Task and Machine Heterogeneities: Higher Moments Matter

Abdulla M. Al-Qawasmeh[1], Anthony A. Maciejewski[1], Howard Jay Siegel[1,2], Jay Smith[1,3], and Jerry Potter[1]
[1]Department of Electrical and Computer Engineering, [2]Department of Computer Science,
Colorado State University, Fort Collins, Colorado, USA
[3]DigitalGlobe Inc., Longmont, Colorado, USA
{Abdulla.Al-Qawasmeh, aam, hj, jerry.potter}@colostate.edu
jtsmith@digitalglobe.com

Abstract - *One type of heterogeneous computing (HC) systems consists of machines with diverse capabilities harnessed together to execute a set of tasks that vary in their computational complexity. An HC system can be characterized using an Estimated Time to Compute (ETC) matrix. Each value in this matrix represents the ETC of a specific task on a specific machine when executed exclusively. Heuristics use the values in the ETC matrix to allocate tasks to machines in the HC system. The performance of resource allocation heuristics can be affected significantly by factors such as task and machine heterogeneities. Therefore, quantifying heterogeneity will allow a system to select a heuristic appropriate for the given heterogeneous environment. In this paper, we identify different central moments used to quantify the heterogeneity of ETC matrices obtained from real world systems and benchmark data, and show the effect of these moments on the performance of heuristics both through simple examples and simulations.*

Keywords: distributed systems; heterogeneity; heterogeneous systems; mapping heuristics; task allocation

1 Introduction

One type of heterogeneous computing (HC) systems consists of machines with diverse capabilities harnessed together to execute a set of tasks that vary in their computational complexity. The Estimated Time to Compute (ETC) each task on each machine in an HC system is arranged in an ETC matrix, where entry *ETC*(*i*, *j*) is the estimated execution time of task *i* on machine *j* when executed alone. The assumption of such ETC information is a common practice in resource allocation research (e.g., [5, 11, 14, 16, 17, 22, 24]). Examples of ETC matrices are given in Subsections 2.3 and 2.4.

Machine heterogeneity is the degree to which the execution time of a given task varies for different machines (the variation along the same row of an ETC matrix). Analogously, task heterogeneity is the degree to which the execution times of different tasks vary for the same machine (the variation along the same column) in the ETC matrix.

In an HC system, tasks should be mapped to the available machines in a way that optimizes some performance objective (e.g., [5, 6, 7, 8, 9, 13, 18, 19, 20, 23]). Mapping tasks to machines in HC systems has been shown to be, in general, an NP-Complete problem [10, 12, 15]. Hence, many heuristics have been developed for allocating tasks to machines in HC systems. It also has been shown in [7] that the performance of resource allocation heuristics is affected significantly by factors such as the level of machine heterogeneity. Therefore, quantifying the heterogeneity of a given heterogeneous environment will allow the selection of a heuristic that is the most appropriate.

In previous work, either the range of the execution time values or their standard deviation was used as a measure of the heterogeneity to generate ETC matrices for simulation studies. These measures do not completely represent the possible variation in heterogeneity. For example, many ETC matrices with the same standard deviation can have other statistical properties (that are vastly different and that may be highly correlated with a mapping heuristic's performance). The decision of what measure best quantifies heterogeneity should be based on how it affects the performance of the heuristics being evaluated in an HC system. The contributions of this paper are: (1) an identification of different central moments (such as skewness and kurtosis) used to quantify task and machine heterogeneity, and (2) task allocation examples and simulations that demonstrate the importance of these moments. The examples show that if these statistical measures are ignored, incorrect assumptions about the potential performance of an HC system and the applied heuristics can be made.

This research was supported by the NSF under grant No. CNS-0615170 and by the Colorado State University George T. Abell Endowment.

This paper is organized as follows. Section 2 discusses different statistical measures and central moments that can be used to quantify heterogeneity. Simulation setup is described in Section 3. Simulation results are given in Section 4. Section 5 discusses related work. Conclusions are given in Section 6.

