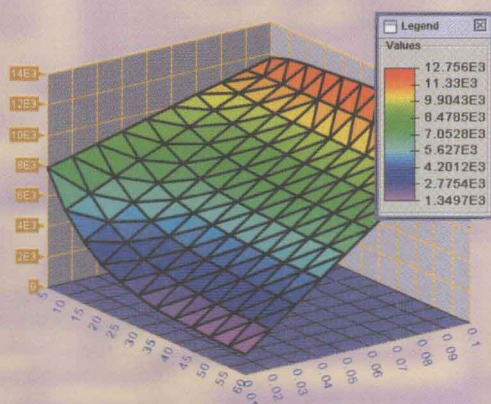


Radu Prodan
Thomas Fahringer

LNCS 4340

Grid Computing

Experiment Management, Tool Integration,
and Scientific Workflows



Radu Prodan Thomas Fahringer

Grid Computing

Experiment Management, Tool Integration,
and Scientific Workflows

 Springer

Authors

Radu Prodan
Thomas Fahringer
University of Innsbruck, Institute for Computer Science
Technikerstr. 21a, 6020 Innsbruck, Austria
E-mail: {radu,thomas.fahringer}@dps.uibk.ac.at

Library of Congress Control Number: 2006939015

CR Subject Classification (1998): C.2, D.4, F.3, H.4, H.3, C.4, I.2.8

LNCS Sublibrary: SL 2 – Programming and Software Engineering

ISSN	0302-9743
ISBN-10	3-540-69261-4 Springer Berlin Heidelberg New York
ISBN-13	978-3-540-69261-4 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media
springer.com

© Springer-Verlag Berlin Heidelberg 2007
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Markus Richter, Heidelberg
Printed on acid-free paper SPIN: 11965800 06/3142 5 4 3 2 1 0

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

University of Dortmund, Germany

Madhu Sudan

Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Moshe Y. Vardi

Rice University, Houston, TX, USA

Gerhard Weikum

Max-Planck Institute of Computer Science, Saarbruecken, Germany

To our families

Preface

In the last decade, the interest in computational Grids has increasingly grown in the scientific community as a means of enabling the application developers to aggregate resources scattered around the globe for solving large-scale scientific problems. As applications get larger, more complex and dynamic, the use of software tools becomes vital for tuning application parameters, identifying performance leaks, or detecting program defects. Extensive efforts within academia and industry over the last decade resulted in a large collection of tools for practical application engineering. Available tools of broad interest include program source and structure browsers, editors, static program analysers, performance predictors, optimising compilers, schedulers, execution control and monitoring environments, sequential and parallel debuggers (providing deadlock detection and deterministic replay mechanisms), checkpoints, data and execution visualisers, performance analysers, or various program tracers.

Despite all these extensive efforts, building applications that can effectively utilise the Grid still remains an art due to the lack of appropriate high-level tools to support the developers. In this monograph, we address four critical software development aspects for the engineering and execution of applications on parallel and Grid architectures.

First of all, existing available performance analysis tools target single application execution which is not sufficient for effective performance tuning of parallel applications. The most popular performance metrics such as speedup or efficiency require repeated execution of the application for various machine sizes for which no automatic tool support exists so far. Additionally, parallelising and tuning of applications for a certain compute platform requires repeated experimentation for various data distributions, loop iteration scheduling strategies, or compiler optimisation options, which is known to be an NP-complete problem.

Second, tool portability is critical since tools are often based on a monolithic design that does not isolate the inherent platform dependencies required to support advanced and in-depth analysis. For example, when using a new

parallel system the users must in most cases learn about and familiarise themselves with new tools with different functionality and interfaces, which is in many cases very time consuming and can be a major barrier in using novel modern computer architectures.

Third, existing tools cannot be used in cooperation on the same application instance to enhance the performance and correctness debugging engineering process since they are not designed for interoperability and often are based on incompatible instrumentation or monitoring systems.

