
Contents

List of Figures	xv
List of Tables	xix
Preface	xxi
Presentation of the Book	xxv
I The Birth	1
1 Volunteer Computing and BOINC	3
<i>David Anderson</i>	
1.1 Background	3
1.1.1 Volunteer Computing	4
1.1.2 BOINC: Middleware for Volunteer Computing	5
1.2 Volunteer Features	6
1.2.1 Projects, Clients, and Attachment	6
1.2.2 Account Managers	9
1.2.3 Project Website Features	10
1.2.4 Preferences	10
1.2.5 Teams	11
1.2.6 Credit	11
1.2.7 Volunteer Support	12
1.3 Data Model	12
1.4 Computing Model	14
1.4.1 Applications and Versions	14
1.4.2 Jobs	15
1.4.3 Result Validation	15
1.4.4 Assigned Jobs	16
1.4.5 Long-Running Jobs	16
1.4.6 Non-Compute-Intensive Applications	16
1.5 Applications	17
1.5.1 BOINC-Aware Applications	17
1.5.2 The BOINC Wrapper	18
1.5.3 The BOINC Virtualbox Wrapper	18
1.6 Communication Structure	19

1.7	Server Software	20
1.8	Client Software	22
1.9	Research Tools and Directions	24
	Bibliography	25
2	Open, Scalable and Self-Regulated Federations of Desktop Grids with OurGrid	29
	<i>Francisco Brasileiro and Nazareno Andrade</i>	
2.1	Case for a Desktop Grid for and by Small and Medium-Sized Laboratories	29
2.2	Bird's Eye View of OurGrid	32
2.3	Architecting Cooperation	33
	2.3.1 Network of Favors	34
	2.3.2 Accounting for Computation	36
	2.3.3 Multiple-Service Peer-to-Peer Grids	36
2.4	Worker Perspective	37
2.5	Managing the Resources of Each Site	38
2.6	Scheduling Applications on the Grid	40
	2.6.1 Scheduling with No Information	40
	2.6.1.1 Worqueue with Replication	41
	2.6.1.2 System-Wide Impact of Replication	42
	2.6.2 Other OurGrid Schedulers	43
	2.6.2.1 Data-Aware Schedulers	43
	2.6.2.2 Scheduling with Partial Information	44
2.7	Extending an OurGrid System with Public Cloud Resources	44
2.8	Open Challenges and Future Directions	46
	Bibliography	47
3	The XtremWebCH Volunteer Computing Platform	53
	<i>Nabil Abdennadher, Marko Niinimäki, and Mohamed BenBelgacem</i>	
3.1	Introduction	53
3.2	XWCH Architecture and Functionality	55
	3.2.1 Architecture	55
	3.2.1.1 Communication Protocol	56
	3.2.1.2 Dynamic Job Generation	57
	3.2.1.3 Data Replication	57
	3.2.1.4 Controlling and Monitoring of Jobs	57
	3.2.2 XWCH Programming Model	58
3.3	XWCH High-Level API	62
3.4	Applications	65
	3.4.1 NeuroWeb	65
	3.4.2 GIFT	68
	3.4.3 MetaPIGA	69
3.5	Integration with the ARC Grid Middleware	71
3.6	Conclusion	74

Bibliography	75
4 XtremWeb-HEP: Designing Desktop Grid for the EGEE Infrastructure	79
<i>Oleg Lodygensky, Etienne Urbah, and Simon Dadoun</i>	
4.1 Introduction	80
4.2 Architecture	80
4.3 Security Considerations	82
4.3.1 Secured Communications	82
4.3.2 Authentication of Server, Client, and Worker	82
4.3.3 Authorizations	83
4.3.4 Ownership	84
4.3.5 Access Rights	84
4.3.6 Confinements	85
4.3.6.1 Public Confinement	86
4.3.6.2 Group Confinement	88
4.3.6.3 Private Confinement	89
4.3.7 Data Confidentiality	90
4.3.8 Data Integrity	91
4.3.9 Data Persistence	91
4.3.10 Result Certification	91
4.4 Bridging XWHEP to Other Grids	91
4.4.1 Xtremweb-HEP Plugin of the 3G Bridge	92
4.4.2 XWHEP Bridge to gLite	93
4.4.2.1 Bridging Security for XWHEP Job Submission to gLite	93
4.4.2.2 Architecture of the XWHEP Bridge to gLite	94
4.4.2.3 Usage of SG Resources by the XWHEP Bridge	95
4.5 Acknowledgments	96
Bibliography	96
5 A Volunteer Computing Platform Experience for Neuromuscular Disease Problems	99
<i>Nicolas Bard, Viktors Bertis, Raphaël Bolze, and Frédéric Desprez</i>	
5.1 Introduction	100
5.2 Help Cure Muscular Dystrophy Project (HCMD)	101
5.2.1 MAXDo Program	102
5.3 World Community Grid: A Volunteer Grid	103
5.3.1 Desktop Grid Description	104
5.3.2 Needs and Requirements	105
5.4 Workunits Preparation	106
5.4.1 Phase 1	106
5.4.1.1 Analysis of MAXDo Program Behavior	106
5.4.1.2 Workunit Packaging	109
5.4.2 Phase 2	110

