

REFERENCE

- Abbas, A., 2003, *Grid Computing: A Practical Guide to Technology and Applications*, Charles River Media, Inc, Hingham, Massachusetts.
- Alhamad, M., Dillon, T. & Chang, E., 2010, 'Conceptual SLA framework for cloud computing.' *In Proceedings of the IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, Dubai, pp. 606 - 610.
- Amazon, 2010a, 'Amazon Elastic Compute Cloud', Available from: <http://aws.amazon.com/ec2/#pricing>, [Accessed June 2010].
- Amazon, 2010b, 'Amazon Simple Storage Service', Available from: <http://aws.amazon.com/s3/>, [Accessed June 2010].
- Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S. & Xu, M., 2005, 'Web Services Agreement Specification (WS-Agreement) Version 2005/09.', *Technical Report*, Global Grid Forum.
- Andrzejak, A., Reinefeld, A., Schintke, F. & Schütt, T., 2004, 'On Adaptability in Grid Systems.' *In Proceedings of the Workshop of Future Generation Grids*, Dagstuhl, Germany, pp. 29 - 46.
- Aoun, R. & Gagnaire, M., 2009, 'Towards a fairer benefit distribution in grid environments.' *In Proceedings of the IEEE/ACS International Conference on Computer Systems and Applications (AICCSA 2009)*, Rabat, Morocco, pp. 21 - 26.
- Arfa, R. & Broeckhove, J., 2009, 'Modeling resource prices in computational commodity markets.', *In Proceedings of the IEEE/ACS International Conference on Computer Systems and Applications (AICCSA 2009)* Rabat, Morocco, pp. 145 - 152.
- Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. & Zaharia, M., 2009, 'Above the Clouds: A Berkeley View of Cloud Computing', *Technical Report*, Berkeley RAD Lab.
- Baker, M., Buyya, R. & Laforenza, D., 2002, 'Grids and grid technologies for wide-area distributed computing.' *Journal of Software-Practice & Experience*, Vol. 32, No. 15, pp. 1437 - 1466.
- Baskiyar, S. & Dickinson, C., 2005, 'Scheduling directed a-cyclic task graphs on a bounded set of heterogeneous processors using task duplication.' *Journal of Parallel and Distributed Computing*, Vol. 65, No. 8, pp. 911 - 921.
- Bellwood, T., 2002, 'UDDI version 2.04 API specification', *UDDI Committee Specification*, OASIS.
- Bolloor, K., Chirkova, R., Salo, T. & Viniotis, Y., 2010, 'Heuristic-Based Request Scheduling Subject to a Percentile Response Time SLA in a Distributed Cloud.' *In Proceedings of the IEEE Global Telecommunications Conference GLOBECOM*, Miami, Florida, pp. 1 - 6.
- Braun, T. D., Siegel, H. J., Beck, N., Boloni, L. L., Maheswaran, M., Reuther, A. I., Robertson, J. P., Theys, M. D., Yao, B., Hensgen, D. & Freund, R. F., 2001, 'A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems.' *Journal of Parallel and Distributed Computing*, Vol. 61, No. 6, pp. 810 - 837.
- Buyya, R., 1999, *High Performance Cluster Computing: Architectures and Systems*, Prentice Hall PTR, United States of America.
- Buyya, R., Abramson, D., Giddy, J. & Stockinger, H., 2002, 'Economic Models for Resource Management and Scheduling in Grid Computing.' *Journal of*

- Concurrency and Computation: Practice and Experience*, Vol. 14, No. 13 - 15, pp. 1507 - 1542.
- Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J. & Brandic, I., 2009, 'Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility.' *Future Generation Computer Systems*, Vol. 25, No. 6, pp. 599 - 616.
- Chaisiri, S., Lee, B. & Niyato, D., 2011, 'Optimization of Resource Provisioning Cost in Cloud Computing.' *IEEE Transactions on Services Computing*, Vol. No. pp.
- Chervenak, A., Foster, I., Kesselman, C., Salisbury, C. & Tuecke, S., 2000, 'The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets.' *Journal of Network and Computer Applications*, Vol. 23, No. 3, pp. 187 - 200.
- Clematis, A., Corana, A., D'Agostino, D., Galizia, A. & Quarati, A., 2010, 'Job-resource matchmaking on Grid through two-level benchmarking.' *Future Generation Computer Systems*, Vol. 26, No. 8, pp. 1165 - 1179.
- Coulouris, G., Dollimore, J. & Kindberg, T., 2005, *Distributed Systems: Concepts and Design*, 4th edn, Pearson Education Limited, England.
- Deelman, E., Gannon, D., Shields, M. & Taylor, I., 2009, 'Workflows and e-Science: An overview of workflow system features and capabilities.' *Future Generation Computer Systems*, Vol. 25, No. 5, pp. 528 - 540.
- Deitel, P. J. & Deitel, H. M., 2009, *Internet & World Wide Web How to Program*, 4th edn, Pearson Education International, New Jersey.
- Diamadopoulou, V., Makris, C., Panagis, Y. & Sakkopoulos, E., 2008, 'Techniques to support Web Service selection and consumption with QoS characteristics.' *Journal of Network and Computer Applications*, Vol. 31, No. 2, pp. 108 - 130.
- Diaz, J., Muoz-Caro, C., Nio, A., 2009, 'A Fault Tolerant Adaptive Method for the Scheduling of Tasks in Dynamic Grids.' *In Proceedings of the Third International Conference on Advanced Engineering Computing and Applications in Sciences*, Sliema, Malta, pp. 51 - 56.
- Dikaiakos, M. D., 2007, 'Grid benchmarking: vision, challenges, and current status: Research Articles.' *Concurrency and Computation: Practice & Experience*, Vol. 19, No. 1, pp. 89 - 105.
- Ding, D., Siwei, L. & Zhan, G., 2009, 'A Dual Heuristic Scheduling Strategy Based on Task Partition in Grid Environments.' *In Proceedings of the International Joint Conference on Computational Sciences and Optimization (CSO 2009)*, Hainan, China, pp. 63 - 67.
- Dong, F. & Akl, S. G., 2009, 'Scheduling Algorithms for Grid Computing: State of the Art and Open Problems.', *Technical Report*, School of Computing, Queen's University, Canada.
- Dong, F. & Selim, G. A., 2007, 'PFAS: A Resource-Performance-Fluctuation-Aware Workflow Scheduling Algorithm for Grid Computing.' *In Proceedings of the Parallel and Distributed Processing Symposium*, Long Beach, CA, pp. 1 - 9.
- Doulamis, N. D., Doulamis, A. D., Varvarigos, E. A. & Varvarigou, T. A., 2007, 'Fair Scheduling Algorithms in Grids.' *IEEE Transactions on Parallel and Distributed Systems*, Vol. 18, No. 11, pp. 1630 - 1648.
- Erl, T., 2005, *Service-Oriented Architecture: Concepts, Technology, and Design*, Pearson Education, United States of America.