Fourth, the workflow model that recently emerged as a new attractive paradigm for programming loosely coupled Grid infrastructures requires novel tools that offer appropriate high-level support, including abstract specification mechanisms, optimised scheduling, and scalable fault-tolerant execution, which are of paramount importance to effectively running distributed large-scale applications. These topics attract a lot of interest within the Grid community that aims to evolve the Grid to a commodity platform that transparently aggregates high-performance resources scattered around the globe in a single virtual supercomputer for performing scientific simulations.

In this monograph, we first propose a new directive-based language called ZEN for compact specification of wide value ranges of interest for arbitrary application parameters, including problem or machine sizes, array or loop distributions, software libraries, interconnection networks, or target execution machines. We design the ZEN directives as problem-independent with global or fine-grain scopes that do not change the semantics of the application, nor require any application modification or special preparation. Irrelevant or meaningless experiments can be filtered through well-defined constraints. Additionally, the ZEN directives can be used to request a wide range of performance metrics to be collected from the application for arbitrary code regions.

Based on the ZEN language, we develop a novel experiment management tool called ZENTURIO for automatic experiment management of large-scale performance and parameter studies on parallel and Grid architectures. ZENTURIO offers automatic analysis and visualisation support across multiple experiments based on the performance and output data collected and organised in a common shared data repository. In contrast to existing parameter study tools, ZENTURIO requires no special preparation of the application, nor does it restrict the parametrisation to input files or to global input arguments. We validated the functionality and usefulness of ZENTURIO on several real-world parallel applications from various domains, including theoretical chemistry, photonics, finances, and numerical mathematics.

We designed ZENTURIO as a comprehensive distributed service-oriented architecture for interoperable tool development based on the latest state-of-the-art Web and Grid services technologies. We illustrate how a service-oriented architecture facilitates the integration of a broad set of tools and enables a range of useful tool interoperability scenarios that facilitate the

engineering effort of applications. We illustrate a variety of novel adaptations of state-of-the-art Web technologies for Grid computing which anticipated several existing standardisation efforts.

Based on the ZENTURIO experiment management architecture, we propose a generic optimisation framework that integrates general-purpose meta-heuristics for solving NP-complete performance and parameter optimisation problems in an exponential search space specified using the ZEN experiment specification language. We illustrate a generic problem-independent realisation of the search engine using a genetic algorithm that allows new optimisation problems to be formulated through appropriate objective functions, for example, a performance metric using the ZEN language. We illustrate three case studies that instantiate the framework for Grid workflow scheduling, throughput scheduling of parameter studies, and performance tuning of parallel applications on the Grid using irregular array distributions.

Finally, we propose a timely approach for modelling and executing scientific workflows in dynamic and heterogeneous Grid environments. We introduce an abstract formal model for hierarchical representation of complex directed graph-based workflows using composite activities (such as parallel and sequential loops or conditional activities) interconnected through control and data flow dependencies comprising advanced collective communication patterns such as broadcast, scatter, and gather. We propose and comparatively analyse three heuristic-based algorithms for scheduling two real-world scientific workflows from material science and meteorology domains. The scheduled applications achieve good performance on the Austrian Grid environment using advanced runtime techniques such as partitioning, workflow optimisation, and load balancing. We design a steering algorithm that performs runtime monitoring and workflow schedule adaptations which ensure that certain quality of service performance contracts are preserved during execution of the workflow. We conclude with a classification of the most important performance overheads that may slow down the performance of scientific workflows and validate them through several experiments.

Innsbruck, October 2006

Radu Prodan
Thomas Fahringer

Lecture Notes in Computer Science

For information about Vols. 1–4264

please contact your bookseller or Springer

Vol. 4377: M. Abe (Ed.), *Topics in Cryptology – CT-RSA 2007*. XI, 403 pages. 2006.

Vol. 4369: M. Umeda, A. Wolf, O. Bartenstein, U. Geske, D. Seipel, O. Takata (Eds.), *Declarative Programming for Knowledge Management*. X, 229 pages. 2006. (Sublibrary LNAI).

Vol. 4361: H.J. Hoogeboom, G. Păun, G. Rozenberg, A. Salomaa (Eds.), *Membrane Computing*. IX, 555 pages. 2006.

Vol. 4357: L. Buttyán, V. Gligor, D. Westhoff (Eds.), *Security and Privacy in Ad-hoc and Sensor Networks*. X, 193 pages. 2006.

