Annotated Bibliography for Resource Allocation Algorithms

1 Introduction

This annotated bibliography covers resource allocation algorithms for high performance computing (HPC) applications. It focuses more on distributed scheduling and ad-hoc approaches to allocating resources. I am less interested in the classical centralized scheduling algorithms where full knowledge is assumed. Seeing the application to distributed systems, the agent-based and market oriented approaches are currently more interesting.

2 Task Categories

There are many different categories of tasks. Tasks are simply work that users would like to accomplish. In some research areas tasks are also referred to as loads, jobs, or work.

2.1 Interactive

Jobs that are actively being manipulated by the user are interactive jobs.

2.2 Indivisible

Jobs not capable of being easily divided usually due to the lack of effort or the tight interconnection of the algorithm are referred to as indivisible jobs.

2.3 Divisible

Usually represented by directed acyclic graphs (DAG) representing the task dependencies. These tasks are usually very structured and well known.

2.4 Arbitrarily Divisible

A task that can be split an (almost) infinite number of times into smaller tasks and executed in parallel.

2.5 Parallel

These tasks are internally parallelized. They utilize message passing, multiple threads, and/or sub-tasks to distribute work across multiple cores and multiple machines while the jobs is running.

3 Computing Platforms

There are various types of high performance computing models used today. The platforms are listed with increasing coupling or connectedness. Interestingly enough they are also listed in decreasing fault tolerance. [Schroeder and Gibson, 2006]

3.1 Grid

Computing grids are very loosely coupled machines. They reside on the same network but can have different operating systems, architectures, and availability. [Li and Buyya, 2009] [Sulistio et al., 2003] [Foster and Kesselman, 2003] [Abramson et al., 2000]

3.1.1 Volunteer Computing

Volunteer computing is a relatively modern computing strategy where computing resources are volunteered to be part of the project for either recognition, fun, and sometimes monetary gain. Popular examples include SETI@home [Anderson et al., 2002] FOLDING@home [Beberg et al., 2009]

3.2 Cloud

Cloud computing has recently become a very popular computing platform with applications to HPC. Cloud computing has some key properties: pay for only what you

use, dynamic/elastic, virtualized networks, storage, and computing, secure, and internet connected.[Antonopoulos and Gillam, 2010] HPC instances have recently been added to Amazon EC2. Clouds utilized economies of scale to minimize the cost of its offerings.

Performance [Fenn et al., 2009] [Napper and Bientinesi, 2009]

Business model [Buyya et al., 2008]

3.3 Cluster

Clusters consist of many tightly coupled computing nodes. Typically they are connected with a fast interconnect. Each node runs its own instance of a common operating system. The top 500 supercomputers almost always are the cluster architecture. [Yeo et al., 2006]

Job arrival model [Lublin and Feitelson, 2003]

3.4 Shared Memory Machines

Large shared memory machines contain a large number of CPUs which all share memory and thus run one large and highly scalable operating system.

4 Resource Allocation

The goal of a resource allocation algorithms is to assign tasks to machines (or collections of machines) while trying to optimize a given objective. Objective functions include minimum time, minimum cost, and most reliable/fault-tolerant. [Sulistio et al., 2004]

4.1 Clustering

Given a collection of tasks, find the clusters of tasks that should be run on the same machines. This is somethings referred to as DAG scheduling. [Hall et al., 2007][Cao et al., 2010] [Gerasoulis and Yang, 1992]

4.2 Time Scheduling

This is the classic scheduling problem that is very similar to the NP-Hard bin packing problem. Determine the optimal placement of tasks on machines. [Bhat, 2008] [Benoit et al., 2010] [Brucker, 2007]

4.2.1 Shared Memory

Work stealing [Reinders, 2007] [Navarro et al., 2009]

Parallel containers [Buss et al., 2010]

Parallel FOR loops [Chandra et al., 2001]

Robust [Wang et al., 2009]

4.2.2 Distributed Memory

[Bobroff et al., 2008] [Jette et al., 2002] [Dusseau et al., 1996]