2 Measuring the Heterogeneity of an ETC Matrix

2.1 Introduction

In this section, we describe three statistical measures: (a) coefficient of variation, (b) skewness, also called the third central moment, and (c) kurtosis, also called the fourth central moment. An ETC matrix for a given environment can be obtained from user supplied information, experimental data, or task profiling and analytical benchmarking [1, 14, 17, 25]. The statistical measures described in this section can be used to characterize the heterogeneity of an existing ETC matrix.

We give two simple examples that illustrate the effect of skewness and kurtosis on the performance of two heuristics: Min-Min [15] and Max-Min [15]. These examples are given in Subsections 2.3 and 2.4, respectively.

The following variables will be used in the calculation of each of the statistical measures: $\underline{T}$ is the number of tasks, $\underline{M}$ is the number of machines, $\underline{\mu_i^{(\mathrm{t})}}$ is the mean ETC of task i over all machines, given by

$$\mu_i^{(\mathrm{t})} = \frac{1}{M}\sum_{j=1}^{M}\mathrm{ETC}(i, j)\,.$$

The mean ETC of all tasks on machine j, $\underline{\mu_j^{(\mathrm{m})}}$, is given by

$$\mu_j^{(\mathrm{m})} = \frac{1}{T}\sum_{i=1}^{T}\mathrm{ETC}(i, j)\,.$$

The standard deviation of the ETC of task i over all machines, $\underline{\sigma_i^{(\mathrm{t})}}$, is given by

$$\sigma_i^{(\mathrm{t})} = \sqrt{\frac{1}{M}\sum_{j=1}^{M}(\mathrm{ETC}(i, j) - \mu_i^{(\mathrm{t})})^2}\,.$$

The standard deviation of the ETC of all tasks on machine j, $\underline{\sigma_j^{(\mathrm{m})}}$, is given by

$$\sigma_j^{(\mathrm{m})} = \sqrt{\frac{1}{T}\sum_{i=1}^{T}(\mathrm{ETC}(i, j) - \mu_j^{(\mathrm{m})})^2}\,.$$

2.2 Coefficient of Variation

The COV of a set of values with standard deviation σ and mean μ is given by

$$\mathrm{COV} = \frac{\sigma}{\mu}\,.$$

Let $V_i^{(\mathrm{t})}$ be machine COV for task i, given by

$$V_i^{(\mathrm{t})} = \frac{\sigma_i^{(\mathrm{t})}}{\mu_i^{(\mathrm{t})}}\,.$$

Let $V_j^{(\mathrm{m})}$ be task COV for machine j, given by

$$V_j^{(\mathrm{m})} = \frac{\sigma_j^{(\mathrm{m})}}{\mu_j^{(\mathrm{m})}}\,.$$

Task heterogeneity as measured by the Average Task COV (ATC) is given by

$$\mathrm{ATC} = \left[\sum_{j=1}^{M} V_j^{(\mathrm{m})}\right] / M\,.$$

Machine heterogeneity as measured by the Average Machine COV (AMC) is given by

$$\mathrm{AMC} = \left[\sum_{i=1}^{T} V_i^{(\mathrm{t})}\right] / T\,.$$

Although both ATC and AMC quantify the variation of the execution time values, they do not indicate whether most of the values are less than or greater than the mean, and whether the variation is caused by many values having an average deviation from the mean, or small number of values having large deviation from the mean. These are quantified by the skewness and the kurtosis, respectively. Subsections 2.3 and 2.4 describe how the skewness and kurtosis may have a great effect on the performance of heuristics. Thus, ignoring these heterogeneity measures may lead to the wrong choice of a heuristic for mapping the tasks in a given HC system.

2.3 Skewness (Third Central Moment)

The skewness of a set of values measures the degree of asymmetry of the values over the mean. Positive skewness means that most of the values are below the mean and negative skewness means that most of the values are greater than the mean.