Vol. 4355: J. Julliand, O. Kouchnarenko (Eds.), *B 2007: Formal Specification and Development in B*. XIII, 293 pages. 2006.

Vol. 4353: T. Schwentick, D. Suciu (Eds.), *Database Theory – ICDT 2007*. XI, 419 pages. 2006.

Vol. 4348: S.T. Taft, R.A. Duff, R.L. Brukardt, E. Ploedereder, P. Leroy (Eds.), *Ada 2005 Reference Manual*. XXII, 765 pages. 2006.

Vol. 4347: J. Lopez (Ed.), *Critical Information Infrastructures Security*. X, 286 pages. 2006.

Vol. 4345: N. Maglaveras, I. Chouvarda, V. Koutkias, R. Brause (Eds.), *Biological and Medical Data Analysis*. XIII, 496 pages. 2006. (Sublibrary LNBI).

Vol. 4344: V. Gruhn, F. Oquendo (Eds.), *Software Architecture*. X, 245 pages. 2006.

Vol. 4342: H. de Swart, E. Orłowska, G. Schmidt, M. Roubens (Eds.), *Theory and Application of Relational Structures as Knowledge Instruments II*. X, 373 pages. 2006. (Sublibrary LNAI).

Vol. 4341: P.Q. Nguyen (Ed.), *Progress in Cryptology – VIETCRYPT 2006*. XI, 385 pages. 2006.

Vol. 4340: R. Prodan, T. Fahringer, *Grid Computing*. XXIII, 317 pages. 2007.

Vol. 4339: E. Ayguadé, G. Baumgartner, J. Ramanujam, P. Sadayappan (Eds.), *Languages and Compilers for Parallel Computing*. XI, 476 pages. 2006.

Vol. 4338: P. Kalra, S. Peleg (Eds.), *Computer Vision, Graphics and Image Processing*. XV, 965 pages. 2006.

Vol. 4337: S. Arun-Kumar, N. Garg (Eds.), *FSTTCS 2006: Foundations of Software Technology and Theoretical Computer Science*. XIII, 430 pages. 2006.

Vol. 4333: U. Reimer, D. Karagiannis (Eds.), *Practical Aspects of Knowledge Management*. XII, 338 pages. 2006. (Sublibrary LNAI).

Vol. 4332: A. Bagchi, V. Atluri (Eds.), *Information Systems Security*. XV, 382 pages. 2006.

Vol. 4331: G. Min, B. Di Martino, L.T. Yang, M. Guo, G. Ruenger (Eds.), *Frontiers of High Performance Computing and Networking – ISPA 2006 Workshops*. XXXVII, 1141 pages. 2006.

Vol. 4330: M. Guo, L.T. Yang, B. Di Martino, H.P. Zima, J. Dongarra, F. Tang (Eds.), *Parallel and Distributed Processing and Applications*. XVIII, 953 pages. 2006.

Vol. 4329: R. Barua, T. Lange (Eds.), *Progress in Cryptology – INDOCRYPT 2006*. X, 454 pages. 2006.

Vol. 4328: D. Penkler, M. Reitenspiess, F. Tam (Eds.), *Service Availability*. X, 289 pages. 2006.

Vol. 4327: M. Baldoni, U. Endriss (Eds.), *Declarative Agent Languages and Technologies IV*. VIII, 257 pages. 2006. (Sublibrary LNAI).

Vol. 4326: S. Göbel, R. Malkewitz, I. Iurgel (Eds.), *Technologies for Interactive Digital Storytelling and Entertainment*. X, 384 pages. 2006.

Vol. 4325: J. Cao, I. Stojmenovic, X. Jia, S.K. Das (Eds.), *Mobile Ad-hoc and Sensor Networks*. XIX, 887 pages. 2006.

Vol. 4320: R. Gotzhein, R. Reed (Eds.), *System Analysis and Modeling: Language Profiles*. X, 229 pages. 2006.

Vol. 4319: L.-W. Chang, W.-N. Lie (Eds.), *Advances in Image and Video Technology*. XXVI, 1347 pages. 2006.