Heterogenous [Kalinov and Klimov, 2005]

Deadline uncertainty [Yeo and Buyya, 2006a]

Meta-scheduling [Hamscher et al., 2000]

Integer linear programming [Pan and Chen, 2005]

4.3 Environmental

Environmental aware schedulers consider environmental factors such as CO_2 emissions and electrical energy in the scheduling problem. These objectives may be part of the primary objective or a secondary objective. [Kim et al., 2011] [Al-Daoud et al., 2010] [Buyya et al., 2010]

4.4 Cloud

4.4.1 Instance Allocation

Determine which machine is best suited to host a particular instance. Current schedulers are fairly naive and include round-robin, most used first (to try to cluster instances and thus allow unused machines to sleep), by instance type, and by user. Instance allocation is performed by the cloud provider but may be aided by the user. [Li, 2009]

4.4.2 Instance Scheduling

Determining how many, and of what size, instances are required to execute a particular set of jobs. Also determine when to terminate the running instances when no longer needed. [Li, 2009]

Nash equilibrium [Sun et al., 2010]

Load balancing [Fang et al., 2010]

Over-subscribed [Chard et al., 2010] [Li, 2009]

Local cluster overflow, cloud bursting [de Assuncao et al., 2009]

4.5 Market-Oriented

One of the first problems encountered when dealing with over subscribed systems is how to assign priorities. Market-oriented scheduling policies side-step this problem by using cost as an alternative to artificial priorities. Users then have a finite amount of money or credits assigned to then (or earned) to which they can exchange for computing resources. These schedulers are often called matchmakers because they match offers to buyers of computing services. Cloud computing utilizes a market-oriented scheduling approach. Some general market-oriented scheduling papers include: [Yeo and Buyya, 2006b] [Salehi and Buyya, 2010] [Chard et al., 2010] [Xiao et al., 2008] [Buyya et al., 2008] [Buyya et al., 2009a] [Yeo and Buyya, 2007b] [Yeo and Buyya, 2004]

Utility computing [Yeo et al., 2010] [Buyya et al., 2009b] [Yeo and Buyya, 2007a] [Yeo and Buyya, 2009] [Yeo et al., 2007]

Service level agreement [Yeo and Buyya, 2005]

Agent-based [Veit, 2004] [Sun et al., 2010]

Brokering [Veit, 2004] [Abramson et al., 2002]

Auction [Zhao and Li, 2010]

4.6 Divisible Load Theory

Scheduling of communication heavy, arbitrarily divisible loads to graph of machines. These scheduling techniques are almost always optimal polynomial time algorithms. In addition then are often computable agent-recursively (removing the need for a centralized scheduler). Divisible load theory naturally handles heterogenous clusters of machines. [Bharadwaj et al., 2003] [Shokripour and Othman, 2009] [Lin et al., 2007] [Viswanathan et al., 2007] [Bharadwaj et al., 1996] [Blazewicz et al., 1999] [Robertazzi, 2003] [Benoit et al., 2010]

Linear and bus networks [Sohn and Robertazzi, 1998]

Buffer constraints [Li et al., 2000]

k-dimensional mesh networks [Chang et al., 2007] [Drozdowski and Glazek, 1999]

Tree networks [Beaumont et al., 2005] [Li and Veeravalli, 2010]

General networks [Jia et al., 2007] [Beaumont et al., 2003]

Applications [Altilar and Paker, 2002] [Chin et al., 2009] [Chan et al., 2001] [Viswanathan et al., 2007] [Marchal et al., 2005]

Dishonest machines [Carroll and Grosu, 2006]

5 Bibliography

References

[Abramson et al., 2002] Abramson, D., Buyya, R., and Giddy, J. (2002). A computational economy for grid computing and its implementation in the nimrod-g resource broker. *Future Gener. Comput. Syst.*, 18:1061–1074.