Let $S_i^{(t)}$ be machine skewness for task *i*, given by

$$S_i^{(t)} = \left[\frac{1}{M} \sum_{j=1}^{M} (\mathrm{ETC}(i,j) - \mu_i^{(t)})^3 \right] / (\sigma_i^{(t)})^3 .$$

Let $S_j^{(m)}$ be task skewness for machine *j*, given by

$$S_j^{(m)} = \left[\frac{1}{T} \sum_{i=1}^{T} (\mathrm{ETC}(i,j) - \mu_j^{(m)})^3 \right] / (\sigma_j^{(m)})^3 .$$

Task heterogeneity as measured by the Average Task Skewness (ATS) is given by

$$\mathrm{ATS} = \left[\sum_{j=1}^{M} S_j^{(m)} \right] / M .$$

Machine heterogeneity as measured by the Average Machine Skewness (AMS) is given by

$$\mathrm{AMS} = \left[\sum_{i=1}^{T} S_i^{(t)} \right] / T .$$

An example to show the effect of skewness, we consider two heuristics: Max-Min and Min-Min. The two heuristics have been studied widely (e.g., [2, 3, 7, 8, 15, 19]). Figure 1 shows the procedure for the Min-Min heuristic. Min-Min starts by assigning tasks with small execution times before ones with longer execution times. The procedure for Max-Min is similar to that of Min-Min except that in step (b) instead of mapping the task that has the minimum completion time it maps the task the has the maximum completion time. Max-Min heuristic starts by assigning the tasks with longer execution times.

Do the following steps while there are unmapped tasks

(1) For each unmapped task determine the machine that gives the task its *minimum* completion time.

(2) Among the task-machine pairs determined in (1) map the task that has the *minimum* completion time to the corresponding machine

(3) update the ready times of the machine where the task was mapped to.

Figure 1. Procedure for Min-Min.

Min-Min performs better than Max-Min in most cases [7, 8]. However, Max-Min outperforms Min-Min when there are many more shorter tasks than there are longer ones (i.e., the corresponding ETC matrix has high positive task skewness). This is because Max-Min starts by assigning the longer tasks to their best machine. Table 1 shows a scenario in which the Max-Min heuristic outperforms the Min-Min heuristic. The ETC shown in Table 1 has a positive ATS value of 0.62 and an ATC value of 0.7. The makespan, which is the greatest finish time among all the machines, of the mapping produced by the Min-Min heuristic is 22 and the makespan of the mapping produced by Max-Min is 18. A pictorial representation of the assignments made by each heuristic for the ETC in Table 1 is given in Figure 2.

Table 1. An example ETC matrix that illustrates the situation where the Max-Min heuristic outperforms the Min-Min heuristic for an ETC matrix with high task skewness.

	m_1	m_2
t_1	3	5
t_2	7	4
t_3	20	18

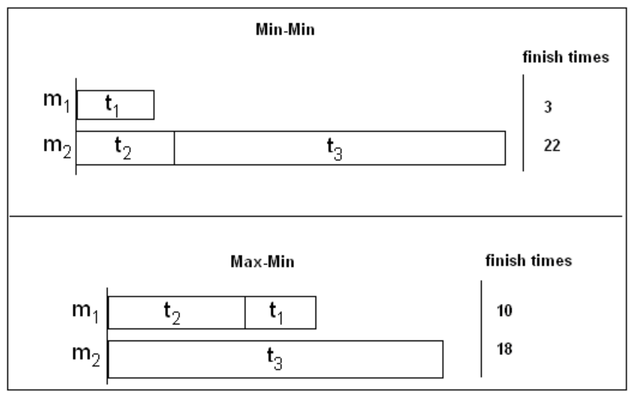

Figure 2. A pictorial representation of the mapping produced by the Max-Min and Min-Min heuristics for the ETC matrix given in Table 1.

An example where the Min-Min heuristic outperforms Max-Min for an ETC matrix with negative task skewness is given in Table 2. The ETC in Table 2 has a negative ATS value of –0.54, and an ATC value of 0.45. The makespan of the mapping produced by Min-Min is 23 and the makespan of the mapping produced by Max-Min is 25. A pictorial representation of the mapping produced by each heuristic is given in Figure 3.