Vol. 4318: H. Lipmaa, M. Yung, D. Lin (Eds.), *Information Security and Cryptology*. XI, 305 pages. 2006.

Vol. 4317: S.K. Madria, K.T. Claypool, R. Kannan, P. Uppuluri, M.M. Gore (Eds.), *Distributed Computing and Internet Technology*. XIX, 466 pages. 2006.

Vol. 4316: M.M. Dalkilic, S. Kim, J. Yang (Eds.), *Data Mining and Bioinformatics*. VIII, 197 pages. 2006. (Sublibrary LNBI).

Vol. 4313: T. Margaria, B. Steffen (Eds.), *Leveraging Applications of Formal Methods*. IX, 197 pages. 2006.

Vol. 4312: S. Sugimoto, J. Hunter, A. Rauber, A. Morishima (Eds.), *Digital Libraries: Achievements, Challenges and Opportunities*. XVIII, 571 pages. 2006.

Vol. 4311: K. Cho, P. Jacquet (Eds.), *Technologies for Advanced Heterogeneous Networks II*. XI, 253 pages. 2006.

Vol. 4309: P. Inverardi, M. Jazayeri (Eds.), *Software Engineering Education in the Modern Age*. VIII, 207 pages. 2006.

Vol. 4308: S. Chaudhuri, S.R. Das, H.S. Paul, S. Tirthapura (Eds.), *Distributed Computing and Networking*. XIX, 608 pages. 2006.

Vol. 4307: P. Ning, S. Qing, N. Li (Eds.), *Information and Communications Security*. XIV, 558 pages. 2006.