[Abramson et al., 2000] Abramson, D., Giddy, J., and Kotler, L. (2000). High performance parametric modeling with nimrod/g: Killer application for the global grid? In *Proceedings of the 14th International Symposium on Parallel and Distributed Processing*, pages 520–, Washington, DC, USA. IEEE Computer Society.

[Al-Daoud et al., 2010] Al-Daoud, H., Al-Azzoni, I., and Down, D. G. (2010). Power-aware linear programming based scheduling for heterogeneous computer clusters. In *Proceedings of the International Conference on Green Computing*, GREENCOMP '10, pages 325–332, Washington, DC, USA. IEEE Computer Society.

[Altilar and Paker, 2002] Altilar, D. T. and Paker, Y. (2002). Optimal scheduling algorithms for communication constrained parallel processing. In *Proceedings of the 8th International Euro-Par Conference on Parallel Processing*, Euro-Par '02, pages 197–206, London, UK. Springer-Verlag.

[Anderson et al., 2002] Anderson, D. P., Cobb, J., Korpela, E., Lebofsky, M., and Werthimer, D. (2002). Seti@home: an experiment in public-resource computing. *Commun. ACM*, 45:56–61.

[Antonopoulos and Gillam, 2010] Antonopoulos, N. and Gillam, L., editors (2010). *Cloud Computing: Principles, Systems and Applications.* Computer Communications and Networks. Springer, 1 edition.

[Beaumont et al., 2005] Beaumont, O., Casanova, H., Legrand, A., Robert, Y., and Yang, Y. (2005). Scheduling divisible loads on star and tree networks: Results and open problems. *IEEE Trans. Parallel Distrib. Syst.*, 16:207–218.

[Beaumont et al., 2003] Beaumont, O., Legrand, A., and Robert, Y. (2003). Scheduling divisible workloads on heterogeneous platforms. *Parallel Comput.*, 29:1121–1152.

[Beberg et al., 2009] Beberg, A. L., Ensign, D. L., Jayachandran, G., Khaliq, S., and Pande, V. S. (2009). Folding@home: Lessons from eight years of volunteer distributed computing. In *Proceedings of the 2009 IEEE International Symposium on*

Parallel&Distributed Processing, pages 1–8, Washington, DC, USA. IEEE Computer Society.

[Benoit et al., 2010] Benoit, A., Robert, Y., Rosenberg, A., and Vivien, F. (2010). Static worksharing strategies for heterogeneous computers with unrecoverable interruptions. *Parallel Computing*.

[Bharadwaj et al., 2003] Bharadwaj, V., Ghose, D., and Robertazzi, T. G. (2003). Divisible load theory: A new paradigm for load scheduling in distributed systems. *Cluster Computing*, 6:7–17.

[Bharadwaj et al., 1996] Bharadwaj, V., Robertazzi, T. G., and Ghose, D. (1996). *Scheduling Divisible Loads in Parallel and Distributed Systems*. IEEE Computer Society Press, Los Alamitos, CA, USA.

[Bhat, 2008] Bhat, U. N. (2008). *An Introduction to Queueing Theory: Modeling and Analysis in Applications (Statistics for Industry and Technology)*. Birkhäuser Boston, 1 edition.

[Blazewicz et al., 1999] Blazewicz, J., Drozdowski, M., and Markiewicz, M. (1999). Divisible task scheduling, concept and verification. *Parallel Comput.*, 25:87–98.

[Bobroff et al., 2008] Bobroff, N., Dasgupta, G., Fong, L., Liu, Y., Viswanathan, B., Benedetti, F., and Wagner, J. (2008). A distributed job scheduling and flow management system. *SIGOPS Oper. Syst. Rev.*, 42:63–70.

[Brucker, 2007] Brucker, P. (2007). *Scheduling Algorithms*. Springer Publishing Company, Incorporated, 5th edition.