5.4.2.1	Analysis of MAXDo Program Behavior . . .	110
5.4.2.2	Workunit Packaging	110
5.4.3	Porting to World Community Grid	111
5.5	HCMD Project Launched on World Community Grid	112
5.5.1	Phase 1	112
5.5.1.1	Computing Phases	112
5.5.1.2	Result Processing and Verification	114
5.5.2	Phase 2	114
5.5.2.1	Computing Phases	114
5.5.2.2	Result Processing and Verification	115
5.6	Comparative Performance with a Dedicated Grid Phase I . .	117
5.7	Conclusion	119
	Acknowledgments	120
	Bibliography	120
6	How to Work with XtremWeb, Condor, BOINC on Top of BonjourGrid	123
	<i>Christophe Cérin, Heithem Abbes, and Walid Saad</i>	
6.1	Introduction	124
6.2	BonjourGrid System	124
6.2.1	Overview of BonjourGrid System	125
6.2.2	Service-Oriented Architecture for Building a Computing Element	127
6.2.2.1	From Idle to Coordinator State	127
6.2.2.2	From Idle to Worker State	129
6.2.3	Implementation	130
6.3	Installing BonjourGrid	131
6.3.1	How to Create and Set Up Your Virtual Desktop Grid Based on BonjourGrid	131
6.3.1.1	Preparing the Environment for the Grid . . .	133
6.3.1.2	Configure a Virtual Machine for the Grid . .	136
6.3.1.3	Secure Communication within the Grid . . .	136
6.3.1.4	Configure the Supervisor Machine (Home Machine, Front Machine)	137
6.3.1.5	Configuring MySQL	138
6.3.2	Installing BonjourGrid from Sources	138
6.3.2.1	Installation and Configuration Package mDNSResponder-107.5	138
6.3.2.2	Installation of Bonjour-py-0.1 (Programming Interface for Python)	139
6.3.2.3	Installation of BOINC	139
6.3.2.4	Installation of Condor	140
6.3.2.5	Installation of XtremWeb	141
6.4	Deploying an Application with BonjourGrid Middleware . .	142

6.4.1	Deploying an Application with BOINC as the Desktop Grid Middleware	142
6.4.1.1	Gridifying Application	142
6.4.1.2	Deploying Application	143
6.4.2	Deploying an Application with Condor as the Desktop Grid Middleware	145
6.4.2.1	Gridifying Application	145
6.4.2.2	Deploying Application	145
6.4.3	Deploying an Application with XtremWebCH as the Desktop Grid Middleware	147
6.4.3.1	Gridifying Application	147
6.4.3.2	Deploying Application	147
6.5	Glossary	149
	Bibliography	149
7	How to Work with PastryGrid	151
	<i>Christophe Cérin and Heithem Abbes</i>	
7.1	Introduction	151
7.2	PastryGrid System	153
7.2.1	How a User Can Describe His Distributed Application	154
7.2.2	Design of PastryGrid	155
7.2.2.1	Addressing Scheme	155
7.2.2.2	Resource Discovery	156
7.2.3	Concept of RDV	157
7.2.3.1	Execution Initialization	157
7.2.3.2	Communication with RDV	157
7.2.4	Coordination and Data Transfer	158
7.3	Installing PastryGrid	159
7.4	Deploying an Application	159
7.5	Glossary	161
	Bibliography	162
II	Maturity and Beyond	165
8	Challenges in Designing Scheduling Policies in Volunteer Computing	167
	<i>Trilce Estrada and Michela Taufer</i>	
8.1	Introduction	167
8.2	Simulation and Emulation Tools to Design and Test VC Scheduling Policies	170
8.3	Scheduling Policies	176
8.3.1	Naive and Knowledge-Based Scheduling Policies . . .	176
8.3.2	Adaptive Scheduling Policies	179
8.4	Beyond Scheduling: Building Autonomic VC Environments .	187
	Bibliography	188

