with APST</b>	<b>773</b>
	<i>Henri Casanova and Fran Berman</i>	
33.1	Introduction	773
33.2	The APST project	774
33.2.1	Motivation	774
33.2.2	Principles and architecture	775
33.2.3	Software architecture	776
33.2.4	Scheduling	777
33.2.5	Implementation	778

33.3	APST: usage and applications	780
33.3.1	APST and Grid resources	780
33.3.2	Running applications with APST	781
33.3.3	Discussion	782
33.4	Related work	783
33.5	Conclusion and future developments	784
	References	784
<b>34</b>	<b>Storage manager and file transfer Web services</b>	<b>789</b>
	<i>William A. Watson III, Ying Chen, Jie Chen, and Walt Akers</i>	
34.1	Introduction	789
34.1.1	Motivation	790
34.1.2	Architecture	790
34.2	Data Grid Web services	791
34.2.1	Data Grids	791
34.2.2	File/dataset catalogs	792
34.2.3	Storage resource management Web service – JSRM	793
34.2.4	Reliable file transfer	795
34.2.5	Security	797
34.3	Grid file Manager	797
34.3.1	Grid file interface – a Java API	797
34.3.2	Grid File Manager	798
34.4	The lattice portal	800
34.5	Lessons learned and future plans	800
	References	801
<b>PART D</b>	<b>Grid applications</b>	<b>803</b>
<b>35</b>	<b>Application overview for the book: Grid computing – making the global infrastructure a reality</b>	<b>805</b>
	<i>Fran Berman, Geoffrey Fox, and Tony Hey</i>	
35.1	Introduction	805
35.2	Grid applications	806
	References	808
<b>36</b>	<b>The data deluge: an e-Science perspective</b>	<b>809</b>
	<i>Tony Hey and Anne Trefethen</i>	
36.1	Introduction	809
36.2	The imminent scientific data deluge	811
36.2.1	Introduction	811
36.2.2	Normalization	812
36.2.3	Astronomy	813
36.2.4	Bioinformatics	813
36.2.5	Environmental science	814

36.2.6	Particle physics	815
36.2.7	Medicine and health	815
36.2.8	Social sciences	815
36.3	Scientific metadata, information and knowledge	816
36.4	Data Grids and digital libraries	818
36.5	Open archives and scholarly publishing	819
36.6	Digital preservation and data curation	820
36.7	Concluding remarks	821
	Acknowledgments	822
	References	822
<b>37</b>	<b>Metacomputing</b>	<b>825</b>
	<i>Larry Smarr and Charles E. Catlett</i>	
37.1	LAN metacomputer at NCSA	827
37.2	Metacomputing at Siggraph'92 showcase'92	829
37.2.1	Theoretical simulation	829
37.2.2	Molecular virtual reality	831
37.2.3	User-executed simulation/analysis of severe thunderstorm phenomena	831
37.2.4	Instrument/sensor control	832
37.2.5	Interactive imaging of atomic surfaces	832
37.2.6	Data navigation	832
37.3	Interactive four-dimensional imaging	832
37.4	Scientific multimedia digital library	833
37.5	Navigating simulated and observed cosmological structures	834
	Acknowledgments	835
	Further reading	835
<b>38</b>	<b>Grids and the virtual observatory</b>	<b>837</b>
	<i>Roy Williams</i>	
38.1	The virtual observatory	837
38.1.1	Data federation	838
38.2	What is a Grid?	839
38.2.1	Virtual Observatory middleware	840
38.3	Image computing	840
38.3.1	Virtual Sky: multiwavelength imaging	840
38.3.2	MONTAGE: on-demand mosaics	844
38.3.3	Science with federated images	844
38.3.4	MONTAGE architecture	845
38.3.5	Quest: multitemporal imaging	847
38.3.6	A galaxy morphology study	849
38.4	Database computing	850
38.4.1	The VOTable XML proposal	851
38.4.2	Database mining and visualization	852