Table 2. An Example ETC Matrix that illustrates the situation where the Min-Min heuristic outperforms the Max-Min heuristic for an ETC matrix with negative task skewness

	m_1	m_2
t_1	3	6
t_2	25	12
t_3	20	14

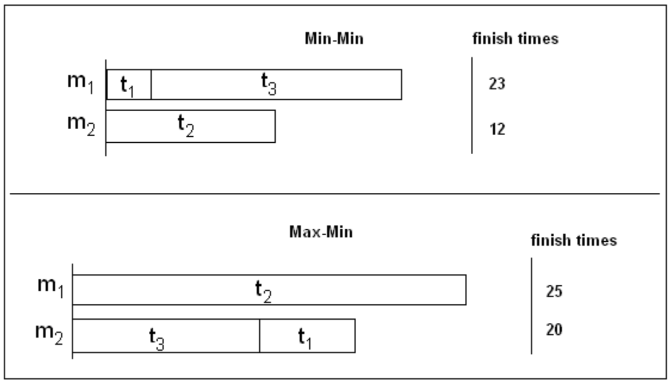

Figure 3. Pictorial representation of the mapping produced by Min-Min and Max-Min heuristics for the ETC given in Table 2.

2.4 Kurtosis (Fourth Central Moment)

The kurtosis of a set of values measures the extent to which the deviation is caused by a small number of values having extreme deviation from the mean versus large number of values having modestly-sized deviations. Higher values of kurtosis indicate that the standard deviation is caused by fewer values having extreme deviation. The definition of kurtosis that we use in this paper is the excess kurtosis. Excess kurtosis equals kurtosis minus three. This makes the excess kurtosis of the Gaussian (normal) distribution equal to zero.

Let $\underline{K_i^{(\mathrm{t})}}$ be machine kurtosis for task *i*, given by

$$K_i^{(\mathrm{t})} = \left[\left(\frac{1}{M}\sum_{j=1}^{M}(\mathrm{ETC}(i,j)-\mu_i^{(\mathrm{t})})^4\right)/(\sigma_i^{(\mathrm{t})})^4\right]-3.$$

Let $\underline{K_j^{(\mathrm{m})}}$ be task kurtosis for machine *j*, given by

$$K_j^{(\mathrm{m})} = \left[\left(\frac{1}{T}\sum_{i=1}^{T}(\mathrm{ETC}(i,j)-\mu_j^{(\mathrm{m})})^4\right)/(\sigma_j^{(\mathrm{m})})^4\right]-3.$$

Task heterogeneity as measured by the Average Task Kurtosis (ATK) is given by

$$\mathrm{ATK} = \frac{1}{M}\sum_{j=1}^{M}K_j^{(\mathrm{m})}.$$

Machine heterogeneity as measured by the Average Machine Kurtosis (AMK) is given by

$$\mathrm{AMK} = \frac{1}{T}\sum_{i=1}^{T}K_i^{(\mathrm{t})}.$$

To show how kurtosis may affect the performance of two heuristics, we give an example in Table 3 of an ETC matrix with high kurtosis where the Max-Min heuristic outperforms Min-Min. The pictorial representation of the mapping produced by Min-Min and Max-Min of the ETC in Table 3 is given in Figure 5. The makespan for the mapping produced by Max-Min is 212 and the makespan for the mapping produced by Min-Min is 243. Although ATS in this table is close to zero (–0.039), this ETC matrix still has the property that there are more machines with smaller execution time tasks than there are ones with greater execution time (t8 in Table 3). The reason that this matrix has a small magnitude of skewness is that there is a task (t1 in Table 3) with a very short average execution time. However, this ETC matrix has a high ATK value (0.92) compared to the normal distribution, which has zero kurtosis, and the uniform distribution, which has kurtosis of –1.2. The ATC of the ETC in Table 3 is 0.47. A pictorial representation of the mapping produced by each heuristic for the ETC matrix in Table 3 is given in Figure 4.