- Falzon, G. & Li, M., 2009, 'Evaluating Heuristics for Grid Workflow Scheduling.' *In Proceedings of the Fifth International Conference on Natural Computation*, Tianjin, China, pp. 227 - 231.
- Fard, H. M. & Deldari, H., 2008, 'An Economic Approach for Scheduling Dependent Tasks in Grid Computing.' *In Proceedings of the 11th IEEE International Conference on Computational Science and Engineering Workshops (CSEWORKSHOPS '08)*, San Paulo, Brazil, pp. 71 - 76.
- Fatos, X., Leonard, B., Antonio, G. J. & Pawel, J., 2009, 'A Static Benchmarking for Grid Scheduling Problems.' *In Proceedings of the International Conference on Advanced Information Networking and Applications Workshops (WAINA '09)*, pp. 170 - 175.
- Feng, X., Shen, J. & Fan, Y., 2009, 'REST: An alternative to RPC for Web services architecture.' *In Proceedings of the First International Conference on Future Information Networks, (ICFIN 2009)*, Beijing, China, pp. 7 - 10.
- Ferretti, S., Ghini, V., Panzieri, F., Pellegrini, M. & Turrini, E., 2010, 'QoS-Aware Clouds.' *In Proceedings of the IEEE 3rd International Conference on Cloud Computing*, Miami, Florida, pp. 321 - 328.
- Feuerlicht, G., 2010, 'Next Generation SOA: Can SOA Survive Cloud Computing?', in Snášel, V., Szczepaniak, P., Abraham, A. & Kacprzyk, J., *Advances in Intelligent Web Mastering - 2*, Springer Berlin / Heidelberg.
- Fielding, R. T., 2000, 'Architectural Styles and the Design of Network-based Software Architectures', *PhD Dissertation*, University of California, Irvine.
- Fito, J. O., Goiri, I. & Guitart, J., 2010, 'SLA-driven Elastic Cloud Hosting Provider.' *In Proceedings of the 18th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, Pisa, pp. 111 - 118.
- Forestiero, A. & Mastroianni, C., 2009, 'A Swarm Algorithm for a Self-Structured P2P Information System.' *IEEE Transactions on Evolutionary Computation*, Vol. 13, No. 4, pp. 681 - 694.
- Foster, I. & Kesselman, C., 1999, *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann Publishers, United States of America.
- Foster, I. & Kesselman, C., 2004, *The Grid 2: Blueprint for a New Computing Infrastructure*, 2nd edn, Morgan Kaufmann Publishers, United States of America.
- Foster, I., Kesselman, C., Nick, J. M. & Tuecke, S., 2002, 'Grid services for distributed system integration.' *IEEE Computer*, Vol. 35, No. 6, pp. 37 - 46.
- Foster, I., Kesselman, C. & Tuecke, S., 2001, 'The anatomy of the Grid: Enabling scalable virtual organizations.', *International Journal of Supercomputer Applications*, Vol. 15, No. 3, pp. 200 - 222.
- Foster, I., Yong, Z., Raicu, I. & Lu, S., 2008, 'Cloud Computing and Grid Computing 360-Degree Compared.' *In Proceedings of the Grid Computing Environments Workshop (GCE '08)*, Austin, Texas, pp. 1 - 10.
- Frumkin, M. & Van der Wijngaart, R. F., 2001, 'NAS Grid Benchmarks: a tool for Grid space exploration.' *In Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing*, pp. 315 - 322.
- FutureMark, 2010, 'PCMark05', Available from: <http://www.futuremark.com/products/pcmark05/>, [Accessed September 2010].
- Gallet, M., Marchal, L. & Vivien, F., 2009, 'Efficient scheduling of task graph collections on heterogeneous resources.' *In Proceedings of the IEEE International Symposium on Parallel & Distributed Processing (IPDPS 2009)*, Rome, Italy, pp. 1 - 11.

- Gao, Z., Luo, S. & Ding, D., 2007, 'A Scheduling Mechanism Considering Simultaneous Running of Grid Tasks and Local Tasks in the Computational Grid.' *In Proceedings of the International Conference on Multimedia and Ubiquitous Engineering (MUE '07)*, Seoul, Korea, pp. 1100 - 1105.
- Garg, S. K., Venugopal, S. & Buyya, R., 2008, 'A Meta-scheduler with Auction Based Resource Allocation for Global Grids.' *In Proceedings of the 14th IEEE International Conference on Parallel and Distributed Systems (ICPADS '08)*, Melbourne, Australia, pp. 187 - 194.
- Google, 2010a, 'Google App Engine', Available from: <http://code.google.com/appengine/>, [Accessed June 2010].
- Google, 2010b, 'Google Apps', Available from: <http://www.google.com/apps/>, [Accessed June 2010].
- Gramma, A., Gupta, A., Karypis, G. & Kumar, V., 2003, *Introduction to Parallel Computing*, 2nd edn, Addison Wesley.
- Hamscher, V., Schwiegelshohn, U., Streit, A. & Yahyapour, R., 2000, 'Evaluation of Job-Scheduling Strategies for Grid Computing.' *In Proceedings of the First IEEE/ACM International Workshop on Grid Computing*, pp. 191 - 202.
- Henning, J. L., 2000, 'SPEC CPU2000: measuring CPU performance in the New Millennium.' *IEEE Computer*, Vol. 33, No. 7, pp. 28 - 35.
- House, B., Marshall, P., Oberg, M., Tufo, H. M. & Woitaszek, M., 2008, 'Grid service hosting on virtual clusters.' *In Proceedings of the 9th IEEE/ACM International Conference on Grid Computing*, Tsukuba, Japan, pp. 304 - 309.
- Hussain, A., Shoeb, A. A. M., Bikas, M. A. N. & Hasan, M. K., 2008, 'Runtime thread rescheduling: An extended scheduling algorithm to enhance the performance of the Gridbus broker.' *In Proceedings of the IEEE International Multitopic Conference (INMIC 2008)*, Khulna, Bangladesh, pp. 412 - 417.
- Hyun Jin, M., Yun, C. & Hacigumus, H., 2010, 'SLA-Aware Profit Optimization in Cloud Services via Resource Scheduling.' *In Proceedings of the 6th World Congress on Services (SERVICES-1)*, Miami, Florida, pp. 152 - 153.
- Joshi, A., Phansalkar, A., Eeckhout, L. & John, L. K., 2006, 'Measuring benchmark similarity using inherent program characteristics.' *IEEE Transactions on Computers*, Vol. 55, No. 6, pp. 769 - 782.
- Kaur, D. & Sengupta, J., 2007, 'Resource discovery in web services based grids.', *In Proceedings of the World Academy of Science, Engineering and Technology*, Amsterdam, Netherlands, pp. 284 - 288.
- Kecskemeti, G., Kacsuk, P., Terstyanszky, G., Kiss, T. & Delaitre, T., 2008, 'Automatic Service Deployment Using Virtualisation.' *In Proceedings of the 16th Euromicro Conference on Parallel, Distributed and Network-Based Processing (PDP 2008)*, Toulouse, France, pp. 628 - 635.
- Kejariwal, A., Veidenbaum, A. V., Nicolau, A., Tian, X., Girkar, M., Saito, H. & Banerjee, U., 2008, 'Comparative architectural characterization of SPEC CPU2000 and CPU2006 benchmarks on the intel® Core™ 2 Duo processor.' *In Proceedings of the International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS 2008)*, Samos, Greece, pp. 132 - 141.
- Korkhov, V. V., Moscicki, J. T. & Krzhizhanovskaya, V. V., 2009, 'The User-Level Scheduling of Divisible Load Parallel Applications With Resource Selection and

- Adaptive Workload Balancing on the Grid.' *IEEE Systems Journal*, Vol. 3, No. 1, pp. 121 - 130.
- Kunrath, L., Westphall, C. B. & Koch, F. L., 2008, 'Towards Advance Reservation in Large-Scale Grids.' *In Proceedings of the Third International Conference on Systems (ICONS 08)*, Cancun, Mexico, pp. 247 - 252.
- Lacour, S., Perez, C. & Priol, T., 2005, 'Generic application description model: toward automatic deployment of applications on computational grids.' *In Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing*, Seattle, Washington, pp. 284 - 287.
- Lee, Y. C., Subrata, R. & Zomaya, A. Y., 2009, 'On the Performance of a Dual-Objective Optimization Model for Workflow Applications on Grid Platforms.' *IEEE Transactions on Parallel and Distributed Systems*, Vol. 20, No. 9, pp. 1273 - 1284.
- Lee, Y. C., Wang, C., Zomaya, A. Y. & Zhou, B. B., 2010, 'Profit-Driven Service Request Scheduling in Clouds.' *In Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid)*, Melbourne, Australia, pp. 15 - 24.
- Lee, Y. C. & Zomaya, A. Y., 2007, 'Practical Scheduling of Bag-of-Tasks Applications on Grids with Dynamic Resilience.' *IEEE Transactions on Computers*, Vol. 56, No. 6, pp. 815 - 825.
- Li, J., Chinneck, J., Woodside, M., Litoiu, M. & Iszlai, G., 2009a, 'Performance model driven QoS guarantees and optimization in clouds.' *In Proceedings of the ICSE Workshop on Software Engineering Challenges of Cloud Computing (CLOUD '09)*, Vancouver, Canada, pp. 15 - 22.
- Li, L., 2009, 'An Optimistic Differentiated Service Job Scheduling System for Cloud Computing Service Users and Providers.' *In Proceedings of the Third International Conference on Multimedia and Ubiquitous Engineering (MUE '09)*, Qingdao, China, pp. 295 - 299.
- Li, M., Yu, B., Rana, O. & Wang, Z., 2008, 'Grid Service Discovery with Rough Sets.' *IEEE Transactions on Knowledge and Data Engineering*, Vol. 20, No. 6, pp. 851 - 862.
- Li, S., Qiao, L., Tang, Z., Cheng, B. & Gao, X., 2009b, 'Performance Characterization of SPEC CPU2006 Benchmarks on Intel and AMD Platform.' *In Proceedings of the First International Workshop on Education Technology and Computer Science (ETCS '09)*, Wuhan, Hubei, pp. 116 - 121.
- Lizhe, W., von Laszewski, G., Jie, T. & Kunze, M., 2009, 'Grid Virtualization Engine: Design, Implementation, and Evaluation.' *IEEE Systems Journal*, Vol. 3, No. 4, pp. 477 - 488.
- Lo, C.-T. D. & Qian, K., 2010, 'Green Computing Methodology for Next Generation Computing Scientists.' *In Proceedings of the Computer Software and Applications Conference (COMPSAC), 2010 IEEE 34th Annual*, pp. 250-251.
- Lu, B. & Ma, J., 2009, 'A Strategy for Resource Allocation and Pricing in Grid Environment Based on Economic Model.' *In Proceedings of the International Conference on Computational Intelligence and Natural Computing (CINC '09)*, Wuhan, China, pp. 221 - 224.
- M.Nithya & Banu, R. S. D. W., 2010, 'Towards Novel And Efficient Security Architecture For Role-Based Access Control In Grid Computing.' *International Journal of Computer Science and Information Technologies*, Vol. 9, No. 3, pp. 12 - 19.