[Buss et al., 2010] Buss, A., Harshvardhan, Papadopoulos, I., Pearce, O., Smith, T., Tanase, G., Thomas, N., Xu, X., Bianco, M., Amato, N. M., and Rauchwerger, L. (2010). Stapl: standard template adaptive parallel library. In *Proceedings of the 3rd Annual Haifa Experimental Systems Conference*, SYSTOR '10, pages 14:1–14:10, New York, NY, USA. ACM.

[Buyya et al., 2010] Buyya, R., Beloglazov, A., and Abawajy, J. (2010). Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges.

[Buyya et al., 2009a] Buyya, R., Venugopal, S., Ranjan, R., and Yeo, C. S. (2009a). The gridbus middleware for market-oriented computing. In Buyya, R. and Bubendorfer,

K., editors, *Market-oriented Grid and Utility Computing*, chapter 26, pages 589–622. John Wiley and Sons, Hoboken, NJ, USA.

[Buyya et al., 2008] Buyya, R., Yeo, C. S., and Venugopal, S. (2008). Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In *Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications (HPCC 2008)*, pages 5–13, Dalian, China. IEEE Computer Society: Los Alamitos, CA, USA.

[Buyya et al., 2009b] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., and Brandic, I. (2009b). Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25(6):599–616.

[Cao et al., 2010] Cao, H., Jin, H., Wu, X., Wu, S., and Shi, X. (2010). Dagmap: efficient and dependable scheduling of dag workflow job in grid. *J. Supercomput.*, 51:201–223.

[Carroll and Grosu, 2006] Carroll, T. E. and Grosu, D. (2006). A strategyproof mechanism for scheduling divisible loads in bus networks without control processors. In *Proceedings of the 20th international conference on Parallel and distributed processing*, IPDPS'06, pages 24–24, Washington, DC, USA. IEEE Computer Society.

[Chan et al., 2001] Chan, S. K., Bharadway, V., and Ghose, D. (2001). Large matrix-vector products on distributed bus networks with communication delays using the divisable load paradigm: performance analysis and simulaton. *Math. Comput. Simul.*, 58:71–92.

[Chandra et al., 2001] Chandra, R., Dagum, L., Kohr, D., Maydan, D., McDonald, J., and Menon, R. (2001). *Parallel programming in OpenMP*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Chang et al., 2007] Chang, Y.-K., Wu, J.-H., Chen, C.-Y., and Chu, C.-P. (2007). Improved methods for divisible load distribution on k-dimensional meshes using multi-installment. *IEEE Trans. Parallel Distrib. Syst.*, 18:1618–1629.

[Chard et al., 2010] Chard, K., Bubendorfer, K., and Komisarczuk, P. (2010). High occupancy resource allocation for grid and cloud systems, a study with drive. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, HPDC '10, pages 73–84, New York, NY, USA. ACM.

[Chin et al., 2009] Chin, T. T., Veeravalli, B., and Jia, J. (2009). Handling large-size discrete wavelet transform on network-based computing systems - parallelization via divisible load paradigm. *J. Parallel Distrib. Comput.*, 69:143–152.

[de Assuncao et al., 2009] de Assuncao, M. D., di Costanzo, A., and Buyya, R. (2009). Evaluating the cost-benefit of using cloud computing to extend the capacity of clusters. In *Proceedings of the 18th ACM international symposium on High performance distributed computing*, HPDC '09, pages 141–150, New York, NY, USA. ACM.

[Drozdowski and Glazek, 1999] Drozdowski, M. and Glazek, W. (1999). Scheduling divisible loads in a three-dimensional mesh of processors. *Parallel Comput.*, 25:381–404.

[Dusseau et al., 1996] Dusseau, A. C., Arpaci, R. H., and Culler, D. E. (1996). Effective distributed scheduling of parallel workloads. *SIGMETRICS Perform. Eval. Rev.*, 24:25–36.

[Fang et al., 2010] Fang, Y., Wang, F., and Ge, J. (2010). A task scheduling algorithm based on load balancing in cloud computing. In Wang, F., Gong, Z., Luo, X., and Lei, J., editors, *Web Information Systems and Mining*, volume 6318 of *Lecture Notes in Computer Science*, pages 271–277. Springer Berlin / Heidelberg.