38.5	A semantic Web of astronomical data	853
38.5.1	The Strasbourg ontology	854
38.5.2	I want my own ontology	856
38.5.3	Ontology mapping	856
38.6	Conclusion	857
	References	857
<b>39</b>	<b>Data-intensive Grids for high-energy physics</b>	<b>859</b>
	<i>Julian J. Bunn and Harvey B. Newman</i>	
39.1	Introduction: scientific exploration at the high-energy frontier	859
39.2	HEP challenges: at the frontiers of information technology	861
39.3	Meeting the challenges: Data Grids as managed distributed systems for global virtual organizations	861
39.4	Emergence of HEP Grids: regional centers and global databases	863
39.4.1	The CMS computing model circa 1996	863
39.4.2	GIOD	864
39.4.3	MONARC	865
39.4.4	ALDAP	866
39.5	HEP Grid projects	867
39.5.1	PPDG	867
39.5.2	GriPhyN	869
39.5.3	iVDGL	870
39.5.4	DataGrid	870
39.5.5	LCG	871
39.5.6	CrossGrid	873
39.6	Example architectures and applications	874
39.6.1	TeraGrid prototype	874
39.6.2	MOP for Grid-enabled simulation production	875
39.6.3	GRAPPA	876
39.6.4	SAM	878
39.7	Inter-Grid coordination	879
39.7.1	HICB	879
39.7.2	GLUE	880
39.7.3	DataTAG	880
39.7.4	Global Grid Forum	881
39.8	Current issues for HEP Grids	882
39.8.1	HEP Grids versus classical Grids	882
39.8.2	Grid system architecture: above (or within) the collective layer	883
39.8.3	Grid system software design and development requirements	885
39.8.4	HEP Grids and networks	886
39.8.5	Strategic resource planning: the key role of modeling and simulation	887



39.9	A distributed server architecture for dynamic HEP Grid services	888
39.9.1	The station server framework	889
39.9.2	Key distributed system features of the JINI prototype	890
39.9.3	Station server operation	890
39.9.4	Possible application to a scalable job scheduling service	891
39.9.5	An agent-based monitoring system using the DDSA	892
39.10	The Grid-enabled analysis environment	897
39.10.1	Requirements: analysis versus production	898
39.10.2	Access to object collections	898
39.10.3	Components of the GAE	899
39.10.4	Clarens	899
39.10.5	Caige	900
39.11	Conclusion: relevance of meeting these challenges for future networks and society	902
	Acknowledgments	903
	References	903
<b>40</b>	<b>The new biology and the Grid</b>	<b>907</b>
	<i>Kim Baldrige and Philip E. Bourne</i>	
40.1	Introduction	907
40.1.1	The first wave: compute-driven biology applications	907
40.1.2	The next wave: data-driven applications	908
40.2	Bioinformatics Grid applications today	910
40.2.1	Example 1: CEPAR and CEPort – 3D protein structure comparison	910
40.2.2	Example 2: Chemport – a quantum mechanical biomedical framework	914
40.3	The challenges of the Grid from the perspective of bioinformatics researchers	918
40.3.1	A future data-driven application – the encyclopedia of life	919
40.4	Summary	921
	Acknowledgments	922
	References	922
<b>41</b>	<b>eDiamond: a Grid-enabled federated database of annotated mammograms</b>	<b>923</b>
	<i>Michael Brady, David Gavaghan, Andrew Simpson, Miguel Mulet Parada, and Ralph Highnam</i>	
41.1	Introduction	923
41.2	Medical images	925
41.3	Mammography	928
41.3.1	Breast cancer facts	928
41.3.2	Mammographic images and standard mammogram form (SMF)	931

41.4	Medical databases	933
41.5	eDiamond	936
41.5.1	Introduction	936
41.5.2	e-Science challenges	937
41.5.3	Objectives	937
41.5.4	Project structure	938
41.6	Related projects	940
41.7	Conclusions	942
	Acknowledgments	942
	References	943
<b>42</b>	<b>Combinatorial chemistry and the Grid</b>	<b>945</b>
	<i>Jeremy G. Frey, Mark Bradley, Jonathan W. Essex, Michael B. Hursthouse, Susan M. Lewis, Michael M. Luck, Luc Moreau, Dave C. De Roure, Mike Surridge, and Alan H. Welsh</i>	
42.1	Introduction	945
42.2	What is combinatorial chemistry?	946
42.3	'Split & mix' approach to combinatorial chemistry	946
42.4	Chemical markup language (cML)	950
42.5	Statistics & design of experiments	952
42.6	Statistical models	953
42.7	The multimedia nature of chemistry information	953
42.8	The pervasive Grid and metadata	955
42.9	Virtual data	956
42.10	Multimedia collaboration	957
42.11	A Grid or intra-Grids	957
42.12	e-Science and e-Business	959
42.13	Conclusions	960
	Appendix 1: The Comb-e-Chem e-Science pilot project	960
	References	961
<b>43</b>	<b>Education and the enterprise with the Grid</b>	<b>963</b>
	<i>Geoffrey Fox</i>	
43.1	Introduction	963
43.2	Web service paradigm for the enterprise	964
43.3	Implementing Web services	969
43.4	Education as a Web service	970
	References	975
<b>Index</b>		<b>977</b>
	Views of the Grid	1002
	Indirect Glossary	1004
	List of Grid Projects	1007

