# Modeling of Parameters in Supercomputer Workloads

Baiyi Song*, Carsten Ernemann†, Ramin Yahyapour
Computer Engineering Institute, University Dortmund, Germany
(email: {song.baiyi, carsten.ernemann, ramin.yahyapour}@udo.edu)

**Abstract:** Evaluation methods for parallel computers often require the availability of relevant workload information. To this end, workload traces recorded on real installations are frequently used. Alternatively, workload models are applied. However, often not all necessary information are available for a specific workload. In this paper, a model is presented to recover an estimated job execution time when this information is not available. The quality of the modelled estimated runtime is evaluated by comparing different workload traces for which this information is available.

## 1 Introduction

During the design process of a parallel computer and its management software, the evaluation of the system is an important task. Here, the availability of appropriate workloads is necessary for quantitative evaluations. Substantial information is required about the workload which is executed on these systems. Ideally, the exact workload is available and can be used during the design process. However, usually the exact workload for a certain system is not known during the design process. For parallel computing, several of such workload traces are available [SWF04]. In addition, models for workload representations exist which can also be used for the system design process [Do97a, Do97b, Fe96, JPF+97, LF01].

This paper deals with the problem that often not all available workload traces contain information about the estimated runtime of a computational job. However, many scheduling systems for parallel computers require that a user provides such information [FW98, LS02, Li95]. Research in the area of workload modelling provided some methods to model several workload parameters, as e.g. inter-arrival time, exact job run-time, required number of processors. However, the correlation with other workload information is usually neglected, as for instance the estimated runtime, user information, or application ID.

To this end, we present the analysis of the estimated runtime in correlation to other available parameters. Based on these results we propose a model to create estimations for the estimated runtime. Such a model can be used to create estimated runtimes for workloads which lack this information and subsequently makes more workloads available for the design and evaluation process in which this information is required.

## 2 Background

The design of a parallel computer system including its scheduling method usually requires the evaluation with suitable workloads. It is known that the system design highly depends

*Baiyi Song is a member of the Graduate School of Production Engineering and Logistics at the University of Dortmund.



†Carsten Ernemann is a member of the Collaborative Research Center 531, "Computational Intelligence", at the University of Dortmund with financial support of the Deutsche Forschungsgemeinschaft (DFG).

on the workload [ESY03]. Unfortunately, not one single scheduling method is best suited for all scenarios. Therefore, the evaluation and subsequently the workload selection are important tasks in the design process.

Currently, there is only a limited number of workload traces available which are recorded from real system installations. To this end, several workload models, e.g. [Do97a, Do97b, Fe96, JPF+97, LF01], have been derived from those traces to provide more flexibility. However, these models consider only some of the job parameters which exist on a real systems. Especially the estimated runtime of a job is neglected. Instead, only the actual job runtime is modelled and used for evaluation. However, the job runtime is usually not known at job submission on a real installation. But most scheduling systems which are actually in use on parallel machines require a user provided estimated job runtime. For instance, some scheduling algorithms use this information to estimate the maximum delay of queued jobs if a particular job is executed. Note, jobs exceeding their estimated runtime are usually killed after a grace time.

Cirne and Berman did consider the relations between the real runtime and estimated runtime in [CB01]. In their method the estimated runtime and the accuracy of the estimated runtime are modelled independently. Based on these two parameters the real runtime of the job is derived. Their model requires the availability of the estimated and real runtime in the underlying workload trace to determine these model parameters. However, for several existing workload traces the estimated runtime is missing, therefore a different model is required to deduce estimated runtimes for such traces.

## 3 Analysis

Seven workload traces have been examined for analyzing the characteristics of the estimated runtime. These workloads are publicly available on [SWF04]. Each contains several thousands of jobs which are submitted on the corresponding machine during a timeframe of several months, as shown in Table 1. Some of these traces include information about estimated runtimes as provided by the users at job submission. Cirne and Berman assumed

| Identifier | NASA | CTC | KTH | LANL | SDSC SP2 | SDSC 95 | SDSC 96 |
|---|---|---|---|---|---|---|---|
| Machine | iPSC/860 | SP2 | SP2 | CM-5 | SP2 | SP2 | SP2 |
| Period | 10/01/1993 12/31/1993 | 06/26/1996 05/31/1997 | 09/23/1996 08/29/1997 | 04/10/1994 09/24/1996 | 04/28/1998 04/30/2000 | 12/29/1994 12/30/1995 | 12/27/1995 12/31/1996 |
| Processors | 128 | 430 | 100 | 1024 | 128 | 416 | 416 |
| Jobs | 42264 | 79302 | 28490 | 201378 | 67667 | 76872 | 38719 |
| Estimated runtime | no | yes | yes | partially | yes | no | no |