[Fenn et al., 2009] Fenn, M., Murphy, M. A., and Goasguen, S. (2009). A study of a kvm-based cluster for grid computing. In *Proceedings of the 47th Annual Southeast Regional Conference*, ACM-SE 47, pages 34:1–34:6, New York, NY, USA. ACM.

[Foster and Kesselman, 2003] Foster, I. and Kesselman, C. (2003). *The Grid 2: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Gerasoulis and Yang, 1992] Gerasoulis, A. and Yang, T. (1992). A comparison of clustering heuristics for scheduling directed acyclic graphs on multiprocessors. *Journal of Parallel and Distributed Computing*, 16(4):276 – 291.

[Hall et al., 2007] Hall, R., Rosenberg, A., and Venkataramani, A. (2007). A comparison of dag-scheduling strategies for internet-based computing. In *Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International*, pages 1 –9.

[Hamscher et al., 2000] Hamscher, V., Schwiegelshohn, U., Streit, A., and Yahyapour, R. (2000). Evaluation of job-scheduling strategies for grid computing. In *Proceedings of the First IEEE/ACM International Workshop on Grid Computing*, GRID '00, pages 191–202, London, UK. Springer-Verlag.

[Jette et al., 2002] Jette, M. A., Yoo, A. B., and Grondona, M. (2002). Slurm: Simple linux utility for resource management. In *In Lecture Notes in Computer Science: Proceedings of Job Scheduling Strategies for Parallel Processing (JSSPP) 2003*, pages 44–60. Springer-Verlag.

[Jia et al., 2007] Jia, J., Veeravalli, B., and Ghose, D. (2007). Adaptive load distribution strategies for divisible load processing on resource unaware multilevel tree networks. *IEEE Trans. Comput.*, 56:999–1005.

[Kalinov and Klimov, 2005] Kalinov, A. and Klimov, S. (2005). Optimal mapping of a parallel application processes onto heterogeneous platform. In *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05) - Workshop 1 - Volume 02*, IPDPS '05, pages 123.2–, Washington, DC, USA. IEEE Computer Society.

[Kim et al., 2011] Kim, K. H., Beloglazov, A., and Buyya, R. (2011). Power-aware provisioning of virtual machines for real-time Cloud services. *Concurrency Computat.: Pract. Exper.*, page n/a.

[Li and Buyya, 2009] Li, H. and Buyya, R. (2009). Model-based simulation and performance evaluation of grid scheduling strategies. *Future Gener. Comput. Syst.*, 25:460–465.

[Li, 2009] Li, L. (2009). An optimistic differentiated service job scheduling system for cloud computing service users and providers. In *Multimedia and Ubiquitous Engineering, 2009. MUE '09. Third International Conference on.*, pages 295 –299.

[Li et al., 2000] Li, X., Bharadwaj, V., and Ko, C. (2000). Divisible load scheduling on single-level tree networks with buffer constraints. *Aerospace and Electronic Systems, IEEE Transactions on.*, 36(4):1298 – 1308.

[Li and Veeravalli, 2010] Li, X. and Veeravalli, B. (2010). Ppdd: scheduling multi-site divisible loads in single-level tree networks. *Cluster Computing*, 13:31–46.

[Lin et al., 2007] Lin, X., Lu, Y., Deogun, J., and Goddard, S. (2007). Real-time divisible load scheduling for cluster computing. In *Real Time and Embedded Technology and Applications Symposium, 2007. RTAS '07. 13th IEEE*, pages 303 –314.

[Lublin and Feitelson, 2003] Lublin, U. and Feitelson, D. G. (2003). The workload on parallel supercomputers: modeling the characteristics of rigid jobs. *J. Parallel Distrib. Comput.*, 63:1105–1122.

[Marchal et al., 2005] Marchal, L., Yang, Y., Casanova, H., and Robert, Y. (2005). A realistic network/application model for scheduling divisible loads on large-scale platforms. In *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05) - Papers - Volume 01*, IPDPS '05, pages 48.2–, Washington, DC, USA. IEEE Computer Society.