# Sample Illustration

StateSpec ::= InternalState  $\Psi$ , EstimatedExternalState  $\bar{\xi}$   
 MethodSpec ::= MethodSet  $\Pi$ , each  $\pi \in \Pi : \Sigma \times \Psi \times \bar{\xi} \rightarrow \Phi \times \Psi \times \bar{\xi}$   
 StrategySpec ::= Efficiency  $\eta$ , Strategy  $\alpha : \Sigma \times \Psi \times \bar{\xi} \rightarrow \Pi$

The functional behavior of C is captured by a relation,  $\beta \subseteq \Sigma \times \Phi$ , where  $\Sigma$  is the input alphabet,  $\Phi$  is the output alphabet and  $\beta$  is a relation specifying valid input–output pair. Thus, if C receives an input  $u \in \Sigma$ , it delivers an output  $v \in \Phi$ , satisfying the relation  $\beta(u, v)$ . The output variability permitted by the relation  $\beta$  (as opposed to a function) is very common to most systems. As illustrated in Figure 13.1, a client is satisfied to get any one of the many possible outputs  $(v, v', \dots)$  for a given input  $u$ , as long as they satisfy some property specified by  $\beta$ . All implementations of the component preserve this functional behavior.

The state information maintained by a component comprises two parts: internal state  $\Psi$  and external state  $\bar{\xi}$ . Internal state,  $\Psi$ , contains the data structures used by an implementation and any other variables used to keep track of input–output history and resource utilization. The external state  $\xi$  is an abstraction of the environment of C and includes information on the input arrival process, the current level of resources available for C and the performance levels of other components of the system whose services are invoked by C. The component C has no control over the variability in the ingredients of  $\xi$ , as they are governed by agents outside C. The input arrival process is clearly outside C. We assume an external global resource manager that may supply or withdraw resources from C dynamically. Finally, the component C has no control over how other components are performing and must expect arbitrary variations (including failure) in their health. Thus the state information,  $\xi$ , is dynamically changing and is distributed throughout the system.

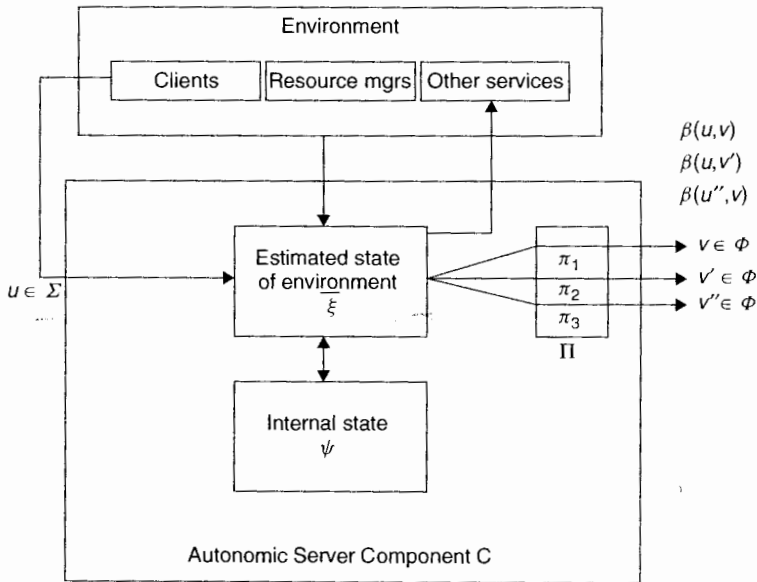


Figure 13.1 Schematic view of an autonomous component and its environment.