- Machtans, E. C., Sato, L. M. & Deppman, A., 2009, 'Improvement on Scheduling Dependent Tasks for Grid Applications.' *In Proceedings of the International Conference on Computational Science and Engineering (CSE '09)*, Vancouver, Canada, pp. 95 - 102.
- Maci, M., Fito, J. O. & Guitart, J., 2010, 'Rule-based SLA management for revenue maximisation in Cloud Computing Markets.' *In Proceedings of the International Conference on Network and Service Management (CNSM)*, Niagara Falls, pp. 354 - 357.
- Mansukhani, M., 2005, 'Service Oriented Architecture White Paper', HP Inc.
- Martinovic, G., Balen, J. & Rimac-Drlje, S., 2010, 'Impact of the host operating systems on virtual machine performance.' *In Proceedings of the 33rd International Convention MIPRO*, Croatia, pp. 613 - 618.
- Mei, L., Chan, W. K. & Tse, T. H., 2008, 'A Tale of Clouds: Paradigm Comparisons and Some Thoughts on Research Issues.' *In Proceedings of the IEEE Asia-Pacific Services Computing Conference (APSCC '08)*, Yilan, Taiwan, pp. 464 - 469.
- Moaddeli, H. R., Dastghaibiyfard, G. & Moosavi, M. R., 2008, 'Flexible Advance Reservation Impact on Backfilling Scheduling Strategies.' *In Proceedings of the Seventh International Conference on Grid and Cooperative Computing (GCC '08)*, Shenzhen, China, pp. 151 - 159.
- Moltó, G., Hernández, V. & Alonso, J. M., 2008, 'A service-oriented WSRF-based architecture for metascheduling on computational Grids.' *Future Generation Computer Systems*, Vol. 24, No. 4, pp. 317 - 328.
- Netto, M. & Buyya, R., 2008, 'Rescheduling co-allocation requests based on flexible advance reservations and processor remapping.' *In Proceedings of the 9th IEEE/ACM International Conference on Grid Computing*, Tsukuba, Japan, pp. 144 - 151.
- Overhage, S. & Thomas, P., 2003, 'WS-Specification: Specifying Web Services Using UDDI Improvements.' *In Proceedings of the Web and database-related workshops*, Erfurt, Germany, pp. 100 - 119.
- PassMark, 2010, 'PerformanceTest 7.0', Available from: <http://www.passmark.com/products/pt.htm>, [Accessed September 2010].
- Plale, B., Jacobs, C., Jensen, S., Liu, Y., Moad, C., Parab, R. & Vaidya, P. 2004. Understanding Grid resource information management through a synthetic database benchmark/workload. *Proceedings of the 2004 IEEE International Symposium on Cluster Computing and the Grid*. IEEE Computer Society.
- Prodan, R. & Wiczorek, M., 2010, 'Bi-Criteria Scheduling of Scientific Grid Workflows.' *IEEE Transactions on Automation Science and Engineering*, Vol. 7, No. 2, pp. 364 - 376.
- Rajah, K., Ranka, S. & Ye, X., 2009, 'Advance Reservations and Scheduling for Bulk Transfers in Research Networks.' *IEEE Transactions on Parallel and Distributed Systems*, Vol. 20, No. 11, pp. 1682 - 1697.
- Ranaldo, N. & Zimeo, E., 2009, 'Time and Cost-Driven Scheduling of Data Parallel Tasks in Grid Workflows.' *IEEE Systems Journal*, Vol. 3, No. 1, pp. 104 - 120.
- Ranjan, R., Harwood, A. & Buyya, R., 2008, 'Peer-to-peer-based resource discovery in global grids: a tutorial.' *IEEE Communications Surveys & Tutorials*, Vol. 10, No. 2, pp. 6 - 33.

- Rasooli, A., Mirza-Aghatabar, M. & Khorsandi, S., 2008, 'Introduction of novel dispatching rules for grid scheduling algorithms.' *In Proceedings of the International Conference on Computer and Communication Engineering (ICCCE 2008)*, Kuala Lumpur, Malaysia, pp. 1072 - 1078.
- Reed, D. A. & Mendes, C. L., 2005, 'Intelligent Monitoring for Adaptation in Grid Applications.' *Proceedings of the IEEE*, Vol. 93, No. 2, pp. 426 - 435.
- Reig, G., Alonso, J. & Guitart, J., 2010, 'Prediction of Job Resource Requirements for Deadline Schedulers to Manage High-Level SLAs on the Cloud.' *In Proceedings of the 9th IEEE International Symposium on Network Computing and Applications (NCA)*, Cambridge, pp. 162 - 167.
- Sadhasivam, S., Nagaveni, N., Jayarani, R. & Ram, R. V., 2009, 'Design and Implementation of an Efficient Two-level Scheduler for Cloud Computing Environment.' *In Proceedings of the International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom '09)*, Kerala, India, pp. 884 - 886.
- Salesforce.com, 2010, 'Salesforce.com', Available from: <http://www.salesforce.com/>, [Accessed June 2010].
- Schopf, J. M., 2004, 'Ten actions when Grid scheduling: the user as a Grid scheduler.', in Nabrzyski, J., Schopf, J. M. & Weglarz, J., *Grid Resource Management: State of the Art and Future Trends*, Kluwer Academic Publishers, Norwell, USA.
- Singh, G., Kesselman, C. & Deelman, E., 2009, 'An End-to-End Framework for Provisioning-Based Resource and Application Management.' *IEEE Systems Journal*, Vol. 3, No. 1, pp. 25 - 48.
- Smith, M., Friese, T. & Freisleben, B., 2004, 'Towards a service-oriented ad hoc grid.' *In Proceedings of the Third International Symposium on Parallel and Distributed Computing*, Cork, Ireland, pp. 201 - 208.
- Snaveley, A., Chun, G., Casanova, H., Wijngaart, R. F. V. d. & Frumkin, M. A., 2003, 'Benchmarks for grid computing: a review of ongoing efforts and future directions.' *ACM SIGMETRICS Performance Evaluation Review*, Vol. 30, No. 4, pp. 27 - 32.
- Sotomayor, B., Montero, R. S., Llorente, I. M. & Foster, I., 2009, 'Virtual Infrastructure Management in Private and Hybrid Clouds.' *IEEE Internet Computing*, Vol. 13, No. 5, pp. 14 - 22.
- SPEC, 2010, 'CPU2006', Available from: <http://www.spec.org/cpu2006/>, [Accessed September 2010].
- Sridhar, T., 2009, 'Cloud Computing - A Primer.' *The Internet Protocol Journal*, Vol. 12, No. 3, pp. 2 - 9.
- Stanoevska-slabeva, K., Wozniak, T. & Ristol, S., 2009, *Grid And Cloud Computing: A Business Perspective On Technology And Applications*, Springer, Madrid, Spain.
- Subrata, R., Zomaya, A. Y. & Landfeldt, B., 2008, 'A Cooperative Game Framework for QoS Guided Job Allocation Schemes in Grids.' *IEEE Transactions on Computers*, Vol. 57, No. 10, pp. 1413 - 1422.
- Sun, C., He, L., Wang, Q. & Willenborg, R., 2008, 'Simplifying Service Deployment with Virtual Appliances.' *In Proceedings of the IEEE International Conference on Services Computing (SCC '08)*, Honolulu, HI, pp. 265 - 272.
- Takabi, H., Joshi, J. B. D. & Ahn, G., 2010, 'Security and Privacy Challenges in Cloud Computing Environments.' *IEEE Security & Privacy*, Vol. 8, No. 6, pp. 24 - 31.