- Vol. 4306: Y. Avrithis, Y. Kompatsiaris, S. Staab, N.E. O'Connor (Eds.), *Semantic Multimedia*. XII, 241 pages. 2006.
- Vol. 4305: A.A. Shvartsman (Ed.), *Principles of Distributed Systems*. XIII, 441 pages. 2006.
- Vol. 4304: A. Sattar, B.-h. Kang (Eds.), *AI 2006: Advances in Artificial Intelligence*. XXVII, 1303 pages. 2006. (Sublibrary LNAI).
- Vol. 4303: A. Hoffmann, B.-h. Kang, D. Richards, S. Tsumoto (Eds.), *Advances in Knowledge Acquisition and Management*. XI, 259 pages. 2006. (Sublibrary LNAI).
- Vol. 4302: J. Domingo-Ferrer, L. Franconi (Eds.), *Privacy in Statistical Databases*. XI, 383 pages. 2006.
- Vol. 4301: D. Pointcheval, Y. Mu, K. Chen (Eds.), *Cryptography and Network Security*. XIII, 381 pages. 2006.
- Vol. 4300: Y.Q. Shi (Ed.), *Transactions on Data Hiding and Multimedia Security I*. IX, 139 pages. 2006.
- Vol. 4299: S. Renals, S. Bengio, J.G. Fiscus (Eds.), *Machine Learning for Multimodal Interaction*. XII, 470 pages. 2006.
- Vol. 4297: Y. Robert, M. Parashar, R. Badrinath, V.K. Prasanna (Eds.), *High Performance Computing - HiPC 2006*. XXIV, 642 pages. 2006.
- Vol. 4296: M.S. Rhee, B. Lee (Eds.), *Information Security and Cryptology - ICISC 2006*. XIII, 358 pages. 2006.
- Vol. 4295: J.D. Carswell, T. Tezuka (Eds.), *Web and Wireless Geographical Information Systems*. XI, 269 pages. 2006.
- Vol. 4294: A. Dan, W. Lamersdorf (Eds.), *Service-Oriented Computing - ICSOC 2006*. XIX, 653 pages. 2006.
- Vol. 4293: A. Gelbukh, C.A. Reyes-Garcia (Eds.), *MICA 2006: Advances in Artificial Intelligence*. XXVIII, 1232 pages. 2006. (Sublibrary LNAI).
- Vol. 4292: G. Bebis, R. Boyle, B. Parvin, D. Koracin, P. Remagnino, A. Nefian, G. Meenakshisundaram, V. Pascucci, J. Zara, J. Molineros, H. Theisel, T. Malzbender (Eds.), *Advances in Visual Computing, Part II*. XXXII, 906 pages. 2006.
- Vol. 4291: G. Bebis, R. Boyle, B. Parvin, D. Koracin, P. Remagnino, A. Nefian, G. Meenakshisundaram, V. Pascucci, J. Zara, J. Molineros, H. Theisel, T. Malzbender (Eds.), *Advances in Visual Computing, Part I*. XXXI, 916 pages. 2006.
- Vol. 4290: M. van Steen, M. Henning (Eds.), *Middleware 2006*. XIII, 425 pages. 2006.
- Vol. 4289: M. Ackermann, B. Berendt, M. Grobelnik, A. Hotho, D. Mladenić, G. Semeraro, M. Spiliopoulou, G. Stumme, V. Svatek, M. van Someren (Eds.), *Semantics, Web and Mining*. X, 197 pages. 2006. (Sublibrary LNAI).
- Vol. 4288: T. Asano (Ed.), *Algorithms and Computation*. XX, 766 pages. 2006.
- Vol. 4287: C. Mao, T. Yokomori (Eds.), *DNA Computing*. XII, 440 pages. 2006.
- Vol. 4286: P. Spirakis, M. Mavronicolas, S. Kontogiannis (Eds.), *Internet and Network Economics*. XI, 401 pages. 2006.
- Vol. 4285: Y. Matsumoto, R. Sproat, K.-F. Wong, M. Zhang (Eds.), *Computer Processing of Oriental Languages*. XVII, 544 pages. 2006. (Sublibrary LNAI).
- Vol. 4284: X. Lai, K. Chen (Eds.), *Advances in Cryptology - ASIACRYPT 2006*. XIV, 468 pages. 2006.
- Vol. 4283: Y.Q. Shi, B. Jeon (Eds.), *Digital Watermarking*. XII, 474 pages. 2006.
- Vol. 4282: Z. Pan, A. Cheok, M. Haller, R.W.H. Lau, H. Saito, R. Liang (Eds.), *Advances in Artificial Reality and Tele-Existence*. XXIII, 1347 pages. 2006.
- Vol. 4281: K. Barkaoui, A. Cavalcanti, A. Cerone (Eds.), *Theoretical Aspects of Computing - ICTAC 2006*. XV, 371 pages. 2006.
- Vol. 4280: A.K. Datta, M. Gradinariu (Eds.), *Stabilization, Safety, and Security of Distributed Systems*. XVII, 590 pages. 2006.
- Vol. 4279: N. Kobayashi (Ed.), *Programming Languages and Systems*. XI, 423 pages. 2006.
- Vol. 4278: R. Meersman, Z. Tari, P. Herrero (Eds.), *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops, Part II*. XLV, 1004 pages. 2006.
- Vol. 4277: R. Meersman, Z. Tari, P. Herrero (Eds.), *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops, Part I*. XLV, 1009 pages. 2006.
- Vol. 4276: R. Meersman, Z. Tari (Eds.), *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE, Part II*. XXXII, 752 pages. 2006.
- Vol. 4275: R. Meersman, Z. Tari (Eds.), *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE, Part I*. XXXI, 1115 pages. 2006.
- Vol. 4274: Q. Huo, B. Ma, E.-S. Chng, H. Li (Eds.), *Chinese Spoken Language Processing*. XXIV, 805 pages. 2006. (Sublibrary LNAI).
- Vol. 4273: I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, L. Aroyo (Eds.), *The Semantic Web - ISWC 2006*. XXIV, 1001 pages. 2006.
- Vol. 4272: P. Havinga, M. Lijding, N. Meratnia, M. Wegdam (Eds.), *Smart Sensing and Context*. XI, 267 pages. 2006.
- Vol. 4271: F.V. Fomin (Ed.), *Graph-Theoretic Concepts in Computer Science*. XIII, 358 pages. 2006.
- Vol. 4270: H. Zha, Z. Pan, H. Thwaites, A.C. Addison, M. Forte (Eds.), *Interactive Technologies and Sociotechnical Systems*. XVI, 547 pages. 2006.
- Vol. 4269: R. State, S. van der Meer, D. O'Sullivan, T. Pfeifer (Eds.), *Large Scale Management of Distributed Systems*. XIII, 282 pages. 2006.
- Vol. 4268: G. Parr, D. Malone, M. Ó Foghlú (Eds.), *Autonomic Principles of IP Operations and Management*. XIII, 237 pages. 2006.
- Vol. 4267: A. Helmy, B. Jennings, L. Murphy, T. Pfeifer (Eds.), *Autonomic Management of Mobile Multimedia Services*. XIII, 257 pages. 2006.
- Vol. 4266: H. Yoshiura, K. Sakurai, K. Rannenbergh, Y. Murayama, S. Kawamura (Eds.), *Advances in Information and Computer Security*. XIII, 438 pages. 2006.
- Vol. 4265: L. Todorovski, N. Lavrač, K.P. Jantke (Eds.), *Discovery Science*. XIV, 384 pages. 2006. (Sublibrary LNAI).