9 Modeling and Optimizing Availability of Non-Dedicated Resources	191
<i>Artur Andrzejak and Derrick Kondo</i>	
9.1 Introduction	191
9.1.1 Problem of Individual Failures	192
9.1.2 Chapter Contents	194
9.2 Modeling Availability of Non-Dedicated Resources	194
9.2.1 Availability Distributions	195
9.2.2 Individual Availability Models	197
9.2.2.1 Short-Term Prediction	197
9.2.2.2 Long-Term Modeling and Host Ranking	198
9.3 Collective Availability	199
9.3.1 Statistical Guarantees and Cost versus Migration Rate	200
9.4 Case Studies for SETI@home-Hosts	201
9.4.1 Availability Prediction	201
9.4.2 Availability Guarantees	203
9.4.3 Cost versus Migration Rate	204
9.5 Conclusion	208
Bibliography	208
10 Security and Result Certification	211
<i>Filipe Araujo and Patrício Domingues</i>	
10.1 Introduction	211
10.2 Architectural Model	212
10.3 Threats to Projects	213
10.3.1 Non-Completed Tasks	213
10.3.2 Incorrect Results	214
10.3.3 Tampered Results	215
10.3.3.1 Attacks to the Server-Side Infrastructure	216
10.4 Threats to Desktop Grid Resources	218
10.4.1 Protecting Desktop Grid Resources	219
10.5 Result Certification	220
10.5.1 Majority Voting	220
10.5.2 Spot-Checking	221
10.5.3 Credibility-Based Systems	222
10.6 Collusion of Volunteer Nodes	224
10.6.1 EigenTrust Algorithm	225
10.6.2 EigenTrust for Volunteer Computing	226
10.6.3 Identification of Malicious Results through Correlation	228
10.6.4 Online Identification of Colluders	229
10.7 Conclusion	230
Bibliography	231

11 Data-Intensive Computing on Desktop Grids	237
<i>Heshan Lin, Gilles Fedak, and Wu-Chun Feng</i>	
11.1 Introduction	238
11.2 State-of-the-Art Data Processing on Desktop Grids	239
11.2.1 Storage Harnessing of Desktop Computers	239
11.2.2 Data-Aware Scheduling and Computing	239
11.2.3 Programing Models	240
11.3 MapReduce Overview	241
11.4 MOON: MapReduce on Opportunistic eNvironments	242
11.4.1 Hadoop Background	242
11.4.2 MOON Data Management	243
11.4.2.1 Non-Uniform Replication and I/O Request Distribution	243
11.4.2.2 I/O Load Monitoring	245
11.4.2.3 Adaptive Replication of Opportunistic Files	245
11.4.3 MOON Task Scheduling	246
11.4.3.1 Identifying Stragglers on Opportunistic Environments	247
11.4.3.2 Two-Phase Scheduling	247
11.4.4 Performance Evaluation	248
11.4.4.1 Hybrid Replication of Intermediate Data	248
11.4.4.2 Overall Performance Impacts of MOON	249
11.5 Practical MapReduce Computing on Desktop Grids with BitDew	251
11.5.1 Architecture of MapReduce over BitDew	251
11.5.1.1 Latency Hiding	252
11.5.1.2 Collective File Operation	252
11.5.1.3 Fault Tolerance	253
11.5.1.4 Barrier-Free Computation	253
11.5.1.5 Scheduling	253
11.5.1.6 Distributed Result Checking	254
11.5.2 Performance Evaluation	254
11.6 Conclusion	255
Bibliography	256
 12 Roles of Desktop Grids in Hybrid Distributed Computing Infrastructures	 261
<i>Simon Delamare and Gilles Fedak</i>	
12.1 Hybrid Distributed Computing Infrastructures (H-DCI)	262
12.2 Using Local Desktop Grids as Cache-to-Cloud Resources	265
12.3 Using Cloud Resources to Improve the QoS of Desktop Grids	268
12.3.1 Motivation	269
12.3.2 SpeQuloS Framework Overview	271
12.3.2.1 Monitoring Desktop Grids	272
12.3.2.2 Billing Cloud Usage to Users	273