- Truong, H.-L. & Dustdar, S., 2010, 'Composable cost estimation and monitoring for computational applications in cloud computing environments.' *Procedia Computer Science*, Vol. 1, No. 1, pp. 2175 - 2184.
- Tsouloupas, G. & Dikaiakos, M. D., 2007, 'GridBench: A tool for the interactive performance exploration of Grid infrastructures.' *Journal of Parallel and Distributed Computing*, Vol. 67, No. 9, pp. 1029 - 1045.
- Vahdat-Nejad, H. & Monsefi, R., 2008, 'Static Parallel Job Scheduling in Computational Grids.' *In Proceedings of the International Conference on Computer and Electrical Engineering (ICCEE 2008)*, pp. 548 - 552.
- Vecchiola, C., Chua, X. & Buyya, R., 2009, 'Aneka: A Software Platform for .NET-based Cloud Computing.' *Distributed, Parallel, and Cluster Computing*, Vol. No. pp. 4622 - 4651.
- Viswanathan, S., Veeravalli, B. & Robertazzi, T. G., 2007, 'Resource-Aware Distributed Scheduling Strategies for Large-Scale Computational Cluster/Grid Systems.' *IEEE Transactions on Parallel and Distributed Systems*, Vol. 18, No. 10, pp. 1450 - 1461.
- Vouk, M. A., 2008, 'Cloud computing - Issues, research and implementations.' *In Proceedings of the 30th International Conference on Information Technology Interfaces (ITI 2008)*, pp. 31 - 40.
- Wang, Y.-H. & Liao, J. C., 2009, 'Why or Why Not Service Oriented Architecture.' *In Proceedings of the IITA International Conference on Services Science, Management and Engineering (SSME '09)*, Beijing, China, pp. 65 - 68.
- Weicker, R., 2002, 'Benchmarking.' *In Proceedings of the Performance Evaluation of Complex Systems: Techniques and Tools, Performance*, pp. 179 - 207.
- Windows, 2010, 'Windows Azure', Available from: <http://www.microsoft.com/windowsazure/windowsazure/>, [Accessed June 2010].
- Wohl, A., 2008, *Succeeding at SaaS: Computing in the Cloud*, Wohl Associates, United States of America.
- Wu, H., Lee, C.-Y., Chen, W.-Y. & Lee, T.-Y., 2007, 'A Job Schedule Model Based on Grid Environment.' *In Proceedings of the First International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2007)*, Vienna, Austria, pp. 43 - 52.
- Xu, M., Cui, L., Wang, H. & Bi, Y., 2009, 'A Multiple QoS Constrained Scheduling Strategy of Multiple Workflows for Cloud Computing.' *In Proceedings of the IEEE International Symposium on Parallel and Distributed Processing with Applications*, Chengdu, China, pp. 629 - 634.
- Yeo, C. S., Venugopal, S., Chu, X. & Buyya, R., 2010, 'Autonomic metered pricing for a utility computing service.' *Future Generation Computer Systems*, Vol. 26, No. 8, pp. 1368 - 1380.
- Zhang, L.-J. & Zhou, Q., 2009, 'CCOA: Cloud Computing Open Architecture.' *In Proceedings of the IEEE International Conference on Web Services (ICWS 2009)*, Los Angeles, CA, pp. 607 - 616.
- Zhang, X., Dwarkadas, S., Folkmanis, G. & Shen, K., 2007, 'Processor hardware counter statistics as a first-class system resource.' *In Proceedings of the 11th USENIX workshop on Hot topics in operating systems*, San Diego, CA, pp. 1 - 6.
- Zhang, Y., Koelbel, C. & Cooper, K., 2009, 'Hybrid Re-scheduling Mechanisms for Workflow Applications on Multi-cluster Grid.' *In Proceedings of the 9th*

- IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID '09)*, Shanghai, China, pp. 116 - 123.
- Zhu, Y., 2003, 'A survey on grid scheduling systems', *Technical Report*, Computer Science Department of Hong Kong University of Science and Technology.
- Zhu, Y. & Guo, X., 2009, 'Grid Dependent Tasks Scheduling Based on Hybrid Adaptive Genetic Algorithm.' *In Proceedings of the WRI Global Congress on Intelligent Systems, (GCIS '09)*, pp. 35 - 38.
- Zikos, S. & Karatza, H. D., 2008, 'Resource Allocation Strategies in a 2-Level Hierarchical Grid System.' *In Proceedings of the 41st Annual Simulation Symposium, (ANSS 2008)*, Ottawa, Canada, pp. 157 - 164.

APPENDIX A TEST CASES

This section describes the tests cases carried out in order to confirm the implementation correctness of the proposed algorithm in Grid and Cloud testbed. The first test case is used to verify the correctness of HSA and automatic deployment mechanism. The second test case is used to verify the correctness of ASA with benchmarking. The final test case is used to verify the correctness of AQoSSA with rescheduling.

A.1 Test Case 1: HSA and Automatic Deployment

This section describes the verifications performed on the HSA and automatic deployment mechanism to validate the correctness of implementation. Table A.1 shows the resources used for the testing and Table A.2 shows the parameter values of different tasks at different time interval. The MIN-MIN, MAX-MIN and HSA are then tested using the configuration below.

Table A.1 Resources Information

Resource ID	Type of Resource	Category	Number of CPU	Core per CPU	CPU Speed (GHz)
29	Server	Primary	2	4	1.86
33	Server	Primary	2	4	2.67
41	Server	Primary	1	2	2.33
37	Cluster	Primary	4	4	2.33
38	Cluster	Primary	4	4	2.33
31	Server (Virtual)	Primary	2	4	1.86
35	Server (Virtual)	Primary	2	4	2.67
1	Desktop Computer	Secondary	1	1	2.40
2	Desktop Computer	Secondary	1	1	2.40
8	Desktop Computer	Secondary	1	1	2.60
9	Desktop Computer	Secondary	1	1	2.60
10	Desktop Computer	Secondary	1	1	2.60

Table A.2 Tasks Information

Task ID	Interval	Web Services	Parameter Value
1	T0	W1	34300000
2	T0	W1	83300000
3	T0	W1	57800000
4	T0	W2	7500
5	T0	W2	5000
6	T0	W2	2300
7	T0	W3	158000
8	T0	W3	234000
9	T1	W1	22700000
10	T1	W1	73800000
11	T1	W1	54300000
12	T1	W2	3800
13	T1	W2	5100
14	T1	W2	4000
15	T1	W2	8100
16	T1	W2	7300
17	T1	W3	246000
18	T1	W3	132000
19	T1	W3	222000
20	T2	W1	80300000
21	T2	W1	2400000
22	T2	W1	86800000
23	T2	W1	6800000
24	T2	W2	3100
25	T2	W2	500
26	T2	W2	7300
27	T3	W1	81100000
28	T3	W1	52000000
29	T3	W2	7700
30	T3	W2	2700
31	T3	W3	240000
32	T3	W3	57000
33	T3	W3	8000
34	T4	W1	44000000
35	T4	W1	7300000
36	T4	W1	26000000
37	T4	W3	214000
38	T4	W3	83000
39	T4	W3	79000

40	T4	W3	157000
----	----	----	--------

Table A.3 shows the results of the MIN-MIN scheduling algorithm. The MIN-MIN scheduling algorithm classifies all the resources as primary resources. The duration of the task is the task execution time running on the resource while the completion time is the task execution time plus task waiting time. At interval T0, the MIN-MIN scheduling algorithm sorts the tasks from the smallest to the largest parameter values. Then, for Web Services W1, Task 1 is executed first followed by Task 3 and then Task 2. The tasks are assigned to the resources that return the minimum completion time. These steps are repeated for all the Web Services in the same interval. The results on Table A.3 clearly show that the implementation is correct. The interval used for the experiment is 20 seconds. The makespan for MIN-MIN scheduling algorithm is 1773.72 seconds.