Contents

1	Introduction	1
1.1	Motivation	2
1.1.1	Performance Tuning	2
1.1.2	Parameter Studies	3
1.1.3	Optimisation	3
1.1.4	Scheduling	3
1.1.5	Parametrisation Language	4
1.1.6	Instrumentation	4
1.1.7	Portability	5
1.1.8	Tool Interoperability	5
1.1.9	Grid Services	5
1.1.10	Scientific Workflows	6
1.2	Goals	6
1.2.1	Experiment Specification Language	6
1.2.2	Experiment Management Tool	7
1.2.3	Optimisation	8
1.2.4	Scientific Workflows	8
1.2.5	Service-Oriented Grid Architecture	9
1.2.6	Grid Services	10
1.3	Outline	10
2	Model	13
2.1	Introduction	13
2.2	Distributed Technology History	14
2.3	Web Services	15
2.3.1	Web Services Stack	16
2.3.2	Web Services Runtime Environment	18
2.4	Grid Security Infrastructure	19
2.5	Globus Toolkit	20
2.6	Grid Architectural Model	22
2.6.1	Machine Layer	22
2.6.2	Grid Services Layer	28
2.6.3	Application Layer	30
2.7	Summary	35

3	The ZEN Experiment Specification Language	37
3.1	Functionality and Use Cases	37
3.1.1	Shared Memory Application Scalability	38
3.1.2	ZEN Transformation System	39
3.1.3	Shared Memory Loop Scheduling	40
3.1.4	Distributed Processor Arrays	41
3.1.5	Distributed Memory Arrays	41
3.1.6	Work Distribution	43
3.1.7	Parameter Studies	43
3.2	Formal Language Specification	44
3.2.1	ZEN Set	44
3.2.2	ZEN Directives	49
3.2.3	ZEN Substitute Directive	50
3.2.4	Local Substitute Directive	51
3.2.5	Homonym ZEN Variables	51
3.2.6	ZEN Assignment Directive	53
3.2.7	Multi-dimensional Value Set	54
3.2.8	ZEN Constraint Directive	55
3.2.9	ZEN Performance Directive	59
3.2.10	Parameter Study Experiment	62
3.2.11	Experiment Generation Algorithm	62
3.2.12	Online Monitoring and Analysis	65
3.3	Summary	68
4	ZENTURIO Experiment Management Tool	69
4.1	User Portal Functionality	69
4.1.1	ZEN Editor	70
4.1.2	Experiment Preparation	71
4.1.3	Experiment Monitor	73
4.1.4	Application Data Visualiser	73
4.2	Performance Studies	76
4.2.1	Ocean Simulation	76
4.2.2	Linearised Augmented Plane Wave	79
4.2.3	Three-Dimensional Particle-in-Cell	84
4.2.4	Benders Decomposition	86
4.2.5	Three-Dimensional FFT Benchmarks	89
4.3	Parameter Studies	94
4.3.1	Backward Pricing	94
4.4	Architecture	105
4.4.1	Experiment Generator	107
4.4.2	Experiment Executor	108
4.4.3	Experiment State Transition Diagram	110
4.4.4	Experiment Data Repository	110
4.5	Summary	111