Table 1: Workloads used in this research.

in [CB01] that a job cannot run longer than the estimated runtime. However, this is not true for all traces as shown in Table 2. The table shows the *accuracy* of the user estimated runtime which is defined in Equation 1. The table also shows the *squashed area* which is the sum of the resource consumptions of all jobs (see Equation 2).

$$\text{accuracy} = \frac{\text{real runtime}}{\text{estimated runtime}} \quad (1)$$

$$\text{squashed area} = \sum_{j \in \text{Jobs}} \text{req\_processors}_j \cdot \text{run\_time}_j \quad (2)$$

We can see that more than 10% of all jobs are exceeding their estimated runtime in the SP2, CTC and LANL workloads. Thereby, these jobs cannot be neglected as they account for more than 20% of the overall workload. However, we can also see that on average, with the exception of LANL, not many jobs exceed their runtime by more than 10%. We neglect jobs with an accuracy $> 1.1$ as they do not account for much amount of workload for the corresponding trace. For LANL, however, more than 20% of the workload is caused by jobs with an accuracy $> 1.1$. Therefore, we have to consider these jobs for this particular workload trace.

| | accuracy $> 1$ | | accuracy $> 1.1$ | |
|---|---|---|---|---|
| Traces | percentage of jobs | squash area | percentage of jobs | squash area |
| KTH | 1% | 2% | 0.2% | 1.1% |
| SP2 | 10% | 26% | 1.1% | 1.1% |
| CTC | 16% | 20% | 0.8% | 1.2% |
| LANL | 16% | 30% | 7% | 22% |

Table 2: Analysis of jobs exeeding the estimated runtime.

Note, that the estimated runtime is given in seconds. However, most users cannot provide exact information about the job execution time.

In Table 3 the estimated runtimes for the workloads KTH, SDSC SP2 and CTC are organized into 20 groups. Those groups are build up from the most frequently provided estimated runtimes by the users. As the results indicate most of the jobs fit to those 20 groups (about 80 % of all jobs). This indicates either that the users provide rounded estimates or that the estimated runtime is related to the configuration of available system queues with certain job default values. In the appendix the corresponding groups for the separate workloads KTH, SDSC SP2, CTC and LANL are presented.

| Group | Requested runtime | Percentage of all jobs | Group | Requested runtime | Percentage of all jobs |
|---|---|---|---|---|---|
| 1 | 1 min | 0.9 | 11 | 3 hours | 3.9 |
| 2 | 5 mins | 9.7 | 12 | 3.33 hours | 0.8 |
| 3 | 10 mins | 7.0 | 13 | 4 hours | 5.0 |
| 4 | 15 mins | 6.8 | 14 | 5 hours | 1.2 |
| 5 | 20 mins | 3.2 | 15 | 6 hours | 4.8 |
| 6 | 30 mins | 4.0 | 16 | 8 hours | 1.9 |
| 7 | 1 hour | 6.3 | 17 | 10 hours | 2.3 |
| 8 | 1.5 hours | 0.8 | 18 | 12 hours | 2.5 |
| 9 | 2 hours | 5.0 | 19 | 15 hours | 1.1 |
| 10 | 2.5 hours | 0.8 | 20 | 18 hours | 14.4 |

Table 3: Summarized alignment of all job within the examined workloads, Total Alignment: 82.67%.

In order to model the accuracy we found that for the workloads of KTH, SP2, CTC a Beta distribution can be used [De86]. Note, the general formula for the Beta distribution function in Equation 3 where p and q are the shape parameters, a and b denote the bounds of the distribution. The beta function is defined in Equation 4. Contrary, for LANL a Gamma distribution is more suitable. The gamma distribution function is defined in Equation 5, where Equation 6 describes the gamma function.

$$f(x) = \frac{(x-a)^{p-1}(b-x)^{q-1}}{B(p,q)(b-a)^{p+q-1}}, a \leq x \leq b; p,q > 0 \quad (3)$$