Table A.3 Task scheduling with MIN-MIN (Parameter Value)

Interval	Task ID	Web Services	Resource ID	Duration	Completion Time
T0	1	W1	29	50.04	50.04
T0	3	W1	33	107.12	107.12
T0	2	W1	29	121.52	171.56
T0	6	W2	33	220.97	328.09
T0	5	W2	31	574.19	574.19
T0	4	W2	29	681.16	852.72
T0	7	W3	37	220.79	220.79
T0	8	W3	38	253.01	253.01
T1	9	W1	41	82.04	82.04
T1	11	W1	35	235.64	235.64
T1	10	W1	41	266.70	348.74
T1	12	W2	33	364.67	672.76
T1	14	W2	35	518.83	754.47
T1	13	W2	37	707.79	908.58
T1	16	W2	38	888.33	1121.34
T1	15	W2	31	855.63	1409.82
T1	18	W3	41	475.40	824.14

T1	19	W3	33	423.24	1096.00
T1	17	W3	37	258.09	1166.68
T2	21	W1	8	19.94	19.94
T2	23	W1	9	56.51	56.51
T2	20	W1	10	667.28	667.28
T2	22	W1	8	721.29	741.23
T2	25	W2	1	226.11	226.11
T2	24	W2	29	281.69	1094.41
T2	26	W2	35	834.99	1569.46
T3	28	W1	9	432.11	468.61
T3	27	W1	2	725.37	725.37
T3	30	W2	41	487.19	1271.33
T3	29	W2	29	699.31	1773.72
T3	33	W3	1	76.09	282.20
T3	32	W3	1	530.20	812.40
T3	31	W3	38	255.55	1336.89
T4	35	W1	9	60.66	509.27
T4	36	W1	9	216.05	725.32
T4	34	W1	10	365.63	992.91
T4	39	W3	33	150.96	1186.96
T4	38	W3	37	189.00	1295.68
T4	40	W3	33	299.47	1486.43
T4	37	W3	37	244.53	1540.21

Table A.4 shows the result of the MAX-MIN scheduling algorithm. The MAX-MIN scheduling algorithm classifies all the resources as primary resources. At interval T0, the MAX-MIN scheduling algorithm sorts the tasks according to the parameter values from the largest to the smallest. Hence, for Web Services W1, Task 2 is executed first followed by Task 3 and then Task 1. Then, the task is assigned to the resource that returns the minimum completion time. The step is repeated for all the Web Services in the same interval. It is shown clearly from the Table A.4 that the implementation is correct. The interval used for the experiment is 20 seconds. The makespan for MAX-MIN is 1821.95 seconds.

Table A.4 Task scheduling with MAX-MIN (Parameter Value)

Interval	Task ID	Web Services	Resource ID	Duration	Completion Time
T0	2	W1	29	121.52	121.52
T0	3	W1	33	107.12	107.12
T0	1	W1	37	101.37	101.37
T0	4	W2	29	681.16	802.68
T0	5	W2	33	479.64	586.76
T0	6	W2	31	329.06	329.06
T0	8	W3	38	253.01	253.01
T0	7	W3	37	220.79	322.16
T1	10	W1	41	266.70	266.70
T1	11	W1	35	235.64	235.64
T1	9	W1	8	188.63	188.63
T1	15	W2	31	855.63	1164.69
T1	16	W2	35	834.99	1070.63
T1	13	W2	38	707.79	940.80
T1	14	W2	37	617.52	919.68
T1	12	W2	33	364.67	931.43
T1	17	W3	37	258.09	1177.78
T1	19	W3	41	799.63	1066.33
T1	18	W3	29	230.91	1013.59
T2	22	W1	9	721.29	721.29
T2	20	W1	10	667.28	667.28
T2	23	W1	1	60.82	60.82
T2	21	W1	2	21.47	21.47
T2	26	W2	33	699.99	1611.42
T2	24	W2	29	281.69	1275.28
T2	25	W2	2	226.11	247.58
T3	27	W1	1	725.37	766.19
T3	28	W1	8	432.11	580.74
T3	29	W2	38	921.15	1821.95
T3	30	W2	35	394.29	1424.92
T3	31	W3	37	255.55	1393.33
T3	32	W3	2	530.20	757.78
T3	33	W3	8	71.59	552.34
T4	34	W1	8	365.63	897.97
T4	36	W1	10	216.05	843.33
T4	35	W1	9	60.66	741.95
T4	37	W3	29	374.36	1609.64
T4	40	W3	31	394.64	1499.33
T4	38	W3	41	298.87	1305.20
T4	39	W3	9	685.00	1426.95

Table A.5 shows the result of the HSA and automatic deployment mechanism. The HSA classifies the resources into primary resources and secondary resources. At interval T0, the HSA sorts the tasks according to the parameter values from the largest to the smallest where Task 2 executes first followed by Task 3 and Task 1 for Web Services W1. Then, the mean parameter is computed using the equation 3.7 and a predefined z value is used for the experiment. The mean value and the z value are used as the threshold to decide whether the task is scheduled to the primary or secondary resource. Finally, the task is assigned to the resource within the group that returns the minimum completion time. The step is repeated for all the Web Services in the same interval.

From Table A.4, Task 1 is assigned to the resource 37 in MAX-MIN but from Table A.5, Task 1 is assigned to the secondary resource 10. This is due to the mean value and the z value chosen. The Table A.5 shows that the implementation is correct. The interval used for the experiment is 20 seconds and the predefined z value is 0.6. The makespan for HSA and automatic deployment mechanism is 1789.98 seconds.

Table A.5 Task scheduling with HSA and Automatic Deployment

Interval	Task ID	Web Services	Resource ID	Duration	Completion Time
T0	2	W1	29	121.52	121.52
T0	3	W1	33	107.12	107.12
T0	1	W1	10	285.03	285.03
T0	4	W2	29	681.16	802.68
T0	5	W2	33	479.64	586.76
T0	6	W2	9	955.67	955.67
T0	8	W3	38	253.01	253.01
T0	7	W3	37	220.79	220.79
T1	10	W1	31	227.67	227.67
T1	11	W1	41	196.23	196.23

T1	9	W1	8	188.63	188.63
T1	15	W2	35	911.63	911.63
T1	16	W2	31	783.00	1010.67
T1	13	W2	37	707.79	908.58
T1	14	W2	38	617.52	850.53
T1	12	W2	41	685.14	881.37
T1	17	W3	33	468.93	1035.69
T1	19	W3	38	247.92	1098.45
T1	18	W3	29	230.91	1013.59
T2	22	W1	29	126.63	1120.22
T2	20	W1	33	148.82	1164.52
T2	23	W1	2	60.82	60.82
T2	21	W1	1	21.47	21.47
T2	26	W2	29	663.00	1783.22
T2	24	W2	35	432.61	1324.24
T2	25	W2	1	226.11	247.58
T3	27	W1	37	239.68	1108.26
T3	28	W1	41	187.92	1029.29
T3	29	W2	31	819.31	1789.98
T3	30	W2	8	1121.83	1270.46
T3	31	W3	38	255.55	1314.00
T3	32	W3	2	530.20	571.02
T3	33	W3	10	71.59	296.62
T4	34	W1	33	81.55	1206.07
T4	36	W1	41	93.96	1103.25
T4	35	W1	10	60.66	337.28
T4	37	W3	37	244.53	1332.79
T4	40	W3	33	299.47	1423.99
T4	38	W3	41	298.87	1402.12
T4	39	W3	1	734.08	941.66

A.2 Test Case 2: ASA with Benchmarking

This section describes the verification performed to validate the implementation correctness of ASA with benchmarking. Table A.6 shows the resources used for the testing and Table A.7 shows the parameter values of different tasks in different time interval. The MIN-MIN, MAX-MIN and ASA are then tested based on the configuration below.