5	Tool Integration	113
5.1	Architecture	114
5.2	Interoperable Tool Set	116
5.2.1	Object Code Browser	117
5.2.2	Function Profiler (Z_prof)	117
5.2.3	Function Tracer (Z_trace)	118
5.2.4	Function Coverager (Z_cov)	119
5.2.5	Sequential Debugger (Z_debug)	121
5.2.6	Memory Allocation Tool (Z_MAT)	121
5.2.7	Resource Tracker (Z_RT ²)	122
5.2.8	Deadlock Detector (Z_deadlock)	122
5.3	Tool Interoperability	122
5.3.1	Classification	122
5.3.2	Interaction with a Browser	123
5.3.3	Performance Steering	124
5.3.4	Just-in-Time Debugging	126
5.3.5	Interaction with a Debugger	127
5.4	The Monitoring Layer	128
5.4.1	Dynamic Instrumentation	128
5.4.2	The Process Manager	130
5.4.3	Dynamic Instrumentation of MPI Applications	134
5.5	The Grid Services Layer	136
5.5.1	Web Application and Services Platform (WASP)	138
5.5.2	Service Repository	139
5.5.3	Abstract Grid Service	140
5.5.4	Factory	142
5.5.5	Registry	143
5.5.6	WSDL Compatibility	144
5.5.7	Dynamic Instrumentor	144
5.5.8	Aggregator	145
5.6	Event Framework	146
5.6.1	Representation	146
5.6.2	Implementation	149
5.6.3	Filters	151
5.7	Firewall Management	151
5.8	WASP Versus GT3 Technology Evaluation	152
5.8.1	Stub Management	153
5.8.2	Service Lifecycle	154
5.8.3	UDDI-Based Service Repository	155
5.8.4	Service Data	155
5.8.5	Events	155
5.8.6	Registry	156
5.8.7	Security	159

5.8.8	Grid Service Throughput	160
5.8.9	Comparison	163
5.9	Summary	164
6	Optimisation Framework	165
6.1	Workflow Scheduling	167
6.1.1	Schedule Dependencies	169
6.1.2	Objective Function	170
6.2	Genetic Search Engine	174
6.2.1	Initial Population	175
6.2.2	Selection	177
6.2.3	Crossover	177
6.2.4	Mutation	178
6.2.5	Elitist Model	178
6.2.6	Fitness Scaling	179
6.2.7	Convergence Criterion	180
6.3	Genetic Workflow Scheduling	180
6.3.1	WIEN2k	180
6.4	Throughput Scheduling	192
6.5	Performance Tuning of Parallel Applications	194
6.5.1	Parallel Applications on the Grid	195
6.6	Summary	201
7	Scientific Grid Workflows	203
7.1	Workflow Model	204
7.1.1	Computational Activity	205
7.1.2	Control Flow Dependencies	206
7.1.3	Data Flow Dependencies	207
7.1.4	Conditional Activity	207
7.1.5	Parallel Loop Activity	208
7.1.6	Sequential Loop Activity	211
7.1.7	Workflow Activity	213
7.2	Scheduler	214
7.2.1	Workflow Converter	214
7.2.2	Scheduling Engine	220
7.2.3	Layered Partitioning	226
7.2.4	WIEN2k	227
7.2.5	Invmod	231
7.3	Enactment Engine	235
7.3.1	Workflow Partitioning	236
7.3.2	Control Flow Management	241
7.3.3	Data Flow Management	242
7.3.4	Virtual Single Execution Environment	243
7.3.5	Workflow Steering	244
7.3.6	Fault Tolerance	248

7.3.7	WIEN2k Execution Experiments	253
7.3.8	Steering Experiments	255
7.4	Overhead Analysis	260
7.4.1	Experiments	263
7.5	Summary	269
8	Related Work	271
8.1	Experiment Management	271
8.2	Performance Study	271
8.3	Parameter Study	273
8.4	Optimisation and Scheduling	273
8.5	Tool Integration	274
8.5.1	Scientific Workflows	276
9	Conclusions	279
9.1	Contributions	279
9.1.1	Experiment Specification	279
9.1.2	Experiment Management	280
9.1.3	Optimisation	281
9.1.4	Tool Integration Design	281
9.1.5	Web Services for the Grid	283
9.1.6	Scientific Workflows	283
10	Appendix	285
10.1	Notations	285
10.2	Code Regions	288
10.3	Abbreviations	289
10.4	Performance Metrics	292
	References	297
	Index	311