[Napper and Bientinesi, 2009] Napper, J. and Bientinesi, P. (2009). Can cloud computing reach the top500? In *Proceedings of the combined workshops on UnConventional high performance computing workshop plus memory access workshop*, UCHPC-MAW '09, pages 17–20, New York, NY, USA. ACM.

[Navarro et al., 2009] Navarro, A., Asenjo, R., Tabik, S., and Caşcaval, C. (2009). Load balancing using work-stealing for pipeline parallelism in emerging applications. In *Proceedings of the 23rd international conference on Supercomputing*, ICS '09, pages 517–518, New York, NY, USA. ACM.

[Pan and Chen, 2005] Pan, J. C.-H. and Chen, J.-S. (2005). Mixed binary integer programming formulations for the reentrant job shop scheduling problem. *Comput. Oper. Res.*, 32:1197–1212.

[Reinders, 2007] Reinders, J. (2007). *Intel Threading Building Blocks: Outfitting C++ for Multi-Core Processor Parallelism*. O'Reilly Media, 1 edition.

[Robertazzi, 2003] Robertazzi, T. G. (2003). Ten reasons to use divisible load theory. *Computer*, 36:63–68.

[Salehi and Buyya, 2010] Salehi, M. and Buyya, R. (2010). Adapting market-oriented scheduling policies for cloud computing. In Hsu, C.-H., Yang, L., Park, J., and Yeo, S.-S., editors, *Algorithms and Architectures for Parallel Processing*, volume 6081 of *Lecture Notes in Computer Science*, pages 351–362. Springer Berlin / Heidelberg.

[Schroeder and Gibson, 2006] Schroeder, B. and Gibson, G. A. (2006). A large-scale study of failures in high-performance computing systems. *Dependable Systems and Networks, International Conference on*, 0:249–258.

[Shokripour and Othman, 2009] Shokripour, A. and Othman, M. (2009). Survey on divisible load theory and its applications. In *Information Management and Engineering, 2009. ICIME '09. International Conference on*, pages 300 –304.

[Sohn and Robertazzi, 1998] Sohn, J. and Robertazzi, T. (1998). Optimal time-varying load sharing for divisible loads. *Aerospace and Electronic Systems, IEEE Transactions on*, 34(3):907 –923.

[Sulistio et al., 2003] Sulistio, A., Yeo, C. S., and Buyya, R. (2003). Visual modeler for grid modeling and simulation (gridsim) toolkit. In *Proceedings of the Workshop on Innovative Solutions for Grid Computing,International Conference on Computational Science (ICCS 2003)*, volume 2659/2003 of *Lecture Notes in Computer Science (LNCS)*, pages 1123–1132, Melbourne, Australia. Springer-Verlag: Heidelberg, Germany.

[Sulistio et al., 2004] Sulistio, A., Yeo, C. S., and Buyya, R. (2004). A taxonomy of computer-based simulations and its mapping to parallel and distributed systems simulation tools. *Software: Practice and Experience*, 34(7):653–673.

[Sun et al., 2010] Sun, D., Chang, G., Wang, C., Xiong, Y., and Wang, X. (2010). Efficient nash equilibrium based cloud resource allocation by using a continuous double auction. In *Computer Design and Applications (ICCDA), 2010 International Conference on*, volume 1, pages V1–94 –V1–99.

[Veit, 2004] Veit, D. J. (2004). *Matchmaking in Electronic Markets: An Agent-Based Approach Towards Matchmaking in Electronic Negotiations (Lecture Notes in Computer Science, 2882.)*. SpringerVerlag.

[Viswanathan et al., 2007] Viswanathan, S., Veeravalli, B., and Robertazzi, T. G. (2007). Resource-aware distributed scheduling strategies for large-scale computational cluster/grid systems. *IEEE Trans. Parallel Distrib. Syst.*, 18:1450–1461.