Table A.6 Resources Information

Resource ID	Type of Resource	Category	Number of CPU	Core per CPU	CPU Speed (GHz)
29	Server	Primary	2	4	1.86
33	Server	Primary	2	4	2.67
41	Server	Primary	1	2	2.33
37	Cluster	Primary	4	4	2.33
38	Cluster	Primary	4	4	2.33
31	Server (Virtual)	Primary	2	4	1.86
35	Server (Virtual)	Primary	2	4	2.67
1	Desktop Computer	Secondary	1	1	2.40
2	Desktop Computer	Secondary	1	1	2.40
8	Desktop Computer	Secondary	1	1	2.60
9	Desktop Computer	Secondary	1	1	2.60
10	Desktop Computer	Secondary	1	1	2.60

Table A.7 Tasks Information

Task	Interval	Web Services	Parameter Value
1	T0	W1	34300000
2	T0	W1	83300000
3	T0	W1	57800000
4	T0	W2	7500
5	T0	W2	5000
6	T0	W2	2300
7	T0	W3	158000
8	T0	W3	234000
9	T1	W1	22700000
10	T1	W1	73800000
11	T1	W1	54300000
12	T1	W2	3800
13	T1	W2	5100
14	T1	W2	4000
15	T1	W2	8100
16	T1	W2	7300
17	T1	W3	246000
18	T1	W3	132000
19	T1	W3	222000
20	T2	W1	80300000
21	T2	W1	2400000
22	T2	W1	86800000

23	T2	W1	6800000
24	T2	W2	3100
25	T2	W2	500
26	T2	W2	7300
27	T3	W1	81100000
28	T3	W1	52000000
29	T3	W2	7700
30	T3	W2	2700
31	T3	W3	240000
32	T3	W3	57000
33	T3	W3	8000
34	T4	W1	44000000
35	T4	W1	7300000
36	T4	W1	26000000
37	T4	W3	214000
38	T4	W3	83000
39	T4	W3	79000
40	T4	W3	157000

Table A.8 shows the result of the MIN-MIN scheduling algorithm with benchmarking. At interval T0, the algorithm sorts the tasks according to the job length estimated from benchmarking from the lowest to the highest. The sorting is performed on all the Web Services at the same interval. From Table A.8, Task 1 executes first followed by Task 3, 2, 6, 7, 8, 5 and 4. Then, each task is assigned to the resource that returns the minimum completion time. It is shown from Table A.8 that the implementation is correct. The interval used for the experiment is 20 seconds. The makespan for MIN-MIN is 1773.73 seconds.

Table A.8 Task scheduling with MIN-MIN (Benchmarking)

Interval	Task ID	Web Services	Resource ID	Duration	Completion Time
T0	1	W1	29	50.04	50.04
T0	3	W1	33	107.12	107.12
T0	2	W1	29	121.52	171.56
T0	6	W2	33	220.97	328.09

T0	7	W3	38	220.79	220.79
T0	8	W3	37	253.01	253.01
T0	5	W2	31	574.19	574.19
T0	4	W2	29	681.16	852.72
T1	9	W1	41	82.04	82.04
T1	11	W1	35	235.64	235.64
T1	10	W1	41	266.7	348.74
T1	18	W3	38	209.77	410.56
T1	12	W2	33	364.67	672.76
T1	14	W2	35	518.83	754.47
T1	19	W3	37	247.92	480.93
T1	17	W3	38	258.09	668.66
T1	13	W2	31	583.27	1137.45
T1	16	W2	37	888.33	1369.25
T1	15	W2	33	776.63	1449.4
T2	21	W1	10	19.94	19.94
T2	23	W1	9	56.51	56.51
T2	25	W2	8	208.00	208.00
T2	20	W1	41	290.19	618.93
T2	22	W1	10	721.29	741.23
T2	24	W2	29	281.69	1094.41
T2	26	W2	38	888.33	1536.98
T3	33	W3	2	76.09	76.09
T3	28	W1	1	465.10	465.10
T3	32	W3	9	494.93	531.44
T3	27	W1	2	725.37	801.47
T3	30	W2	41	487.19	1086.12
T3	31	W3	35	592.51	1306.98
T3	29	W2	29	699.31	1773.73
T4	35	W1	8	60.66	228.66
T4	36	W1	8	216.05	444.71
T4	34	W1	8	365.63	810.34
T4	39	W3	1	734.08	1179.18
T4	38	W3	9	719.56	1231.00
T4	40	W3	31	394.64	1472.09
T4	37	W3	37	244.53	1553.78

Table A.9 shows the result of the MAX-MIN scheduling algorithm with benchmarking. At interval T0, the algorithm sorts the tasks according to the job length estimated from benchmarking from the highest to the lowest. The sorting is performed on all the Web

Services at the same interval. From Table A.9, Task 4 executes first followed by Task 5, 8, 7, 6, 2, 3 and 1. Then, each task is assigned to the resource that returns the minimum completion time. It is shown from the Table A.9 that the implementation is correct. The interval used for the experiment is 20 seconds. The makespan for MAX-MIN is 1620.17 seconds.

Table A.9 Task scheduling with MAX-MIN (Benchmarking)

Interval	Task ID	Web Services	Resource ID	Duration	Completion Time
T0	4	W2	29	681.16	681.16
T0	5	W2	33	479.64	479.64
T0	8	W3	38	253.01	253.01
T0	7	W3	37	220.79	220.79
T0	6	W2	31	329.06	329.06
T0	2	W1	35	289.38	289.38
T0	3	W1	41	208.88	208.88
T0	1	W1	10	285.03	285.03
T1	15	W2	37	953.97	1154.77
T1	16	W2	31	783.00	1092.06
T1	13	W2	35	624.22	893.60
T1	17	W3	38	258.09	491.10
T1	19	W3	38	247.92	739.02
T1	14	W2	33	383.83	843.47
T1	12	W2	41	685.14	874.02
T1	18	W3	29	230.91	892.06
T1	10	W1	9	613.26	613.26
T1	11	W1	8	451.22	451.22
T1	9	W1	2	203.03	203.03
T2	26	W2	33	699.99	1523.46
T2	24	W2	29	281.69	1153.75
T2	22	W1	1	776.35	776.35
T2	20	W1	2	718.22	901.25
T2	25	W2	10	208.00	453.02
T2	23	W1	8	56.51	487.73
T2	21	W1	10	19.94	472.97
T3	29	W2	38	921.15	1620.17
T3	31	W3	37	255.55	1370.32
T3	30	W2	35	394.29	1247.89
T3	27	W1	10	673.92	1126.89
T3	32	W3	8	494.93	962.66

T3	28	W1	9	432.11	1005.37
T3	33	W3	1	76.09	832.45
T4	37	W3	29	374.36	1488.11
T4	40	W3	41	565.46	1379.48
T4	38	W3	31	265.18	1297.25
T4	39	W3	35	285.96	1513.85
T4	34	W1	1	393.54	1205.99
T4	36	W1	2	232.55	1093.80
T4	35	W1	8	60.66	1003.32

Table A.10 shows the result of the HSA with benchmarking. At interval T0, the algorithm sorts the tasks according to the job length estimated from benchmarking from the highest to the lowest. From Table A.10, Task 4 executes first followed by Task 5, 8, 7, 6, 2, 3 and 1. Then, the mean parameter is computed using the equation 4.3 and a predefined z value is used for the experiment. The mean value and the z value are used as the threshold to decide whether the task is scheduled to the primary or secondary resource. Finally, the task is assigned to the resource within the group that returns the minimum completion time.

From Table A.9, Task 6, 2, and 3 are assigned to resource 31, 35 and 41 respectively but from Table 6.10, Task 6, 2, and 3 are assigned to resource 10, 9 and 8 respectively which are the secondary resources. This is due to the mean value and the z value. The Table A.10 shows that the implementation is correct. The interval used for the experiment is 20 seconds and the predefined z value is 0.6. The makespan for HSA with benchmarking is 1671.08 seconds.

Table A.10 Task scheduling with HSA (Benchmarking)

Interval	Task ID	Web Services	Resource ID	Duration	Completion Time
T0	4	W2	29	681.16	681.16
T0	5	W2	33	479.64	479.64
T0	8	W3	38	253.01	253.01

T0	7	W3	37	220.79	220.79
T0	6	W2	10	955.67	955.67
T0	2	W1	9	692.20	692.20
T0	3	W1	8	480.31	480.31
T0	1	W1	2	306.79	306.79
T1	15	W2	31	855.63	855.63
T1	16	W2	35	834.99	834.99
T1	13	W2	37	707.79	908.58
T1	17	W3	38	258.09	491.10
T1	19	W3	38	247.92	739.02
T1	14	W2	41	721.13	721.13
T1	12	W2	33	364.67	824.31
T1	18	W3	1	1225.26	1225.26
T1	10	W1	2	660.08	946.87
T1	11	W1	8	451.22	911.53
T1	9	W1	9	188.63	860.84
T2	26	W2	29	663.00	1304.15
T2	24	W2	33	297.61	1101.92
T2	22	W1	38	256.52	975.54
T2	20	W1	9	667.28	1508.11
T2	25	W2	8	208.00	1099.52
T2	23	W1	10	56.51	972.18
T2	21	W1	2	21.47	948.33
T3	29	W2	31	819.31	1634.94
T3	31	W3	37	255.55	1124.13
T3	30	W2	41	487.19	1168.31
T3	27	W1	10	673.92	1626.10
T3	32	W3	2	530.20	1458.53
T3	28	W1	8	432.11	1511.63
T3	33	W3	1	76.09	1261.35
T4	37	W3	38	244.53	1180.07
T4	40	W3	35	434.47	1209.46
T4	38	W3	33	158.57	1220.49
T4	39	W3	37	187.31	1291.44
T4	34	W1	1	393.54	1634.90
T4	36	W1	2	232.55	1671.08
T4	35	W1	9	60.66	1528.77

Table A.11 shows the result of the ASA. ASA enhances the HSA by introduce the dynamic z value. From Table A.10, Task 6, 2, 3 and 1 are assigned to secondary resources but from Table A.11, only Task 3 and 1 are assigned to secondary resources.