An example where Min-Min outperforms Max-Min for an ETC matrix with low task kurtosis is given in Table 4. The ETC in Table 4 has a low kurtosis value of –0.31. The ATS value of the ETC matrix is –0.37, and the ATC value is 0.1. The makespan of the mapping produced by Min-Min is 190 and the makespan of the mapping produced by Max-Min is 198. A representation of the mapping produced by each heuristic for the ETC matrix in Table 4 is given in Figure 5.

Table 3. An example ETC matrix that illustrates the situation where the Max-Min heuristic outperforms the Min-Min heuristic for an ETC matrix with high kurtosis and low skewness.

	m_1	m_2
t_1	3	5
t_2	55	53
t_3	49	47
t_4	50	52
t_5	54	56
t_6	45	51
t_7	54	50
t_8	105	93

The examples given in Tables 1, 2, 3, and 4 are just to illustrate possible ways different values of kurtosis and skewness effect heuristics. Other heuristics' performance may be correlated in a different manner with the skewness and kurtosis values of the ETC matrix. Max-Min and Min-Min heuristics were chosen for these examples because, in general, they have large difference in performance. These examples show some special cases where Max-Min has better performance.

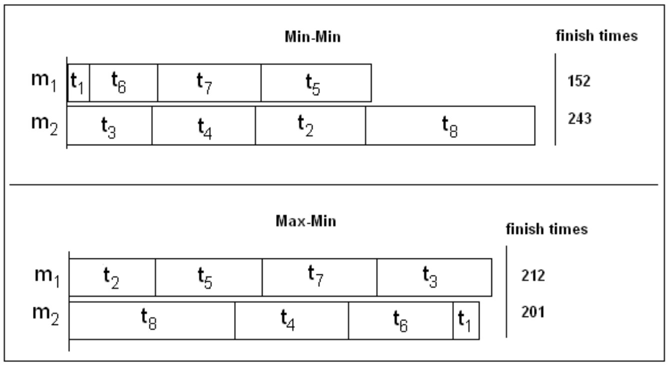

Figure 4. A pictorial representation of the mapping produced by the Max-Min and Min-Min heuristics for the ETC matrix given in Table 3.

Table 4. An example ETC matrix that illustrates the situation where the Min-Min heuristic outperforms the Max-Min heuristic for an ETC matrix with low kurtosis.

	m_1	m_2
t_1	49	44
t_2	55	50
t_3	49	44
t_4	48	52
t_5	50	56
t_6	40	51
t_7	54	46
t_8	52	59

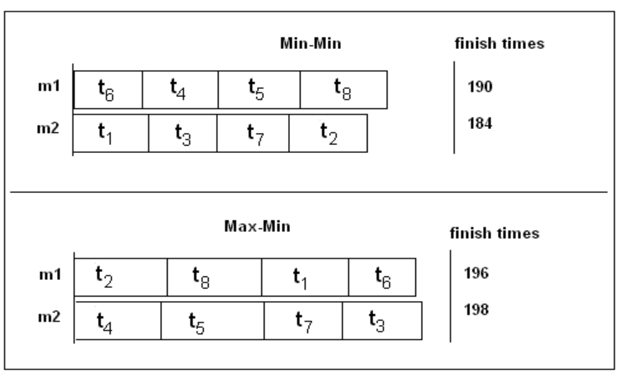

Figure 5. A pictorial representation of the mapping produced by the Max-Min and Min-Min heuristics for the ETC matrix given in Table 4.

3 Simulation Setup

We conducted a number of simulations to assess the effect of different COV and skewness values on the performance of the selected heuristics. The performance of the heuristics is calculated in terms of the makespan where a smaller makespan is better. Each of the ETC matrices that were used in the simulations was generated via the coefficient-of-variation-based method (CVB) proposed in [4]. The CVB method uses the COV to represent task and machine heterogeneity. To generate an ETC matrix, the CVB method takes three parameters: (a) task COV, (b) machine COV, and (c) the mean task execution time. The CVB method uses a gamma distribution to generate the execution time values. The shape parameter α and the scale parameter β can be expressed in terms of task COV, machine COV, and the mean task execution time. The number of tasks and machines in each of the ETC matrices used in the simulation are 128 and 8, respectively. For simplicity, static mapping, similar to the mapping problem in [7], was considered in this study.