List of Figures

2.1	The interoperable Web services stack.	16
2.2	The best practices of publishing a Web service into a UDDI Service Repository.	18
2.3	The Web services runtime environment.	19
2.4	The GSI single sign-on and proxy delegation chain of trust. . .	20
2.5	The Grid architectural model.	23
2.6	The von Neumann architecture.	24
2.7	The Symmetric Multiprocessor (SMP) architecture.	24
2.8	The Cluster of Workstation (COW) architecture.	26
2.9	The SMP cluster architecture.	26
2.10	The Cache Coherent Non-Uniform Memory Access (ccNUMA) architecture.	27
2.11	The Grid hardware architecture.	28
2.12	The stateful Grid service design alternatives.	30
2.13	The parallel application execution model.	31
2.14	The execution model of parallel applications on the Grid.	32
3.1	The ZEN Transformation System.	40
3.2	The (CYCLIC(2), BLOCK) distribution of array $A(8,4)$ onto processor array $P(2,2)$	42
3.3	The ZEN constraint defined by Example 3.5.	44
3.4	The ZEN set element evaluation function.	46
3.5	The ZEN file instances generated by Example 3.2.	50
3.6	The value set constraint defined in Example 3.21.	57
3.7	The experiment generation algorithm data flow.	64
4.1	The ZENTURIO User Portal main panel.	70
4.2	The ZEN editor.	71
4.3	The Experiment Preparation dialog-box.	72
4.4	The Application Data Visualiser for performance studies.	74
4.5	The Application Data Visualiser for parameter studies.	75

4.6	The Stommel model performance results for various intra-node and inter-node machine sizes (I), 200×200 problem size, 20000 iterations.	80
4.7	The Stommel model performance results for various intra-node and inter-node machine sizes (II), 400×400 problem size, 40000 iterations.	81
4.8	The Stommel model performance results (III).	82
4.9	The LAPW0 performance results for various machine sizes. ...	85
4.10	The 3DPIC performance results for various machine sizes. ...	87
4.11	The benders decomposition performance results for various machine sizes.	90
4.12	The parallel three-dimensional FFT computation.	91
4.13	The three-dimensional FFT benchmark results (I).	95
4.14	The three-dimensional FFT benchmark results (II).	96
4.15	The three-dimensional FFT benchmark results (III).	97
4.16	The three-dimensional FFT benchmark results (IV).	98
4.17	The three-dimensional FFT benchmark results (V).	99
4.18	The three-dimensional FFT benchmark results (VI).	100
4.19	The three-dimensional FFT benchmark results (VII).	101
4.20	The constraint defined in Example 4.19.	103
4.21	The backward pricing parameter study results.	104
4.22	The ZENTURIO experiment management tool architecture. .	105
4.23	The Experiment Generator architecture.	107
4.24	The experiment state transition diagram.	110
4.25	The Experiment Data Repository schema.	111
5.1	The tool integration service-oriented architecture.	115
5.2	A snapshot of interoperable online software tools.	124
5.3	The steering configuration.	125
5.4	The cyclic debugging states.	126
5.5	A just-in-time debugging scenario.	127
5.6	The dynamic instrumentation control flow.	129
5.7	The Process Manager architecture.	130
5.8	The instrumentation probe class hierarchy.	132
5.9	The control flow for starting an MPI(CH) application for dynamic instrumentation.	135
5.10	The dynamic MPI library profiling.	137
5.11	The state transition diagram of WASP-based Web services. ...	139
5.12	The Grid services hierarchy.	141
5.13	The ZENTURIO event architecture.	147
5.14	The event hierarchy.	148
5.15	The Registry throughput results.	158
5.16	The secure versus insecure response time comparison.	160
5.17	The throughput results of WASP, GT3, and vanilla Axis services.	162