The Table A.11 shows that the implementation is correct. The ASA sorts the tasks according to the job length estimated from benchmarking from the highest to the lowest. Then, the mean value is computed using the equation 4.3 and a dynamic z value is used for the experiment. The mean value and a dynamic z value are then used as the threshold at each interval to decide whether the task is scheduled to the primary or secondary resource. On completion, the task is assigned to the resource within the group that returns the minimum completion time. The interval used for the experiment is 20 seconds. The makespan for ASA is 1620.17 seconds.

Table A.11 Task scheduling with ASA (Benchmarking)

Interval	Task ID	Web Services	Resource ID	Duration	Completion Time
T0	4	W2	29	681.16	681.16
T0	5	W2	33	479.64	479.64
T0	8	W3	38	253.01	253.01
T0	7	W3	37	220.79	220.79
T0	6	W2	31	329.06	329.06
T0	2	W1	35	289.38	289.38
T0	3	W1	10	480.31	480.31
T0	1	W1	9	285.03	285.03
T1	15	W2	37	953.97	1154.77
T1	16	W2	31	783.00	1092.06
T1	13	W2	35	624.22	893.60
T1	17	W3	38	258.09	491.10
T1	19	W3	38	247.92	739.02
T1	14	W2	41	721.13	721.13
T1	12	W2	33	364.67	824.31
T1	18	W3	8	1142.9	1142.9
T1	10	W1	2	660.08	660.08
T1	11	W1	1	485.67	485.67
T1	9	W1	9	188.63	453.66
T2	26	W2	29	663.00	1304.15
T2	24	W2	33	297.61	1101.92
T2	22	W1	9	721.29	1154.95
T2	20	W1	10	667.28	1107.58

T2	25	W2	1	226.11	691.78
T2	23	W1	2	60.82	700.90
T2	21	W1	1	21.47	713.25
T3	29	W2	38	921.15	1620.17
T3	31	W3	37	255.55	1370.32
T3	30	W2	41	487.19	1168.31
T3	27	W1	2	725.37	1406.27
T3	32	W3	1	530.20	1223.45
T3	28	W1	10	432.11	1519.69
T3	33	W3	8	71.59	1174.49
T4	37	W3	35	543.00	1376.61
T4	40	W3	33	299.47	1361.39
T4	38	W3	31	265.18	1297.25
T4	39	W3	29	138.19	1402.34
T4	34	W1	41	159.01	1307.33
T4	36	W1	9	216.05	1331.00
T4	35	W1	8	60.66	1215.15

A.3 Test Case 3: AQoSSA

This section describes the verification performed to validate the implementation correctness of AQoSSA. Table A.12 shows the resources used for the testing and Table A.13 lists the parameter values of different tasks at different interval time. Each task contains different Class of Service (CoS) and estimated deadline. The task with CoS equals to 1 will have the highest priority. The MIN-MIN QoS, MAX-MIN QoS and AQoSSA with rescheduling are then tested to verify the implementation. The acceptable percentiles for tasks with CoS of 1, 2 and 3 are 4%, 8% and 12% respectively before the SLA is violated.

Table A.12 Resources Information

Resource ID	Type of Resource	Category	Number of CPU	Core per CPU	CPU Speed (GHz)
29	Server	Primary	2	4	1.86
33	Server	Primary	2	4	2.67
41	Server	Primary	1	2	2.33
37	Cluster	Primary	4	4	2.33

38	Cluster	Primary	4	4	2.33
31	Server (Virtual)	Primary	2	4	1.86
35	Server (Virtual)	Primary	2	4	2.67
1	Desktop Computer	Secondary	1	1	2.40
2	Desktop Computer	Secondary	1	1	2.40
8	Desktop Computer	Secondary	1	1	2.60
9	Desktop Computer	Secondary	1	1	2.60
10	Desktop Computer	Secondary	1	1	2.60

Table A.13 Tasks Information

Task	Interval	Web Services	Class of Service	Parameter Value	Deadline
1	T0	W1	1	34300000	100
2	T0	W1	2	83300000	300
3	T0	W1	2	57800000	200
4	T0	W2	3	7500	2400
5	T0	W2	1	5000	450
6	T0	W2	1	2300	250
7	T0	W3	1	158000	300
8	T0	W3	2	234000	800
9	T1	W1	1	22700000	200
10	T1	W1	1	73800000	450
11	T1	W1	1	54300000	400
12	T1	W2	3	3800	1700
13	T1	W2	1	5100	700
14	T1	W2	3	4000	1800
15	T1	W2	1	8100	950
16	T1	W2	3	7300	2300
17	T1	W3	2	246000	900
18	T1	W3	2	132000	700
19	T1	W3	1	222000	450
20	T2	W1	1	80300000	700
21	T2	W1	1	2400000	100
22	T2	W1	2	86800000	800
23	T2	W1	3	6800000	100
24	T2	W2	1	3100	700
25	T2	W2	2	500	700
26	T2	W2	2	7300	1300
27	T3	W1	1	81100000	900
28	T3	W1	3	52000000	1100
29	T3	W2	2	7700	1400

30	T3	W2	1	2700	1250
31	T3	W3	3	240000	2200
32	T3	W3	1	57000	800
33	T3	W3	1	8000	700
34	T4	W1	2	44000000	1100
35	T4	W1	1	7300000	800
36	T4	W1	2	26000000	900
37	T4	W3	3	214000	2000
38	T4	W3	3	83000	1100
39	T4	W3	1	79000	1000
40	T4	W3	3	157000	1500

Table A.14 shows the results of the MIN-MIN QoS scheduling algorithm. At interval T0, the algorithm firsts sorts the tasks according to the CoS from the smallest to the largest and then followed by the job length estimates from benchmarking from the lowest to the highest. From Table A.14, Tasks 1, 7, 6 and 5 which has CoS value equal to 1 are executed first followed by tasks which have CoS value equal to 2 and 3. Once the task is sorted, the task is assigned to the resource that returns the minimum completion time.

Table A.14 also shows whether the task is completed or violated the SLA. If the completion time is more than the acceptable percentile of the deadline, the job is violated. From Table A.14, Task 5 is not completed because the completion time is more than the acceptable percentile of the deadline. The Table A.14 shows that the implementation is correct. The interval used for the experiment is 20 seconds. The total number of job completed is 32 and the reliability is 80%. The total profit gained is 524.30 units.

Table A.14 Task scheduling with MIN-MIN (QoS)

Interval	Web Services	CoS	Task ID	Resource ID	Duration	Completion Time	Deadline	Completed
T0	W1	1	1	29	50.04	50.04	100	Y
T0	W3	1	7	38	220.79	220.79	300	Y

T0	W2	1	6	33	220.97	220.97	250	Y
T0	W2	1	5	29	454.19	504.23	450	N
T0	W1	2	3	37	170.82	170.82	200	Y
T0	W1	2	2	31	241.52	241.52	300	Y
T0	W3	2	8	37	253.01	423.82	800	Y
T0	W2	3	4	35	854.15	854.15	2400	Y
T1	W1	1	9	41	82.04	82.04	200	Y
T1	W1	1	11	41	196.23	278.27	400	Y
T1	W1	1	10	33	136.78	337.74	450	Y
T1	W3	1	19	38	247.92	448.71	450	Y
T1	W2	1	13	31	583.27	804.79	700	N
T1	W2	1	15	33	776.63	1114.38	950	N
T1	W3	2	18	37	209.77	613.60	700	Y
T1	W3	2	17	38	258.09	706.81	900	Y
T1	W2	3	12	29	345.24	829.47	1700	Y
T1	W2	3	14	41	721.13	999.40	1800	Y
T1	W2	3	16	29	663.00	1492.47	2300	Y
T2	W1	1	21	10	19.94	19.94	100	Y
T2	W1	1	20	9	667.28	667.28	700	Y
T2	W2	1	24	37	543.67	1137.26	700	N
T2	W2	2	25	8	208.00	208.00	700	Y
T2	W1	2	22	10	721.29	741.23	800	Y
T2	W2	2	26	31	783.00	1567.79	1300	N
T2	W1	3	23	2	60.82	60.82	100	Y
T3	W3	1	33	1	76.09	76.09	700	Y
T3	W3	1	32	2	530.20	571.02	800	Y
T3	W1	1	27	1	725.37	801.47	900	Y
T3	W2	1	30	38	510.84	1177.65	1250	Y
T3	W2	2	29	35	873.31	1667.46	1400	N
T3	W1	3	28	8	432.11	620.11	1100	Y
T3	W3	3	31	37	255.55	1372.81	2200	Y
T4	W1	1	35	2	65.29	616.31	800	Y
T4	W3	1	39	33	150.96	1205.33	1000	N
T4	W1	2	36	8	216.05	816.16	900	Y
T4	W1	2	34	9	365.63	992.91	1100	Y
T4	W3	3	38	41	298.87	1238.27	1100	N
T4	W3	3	40	38	220.37	1378.02	1500	Y
T4	W3	3	37	37	244.53	1597.34	2000	Y