After each ETC was generated using a specific mean, task COV, and machine COV (parameters), we recalculated these parameters of the generated ETC matrix to obtain their actual values. Actual parameter values differ from those that were used to generate the ETC matrix due to a limited number of values being generated. The heuristics used in the studies are: Min-Min, and Max-Min.

Consistent, inconsistent and partially-consistent matrices were used in the simulations. For a machine consistent matrix, if machine m_i is faster than machine m_j for a given task, then m_i must be faster than m_j for all other tasks [7]. Similarly, for a task consistent matrix, if task t_i runs faster on that task t_j on a given machine, then t_i must run faster than t_j on all machines [21]. An inconsistent ETC matrix is neither machine consistent nor task consistent [7]. A partially-consistent ETC matrix is an inconsistent ETC matrix with a consistent sub-matrix [7]. In the partially-consistent ETC matrices used in our studies, half of the machines (4 machines) and half of the tasks (64 tasks) are consistent. We found that the most significant difference between the performances of the heuristics under study is for partially-consistent ETC matrices, so we show only the results for partially-consistent ETC matrices in the next section.

4 Simulation Results

Figures 6 and 7 show the normalized makespan, which is the makespan of the heuristic divided by the makespan of Max-Min. In Figure 6, machine COV was fixed at 0.1 and the task COV was increased from 0.01 to 1.5. After the ETC matrices were generated the task skewness value was calculated for each ETC matrix. The average machine skewness for all the ETC matrices shown in Figure 6 is 0.09. As shown in the figure, Max-Min outperforms Min-Min for task skewness values greater than 1.4. In Figure 7, task COV was fixed at 0.7 and the machine COV was increased from 0.01 to 1.5. Min-Min outperforms Max-Min for machine COV values greater than 0.5.

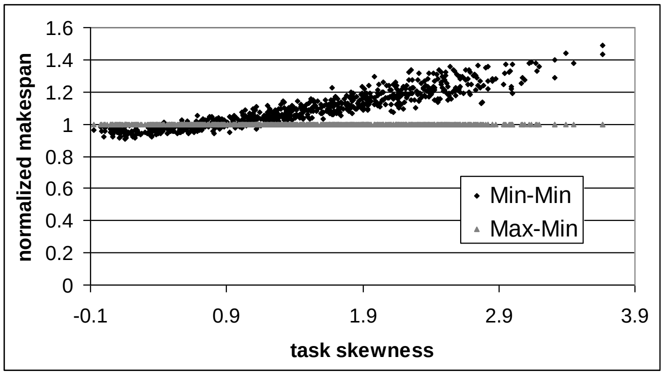

Figure 6. Scatter plot of the makespan of Min-Min normalized with respect to Max-Min. Machine COV in this figure is fixed at 0.1, and the task COV is increased from 0.01 to 1.5.

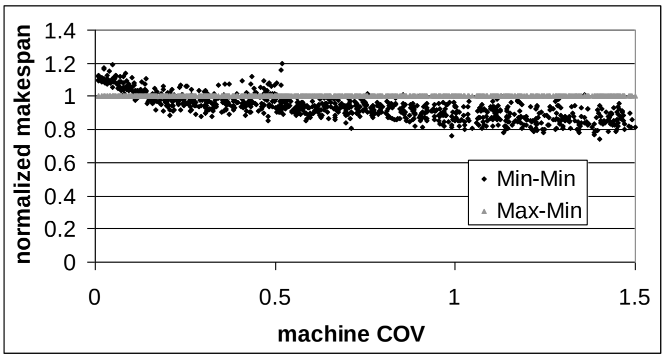

Figure 7. Scatter plot of the makespan of Min-Min normalized with respect to Max-Min. Task COV is fixed at 0.7, and machine COV is increased from 0.01 to 1.5.

Simulations using consistent and inconsistent ETC matrices have shown similar correlation between the performance of Min-Min and Max-Min with the statistical measures. However, the difference in performance was less significant.