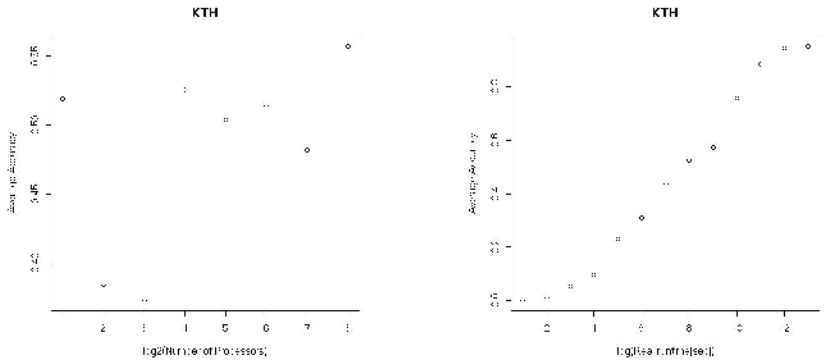


Figure 1: The relations of the accuracy to the number of requested processors (left) and the real runtime (right).

$$B(\alpha,\beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt \tag{4}$$

$$f(x) = \frac{\left(\frac{x-\mu}{\beta}\right)^{\gamma-1}\cdot e^{\left(-\frac{x-\mu}{\beta}\right)}}{\beta\Gamma(\gamma)}, x \geq \mu; \gamma,\beta > 0 \tag{5}$$

$$\Gamma(a) = \int_0^1 t^{a-1}e^{-t}dt \tag{6}$$

It can be seen from the traces that the accuracy depends on the real runtime, as shown in Figure 1. However, Figure 1 for instance shows that the accuracy is not related to the requested number of processors. Similarly, there is no correlation to other available job parameters, as e.g. used memory. Therefore, it is sufficient to include the relation of estimated to real runtime into the modelling.

However, we cannot utilize the method from [CB01], in which the real runtime is modelled by (real runtime $=$ estimated runtime $\cdot$ accuracy) with an independent model for the estimated runtime and accuracy. In our scenario, an integrated model is required for real runtime and estimated runtime.

## 4 Modelling

In the following, we present a method to model the estimated runtimes if they are not available in a trace. To this end, we examine the mentioned workloads with available estimated runtimes to determine suitable modelling parameters.

As mentioned before, we can use the Beta distribution [De86] to model the accuracy distribution for the KTH, CTC and SDSC SP2 workloads. Figure 2 shows the actual and synthetic distributions. The LANL workload is modelled using the Gamma distribution.

However, this distribution model does not consider the relation of accuracy to the real runtime, as shown in Figure 1. This dependency is important as some scheduling algorithms like Backfilling are especially sensitive to the variations of estimated runtimes for jobs requiring a small number of processors [FW98, LS02, Li95]. Therefore, an extended model for the parameterizations of the Beta distribution function is established. In this model the

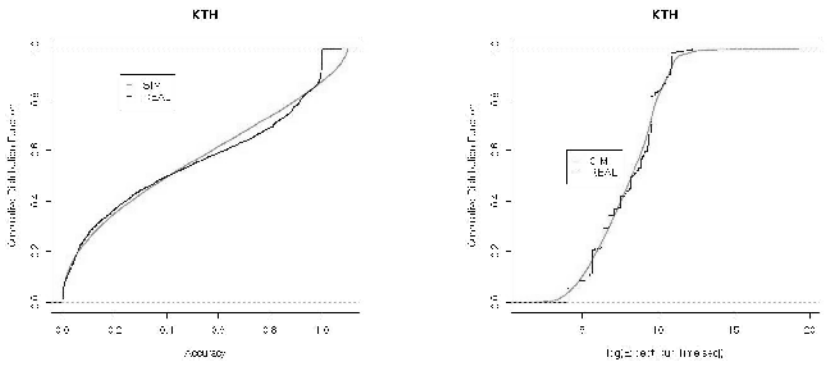


Figure 2: Comparison of the real (REAL) with the synthetic (SIM) accuracy (left) and runtime (right) for KTH.

jobs of each workload are grouped by their runtime. For each of these groups of the different workloads, parameter combination of p and q are identified in order to maintain better similarity between the original and the synthetically generated workload. That is, for each group the values $p = f(\text{real runtime})$ and $q = f(\text{real runtime})$ can be found. The distribution functions are derived from the combination of these parameters by a linear regression. In detail the process works as follows:

**Step 1:** In the first step all jobs are grouped by their real runtime. The jobs are grouped by calculating the integer of the logarithm of the real runtime. In our examples, up to 13 different groups for the different workloads are created. This allows the examination of the influence of the parameters p and q in a smaller subset of all jobs, as the run times vary in a wide range. That is, the runtimes span from 1 to $10^7$ seconds. For each of these subsets a separate combination of the parameters p and q is generated.