[Wang et al., 2009] Wang, X., Yeo, C. S., Buyya, R., and Su, J. (2009). Reliability-driven reputation based scheduling for public-resource computing using ga. In *Proceedings of the 23th IEEE International Conference on Advanced Information Networking and Applications (AINA 2009)*, pages 411–418, Bradford, UK. IEEE Computer Society: Los Alamitos, CA, USA.

[Xiao et al., 2008] Xiao, L., Zhu, Y., Ni, L. M., and Xu, Z. (2008). Incentive-based scheduling for market-like computational grids. *IEEE Trans. Parallel Distrib. Syst.*, 19:903–913.

[Yeo and Buyya, 2004] Yeo, C. S. and Buyya, R. (2004). Pricing for utility-driven resource management and allocation in clusters. In *Proceedings of the 12th International Conference on Advanced Computing and Communications (ADCOM 2004)*, pages 32–41, Ahmedabad, India. Allied Publishers: New Delhi, India.

[Yeo and Buyya, 2005] Yeo, C. S. and Buyya, R. (2005). Service level agreement based allocation of cluster resources: Handling penalty to enhance utility. In *Proceedings*

of the 7th IEEE International Conference on Cluster Computing (Cluster 2005), Boston, MA, USA. IEEE Computer Society: Los Alamitos, CA, USA.

[Yeo and Buyya, 2006a] Yeo, C. S. and Buyya, R. (2006a). Managing risk of inaccurate runtime estimates for deadline constrained job admission control in clusters. In *Proceedings of the 35th International Conference on Parallel Processing (ICPP 2006)*, pages 451–458, Columbus, OH, USA. IEEE Computer Society: Los Alamitos, CA, USA.

[Yeo and Buyya, 2006b] Yeo, C. S. and Buyya, R. (2006b). A taxonomy of market-based resource management systems for utility-driven cluster computing. *Software: Practice and Experience*, 36(13):1381–1419.

[Yeo and Buyya, 2007a] Yeo, C. S. and Buyya, R. (2007a). Integrated risk analysis for a commercial computing service. In *Proceedings of the 21st IEEE International Parallel and Distributed Processing Symposium (IPDPS 2007)*, Long Beach, CA, USA. IEEE Computer Society: Los Alamitos, CA, USA.

[Yeo and Buyya, 2007b] Yeo, C. S. and Buyya, R. (2007b). Pricing for utility-driven resource management and allocation in clusters. *International Journal of High Performance Computing Applications*, 21(4):405–418.

[Yeo and Buyya, 2009] Yeo, C. S. and Buyya, R. (2009). Integrated Risk Analysis for a Commercial Computing Service in Utility Computing. *Journal of Grid Computing*, 7(1):1–24.

[Yeo et al., 2007] Yeo, C. S., Buyya, R., de Assuncao, M. D., Yu, J., Sulistio, A., Venugopal, S., and Placek, M. (2007). Utility computing on global grids. In Bidgoli, H., editor, *Handbook of Computer Networks*. John Wiley and Sons, Hoboken, NJ, USA.

[Yeo et al., 2006] Yeo, C. S., Buyya, R., Pourreza, H., Eskicioglu, R., Graham, P., and Sommers, F. (2006). Cluster computing: High-performance, high-availability, and high-throughput processing on a network of computers. In Zomaya, A. Y., editor, *Handbook of Nature-Inspired and Innovative Computing: Integrating Classical Models with Emerging Technologies*, chapter 16, pages 521–551. Springer, New York, NY, USA.

[Yeo et al., 2010] Yeo, C. S., Venugopal, S., Chu, X., and Buyya, R. (2010). Autonomic metered pricing for a utility computing service. *Future Generation Computer Systems*, 26(8):1368–1380.

[Zhao and Li, 2010] Zhao, H. and Li, X. (2010). Auctionnet: Market oriented task scheduling in heterogeneous distributed environments. In *IPDPS Workshops*, pages 1–4. IEEE.