5 Related Work

ETC matrices were previously used with different degrees of heterogeneity (e.g., [1, 5, 7, 8, 19, 23]). Most of these ETC matrices were generated by the range-based method described in [7] and the CVB described in [4]. Therefore, depending on which method was used, heterogeneity was assumed to be either the range of the execution time values, or the COV. To the best of our knowledge, no previous efforts have used other measures to quantify the heterogeneity of the execution time values.

6 Conclusions

In this paper, the use of higher central moments (skewness and kurtosis) to quantify heterogeneity was proposed. A method to calculate each of the moments and measures for an existing ETC matrix was described. The impact that each heterogeneity measure may have on the performance of Min-Min and Max-Min heuristics was demonstrated through simple examples. In addition, simulations have been conducted to show the impact of the COV and skewness on the heuristics. Further simulations have to be carried out to show the effect that kurtosis may have on the performance of heuristics.

Acknowledgments: The authors thank Arnold Rosenberg, Paul Maxwell, and Luis Briceño for their valuable comments on this work.

References

[1] S. Ali, T. D. Braun, H. J. Siegel, A. A. Maciejewski, N. Beck, L. Boloni, M. Maheswaran, A. I. Reuther, J. P. Robertson, M. D. Theys, and B. Yao, “Characterizing resource allocation heuristics for heterogeneous computing systems,” in *Advances in Computers Volume 63: Parallel, Distributed, and Pervasive Computing*, 2005, pp. 91–128.

[2] S. Ali, J. K. Kim, H. J. Siegel, A. A. Maciejewski, Y. Yu, S. B. Gundala, S. Gertphol and V. Prasanna, “Greedy heuristics for resource allocation in dynamic distributed real-time heterogeneous computing systems,” *2002 International Conference on Parallel and Distributed Processing Technologies and Applications (PDPTA 2002)*, Vol. 2, June 2002, pp. 519–530.

[3] S. Ali, J. K. Kim, H. J. Siegel, and A. A. Maciejewski, “Static Heuristics for Robust Resource Allocation of Continuously Executing Applications,” *Journal of Parallel and Distributed Computing*, Vol. 68, No. 8, Aug. 2008, pp. 1070–1080.

[4] S. Ali, H. J. Siegel, M. Maheswaran, D. Hensgen, and S. Ali., “Representing task and machine heterogeneities for heterogeneous computing systems,” *Tamkang Journal of Science and Engineering*, Special 50th Anniversary Issue, invited, Vol. 3, No.3, Nov. 2000, pp.195–207.

[5] H. Barada, S. M. Sait, and N. Baig, “Task matching and scheduling in heterogeneous systems using simulated evolution,” 10th Heterogeneous Computing Workshop (HCW 2001), in the proceedings of the *15th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2001)*, Apr. 2001.

[6] I. Banicescu and V. Velusamy, “Performance of scheduling scientific applications with adaptive weighted factoring,” 10th Heterogeneous Computing Workshop (HCW 2001), in the proceedings of the *15th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2001)*, Apr. 2001.

[7] T. D. Braun, H. J. Siegel, N. Beck, L. Bölöni, R. F. Freund, D. Hensgen, M. Maheswaran, A. I. Reuther, J. P. Robertson, M. D. Theys, and B. Yao, “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, June 2001, pp. 810–837.

[8] T. D. Braun, H. J. Siegel, A. A. Maciejewski and Y. Hong, "Static mapping heuristics for tasks with dependencies, priorities, deadlines, and multiple versions in heterogeneous environments," *Journal of Parallel and Distributed Computing*, Vol. 68, No. 11, Nov. 2008, pp. 1504–1516.

[9] A. Burns, S. Punnekkat, B. Littlewood, and D. Wright, *Probabilistic Guarantees for Fault Tolerant Real-time Systems*, Technical Report Design for Validation (DeVa) TR No. 44, Esprit Long Term Research Project No. 20072, Department of Computer Science, University of Newcastle upon Tyne, UK, 1997.