Table A.15 shows the result of the MAX-MIN QoS scheduling algorithm. At interval T0, the algorithm firsts sorts the tasks according to the CoS from the smallest to the largest and

then followed by the job length estimates from benchmarking from the highest to the lowest. From Table A.14, Task 5, 6, 7 and 1 which has CoS value equal to 1 are executed first followed by tasks which have CoS value equal to 2 and 3. Once the task is sorted, the task is assigned to the resource that returns the minimum completion time.

Table A.15 also shows whether the task is completed or violated the SLA. If the completion time is more than the acceptable percentile of the deadline, the job is violated. From Table A.15, Task 13 is not completed because the completion time is more than the acceptable percentile of the deadline. The Table A.15 shows that the implementation is correct. The interval used for the experiment is 20 seconds. The total number of jobs completed is 32 and the reliability is 80%. The total profit gained is 770.70 units.

Table A.15 Task scheduling with MAX-MIN (QoS)

Interval	Web Services	CoS	Task ID	Resource ID	Duration	Completion Time	Deadline	Complete
T0	W2	1	5	29	454.19	454.19	450	Y
T0	W2	1	6	33	220.97	220.97	250	Y
T0	W3	1	7	38	220.79	220.79	300	Y
T0	W1	1	1	37	101.37	101.37	100	Y
T0	W3	2	8	37	253.01	354.37	800	Y
T0	W1	2	2	31	241.52	241.52	300	Y
T0	W1	2	3	41	208.88	208.88	200	Y
T0	W2	3	4	35	854.15	854.15	2400	Y
T1	W2	1	15	33	776.63	977.60	950	Y
T1	W2	1	13	31	583.27	804.79	700	N
T1	W3	1	19	38	247.92	448.71	450	Y
T1	W1	1	10	41	266.70	455.59	450	Y
T1	W1	1	11	10	451.22	451.22	400	N
T1	W1	1	9	9	188.63	188.63	200	Y
T1	W3	2	17	37	258.09	592.47	900	Y
T1	W3	2	18	38	209.77	658.48	700	Y
T1	W2	3	16	29	663.00	1097.19	2300	Y
T1	W2	3	14	41	721.13	1176.71	1800	Y

T1	W2	3	12	37	601.11	1193.58	1700	Y
T2	W2	1	24	38	543.67	1182.15	700	N
T2	W1	1	20	8	667.28	667.28	700	Y
T2	W1	1	21	2	21.47	21.47	100	Y
T2	W2	2	26	31	783.00	1567.79	1300	N
T2	W1	2	22	1	776.35	776.35	800	Y
T2	W2	2	25	2	226.11	247.58	700	Y
T2	W1	3	23	9	56.51	225.14	100	N
T3	W2	1	30	35	394.29	1188.44	1250	Y
T3	W1	1	27	9	673.92	879.06	900	Y
T3	W3	1	32	2	530.20	757.78	800	Y
T3	W3	1	33	10	71.59	482.82	700	Y
T3	W2	2	29	33	738.31	1675.91	1400	N
T3	W3	3	31	37	255.55	1409.13	2200	Y
T3	W1	3	28	10	432.11	914.92	1100	Y
T4	W3	1	39	29	138.19	1175.37	1000	N
T4	W1	1	35	8	60.66	687.94	800	Y
T4	W1	2	34	8	365.63	1053.57	1100	Y
T4	W1	2	36	1	232.55	968.90	900	Y
T4	W3	3	37	38	244.53	1386.68	2000	Y
T4	W3	3	40	29	274.64	1450.01	1500	Y
T4	W3	3	38	41	298.87	1415.59	1100	N

Table A.16 shows the result of the AQoSSA with rescheduling. At interval T0, the algorithm firsts sorts the tasks according to the CoS from the smallest to the largest and then followed by the job length estimates from benchmarking from the highest to the lowest. From Table A.16, Tasks 5, 6, 7 and 1 which has CoS value equal to 1 are executed first followed by tasks which have CoS value equal to 2 and 3. Then, the mean value at each interval is computed using the equation 4.3 and a dynamic z value is used for the experiment. The mean value and a dynamic z value at each in interval are then used as the threshold to decide whether the task is schedule to the primary resource or secondary resource. Finally, the task is assigned to the resource within the group that returns the

minimum completion time. Besides, AQoSSA can reschedule the tasks with low priority when the completion times of the tasks do not exceed the deadline.

From Table A.16, Task 28 is rescheduled to the last task since the CoS value is equal to 3 and the completion time does not exceed the deadline. Table A.16 also shows whether the task is completed or violated the SLA. If the completion time is more than the acceptable percentile of the deadline, the job is violated. From Table A.16, Task 9 is not completed because the completion time is more than the acceptable percentile of the deadline. The Table A.16 shows that the implementation is correct. The interval used for the experiment is 20 seconds. The total number of job completed is 37 and the reliability is 92.5%. The total profit gained is 1007.10 units.

Table A.16 Task scheduling with AQoSSA (Rescheduling)

Interval	Web Services	CoS	Task ID	Resource ID	Duration	Completion Time	Deadline	Complete
T0	W2	1	5	29	454.19	454.19	450	Y
T0	W2	1	6	33	220.97	220.97	250	Y
T0	W3	1	7	38	220.79	220.79	300	Y
T0	W1	1	1	37	101.37	101.37	100	Y
T0	W3	2	8	37	253.01	354.37	800	Y
T0	W1	2	2	31	241.52	241.52	300	Y
T0	W1	2	3	41	208.88	208.88	200	Y
T0	W2	3	4	10	2115.63	2115.63	2400	Y
T1	W2	1	15	35	911.63	911.63	950	Y
T1	W2	1	13	33	489.22	690.19	700	Y
T1	W3	1	19	38	247.92	448.71	450	Y
T1	W1	1	10	31	227.67	449.19	450	Y
T1	W1	1	11	41	196.23	385.11	400	Y
T1	W1	1	9	37	67.09	401.47	200	N
T1	W3	2	17	37	258.09	659.56	900	Y
T1	W3	2	18	38	209.77	658.48	700	Y
T1	W2	3	16	9	3032.56	3032.56	2300	N
T1	W2	3	14	8	1661.81	1661.81	1800	Y

T1	W2	3	12	2	1712.35	1712.35	1700	Y
T2	W2	1	24	29	281.69	695.88	700	Y
T2	W1	1	20	41	290.19	655.30	700	Y
T2	W1	1	21	31	123.50	552.69	100	N
T2	W2	2	26	31	783.00	1335.69	1300	Y
T2	W1	2	22	29	126.63	822.51	800	Y
T2	W2	2	25	33	48.52	718.71	700	Y
T2	W1	3	23	1	60.82	60.82	100	Y
T3	W2	1	30	33	259.29	958	1250	Y
T3	W1	1	27	38	239.68	858.16	900	Y
T3	W3	1	32	37	177.98	797.54	800	Y
T3	W3	1	33	41	28.68	663.98	700	Y
T3	W2	2	29	29	699.31	1501.82	1400	Y
T4	W3	1	39	41	284.46	928.44	1000	Y
T4	W1	1	35	37	21.57	799.11	800	Y
T4	W1	2	34	37	130.03	929.15	1100	Y
T4	W1	2	36	38	76.84	915.00	900	Y
T3	W3	3	31	1	2226.15	2226.97	2200	Y
T4	W3	3	37	38	244.53	1159.53	2000	Y
T4	W3	3	40	37	220.37	1149.52	1500	Y
T4	W3	3	38	33	158.57	1096.57	1100	Y
T3	W1	3	28	35	231.37	1063	1100	Y