12.3.2.3	Predicting Completion Time	273
12.3.2.4	Strategies for Cloud Resources Deployment .	273
12.3.2.5	Cloud Workers Management	275
12.3.2.6	Implementation	275
12.3.3	SpeQuloS Performance	276
12.3.3.1	Completion Time Speedup	276
12.3.3.2	Execution Stability	278
12.3.3.3	Completion Time Prediction	278
12.3.3.4	Conclusion on Performance	279
12.3.4	SpeQuloS Deployment in the European Desktop Grid Infrastructure	279
12.3.5	Related Works	281
12.3.6	Conclusion	281
	Bibliography	282

13 Supporting Web 2.0 Communities by Volunteer

Desktop Grids	287
<i>Peter Kacsuk, Attila Marosi, Lovas Robert, and Jozsef Kovacs</i>	
13.1	Introduction 288
13.1.1	Desktop Grids and Volunteer Computing 288
13.1.2	Web 2.0 Technologies 289
13.1.3	Overview of the Web2Grid Initiative 289
13.1.3.1	Aspect 1: Web 2.0, Strength of the Community to Integrate Resources 291
13.1.3.2	Aspect 2: Tasks Arising from the Web 2.0 In- frastructure 291
13.1.4	Aspect 3: Critical Tasks with Respect to the Future Per- spectives of the Web 2.0 Infrastructure 291
13.1.4.1	Aspect 4: Tasks Demanded by the Members of the Web 2.0 Community 292
13.1.4.2	Aspect 5: Extension of Service/Desktop Grid to Business Grid 292
13.1.4.3	Aspect 6: Integrating Volunteer Desktop Grids to a Payment System Framework 292
13.2	Related Works 292
13.3	BOINC and SZDG 294
13.3.1	BOINC and Volunteer Computing 294
13.3.2	SZDG (SZTAKI Desktop Grid) 295
13.4	Survey on Web2Grid Technologies 297
13.5	Bridging Web 2.0 and the Grid 300
13.5.1	Web 2.0 Applications 300
13.5.2	Accounting System 301
13.5.3	Registration and the Payment System 302
13.5.4	Grid 302
13.5.5	Remote Validation of Tasks 303

13.6 Conclusions	304
Acknowledgment	305
Bibliography	305
14 Programming Applications for Desktop Grids	309
<i>Tamas Kiss and Gabor Terstysnszky</i>	
14.1 Generic Methodology for Porting Applications to Desktop and Service Grid Infrastructures	310
14.1.1 Motivations for Designing a Methodology for Grid Application Porting and Development	310
14.1.2 EADM Participants and Roles	311
14.1.3 EADM Stages and Outcomes	312
14.1.3.1 Analysis of Current Application	312
14.1.3.2 Requirements Analysis	314
14.1.3.3 Systems Design	315
14.1.3.4 Detailed Design and Implementation	317
14.1.3.5 Further Stages of EADM	318
14.2 Requirements toward Applications to Be Run on Desktop Grid Infrastructures	319
14.3 Application Scenarios for Combined Desktop and Service Grid Infrastructures	321
14.3.1 Applications Running through the Desktop-to-Service Grid Bridge	321
14.3.2 Applications Running through the SG-to-DG Bridge	322
14.3.3 Applications Using Specific Job Submission or Scheduling Systems to Utilize to SG/DG Resources	323
14.4 Molecular Docking Simulations: Application Case Study	325
14.4.1 Modeling Carbohydrate Recognition	325
14.4.2 Docking Simulations on Desktop and Service Grid Resources	326
14.4.2.1 AutoDock as a BOINC Master Worker Application	327
14.4.2.2 AutoDock as a BOINC Application in the WS P-GRADE Portal	328
14.4.2.3 Running AutoDock on EGI Resources Supported by Desktop Grids	329
14.5 Conclusion	329
Bibliography	330
15 Network Awareness in Volunteer Networks	333
<i>Jon B. Weissman and Jinoh Kim</i>	
15.1 Introduction	333
15.2 Communication Makespan	334
15.2.1 Definition of Communication Makespan	335
15.2.2 Server Selection Heuristics	336

- 15.2.3 Performance Evaluation 338
- 15.3 OPEN Framework 340
 - 15.3.1 Secondhand Estimation 341
 - 15.3.2 OPEN Framework 342
 - 15.3.2.1 Secondhand Estimation Method 342
 - 15.3.2.2 Measurement Sharing Method 345
 - 15.3.3 Evaluation of OPEN-Based Selection 347
- 15.4 Network Dashboard 350
- 15.5 Conclusion 352
- Bibliography 353

Index **357**