[10] E. G. Coffman, ed., *Computer and Job-Shop Scheduling Theory*, John Wiley & Sons, New York, NY, 1976.

[11] M. K. Dhodhi, I. Ahmad, and A. Yatama, "An integrated technique for task matching and scheduling onto distributed heterogeneous computing systems," *Journal of Parallel and Distributed Computing*, Vol. 62, Sept. 2002, pp. 1338–1361.

[12] D. Fernandez-Baca, "Allocating modules to processors in a distributed system," *IEEE Transactions on Software Engineering*, Vol. 15, No. 11, Nov. 1989, pp. 1427–1436.

[13] S. Gertphol, Y. Yu, S. B. Gundala, V. K. Prasanna, S. Ali, J.-K. Kim, A. A. Maciejewski, and H. J. Siegel, "A metric and mixed-integer-programming-based approach for resource allocation in dynamic real-time systems," *16th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2002)*, Apr. 2002.

[14] A. Ghafoor and J. Yang, "A distributed heterogeneous supercomputing management system," *IEEE Computer*, Vol. 26, No. 6, June 1993, pp. 78–86.

[15] O. H. Ibarra and C. E. Kim, "Heuristic algorithms for scheduling independent tasks on nonidentical processors," *Journal of the ACM*, Vol. 24, No. 2, Apr. 1977, pp. 280–289.

[16] M. Kafil and I. Ahmad, "Optimal task assignment in heterogeneous distributed computing systems," *IEEE Concurrency*, Vol. 6, No. 3, July 1998, pp. 42–51.

[17] A. Khokhar, V. K. Prasanna, M. E. Shaaban, and C. Wang, "Heterogeneous computing: Challenges and opportunities," *IEEE Computer*, Vol. 26, No. 6, June 1993, pp. 18–27.

[18] J.-K. Kim, D. A. Hensgen, T. Kidd, H. J. Siegel, D. S. John, C. Irvine, T. Levin, N. W. Porter, V. K. Prasanna, and R. F. Freund, "A flexible multi-dimensional QoS performance measure framework for distributed heterogeneous systems," *Cluster Computing*, Special Issue on Cluster Computing in Science and Engineering, Vol. 6, No. 3, July 2006, pp. 281–296.

[19] M. Maheswaran, S. Ali, H. J. Siegel, D. Hensgen, and R. F. Freund, "Dynamic mapping of a class of independent tasks onto heterogeneous computing systems," *Journal of Parallel and Distributed Computing*, Vol. 59, No. 2, Nov. 1999, pp. 107–121.

[20] Z. Michalewicz and D. B. Fogel, eds., *How to Solve It: Modern Heuristics*, Springer-Verlag, New York, NY, 2000.

[21] V. Shestak, E. Chong, A. A. Maciejewski, and H. J. Siegel, "Robust Sequential Resource Allocation in Heterogeneous Distributed Systems with Random Compute Node Failures," 18th Heterogeneity in Computing Workshop (HCW 2009), in the proceedings of the *23rd IEEE International Parallel and Distributed Processing Symposium (IPDPS 2009)*, May 2009.

[22] H. Singh and A. Youssef, "Mapping and scheduling heterogeneous task graphs using genetic algorithms," *5th IEEE Heterogeneous Computing Workshop (HCW '96)*, 1996, pp. 86–97.

[23] M. Wu, W. Shu, and H. Zhang, "Segmented min-min: A static mapping algorithm for meta-tasks on heterogeneous computing systems," *9th IEEE Heterogeneous Computing Workshop*, Mar. 2000, pp. 375–385.

[24] D. Xu, K. Nahrstedt, and D. Wichadakul, "QoS and contention-aware multi-resource reservation," *Cluster Computing*, Vol. 4, No. 2, Apr. 2001, pp. 95–107.

[25] J. Yang, I. Ahmad, and A. Ghafoor, "Estimation of execution times on heterogeneous supercomputer architectures," *International Conference on Parallel Processing*, Vol. I, Aug. 1993, pp. 219–22.