In detail a group $G_i$ is build as follows:

$$G_i = \{x | [log(\text{real runtime}_x)] = r_i; i \in [1, k]; r_i \in \mathbb{N}\}$$

and a workload consists of $k$ groups. In the following we refer to the real runtime as RRT.

**Step 2:** For each group $G_i$ a combination of p and q can be found for which the Beta distribution resembles the original workload. The results in Figure 3 present the relation between p, q and the runtime for the KTH workload are shown. As it can be seen the parameters are not constant but p generally increases and q decreases with an increasing group number (created depending on their runtime). All other workloads have a similar behavior.

The parameters p and q are yet only described in a qualitative fashion. The next step includes the determination of specific values for p and q depending on the real runtimes. To this end we used a linear regression model for these two parameters. The parameters can be described as follows: $log(p) = log(RRT) \cdot a_1 + b_1$ and $log(log(q)) = log(RRT) \cdot a_2 + b_2$. For q we used the log transformation twice as the results indicated a better fitting. For each group $G_i$ a Beta distribution with specific $p_i$ and $q_i$ exists. So $k$ pairs $(r_i, p_i)$ and $(r_i, q_i)$ can be used to derive the parameters $(a_1, b_1)$ for p and $(a_2, b_2)$ for q. The accuracy can be created for any given real runtime using the resulting functions for $p$ and $q$. The estimated runtime can easily be extracted by using this accuracy and the real runtime itself.

Figure 2 shows the cumulative distribution function (CDF) using the described method to derive the estimated runtime. Here the KTH workload is presented, but all examined workloads show a similar behavior. The distance between the artificially generated and the original accuracy can still be reduced. The results in the figure indicate that an alignment of

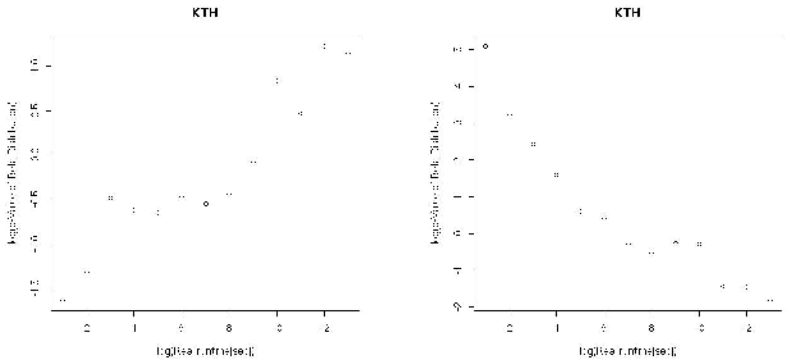


Figure 3: The relations between the real runtime and the parameters of the Beta distributions for KTH. Note, the relations are similar for the traces of SDSC SP2 and CTC.

the estimated runtimes can be used to improve the quality of our modelling, see Section 3.

To this end the transformation method is extended by including an alignment process. From the available workloads we can see in Tables 3 that between 70 and 95% of all jobs are aligned to some round values. Therefore, in our alignment modelling we decided that with a probability of 80% the estimated runtime of a job is aligned to the nearest value according to the rule in Table 4.

As the synthetically generated estimated runtime might be much higher, the synthetically generated squashed area of the different workloads would be much higher than in the original traces. Existing scheduling strategies like Backfilling [Li95] are sensitive to those kinds of changes. To this end, the estimated runtime is bounded. Note, the runtime of all jobs is unchanged and so the squashed area of all jobs of the different workloads is not affected. The bound of the estimated runtime is selected from the highest value of all jobs within the existing workloads. The maximum estimated runtimes of the examined workloads are presented in Table 4. Therefore, we have chosen an upper bound of 60 hours for the synthetically generated estimated runtime.

| Estimated runtime before alignment | aligned to multiples of |
|---|---|
| 1 - 5 min | 1 min |
| 5 min - 60 min | 5 min |
| 1h - 4h | 20 min |
| $>$ 4h | 1h |

| | KTH | SP2 | CTC | LANL |
|---|---|---|---|---|
| Maximum | 60 | 48 | 18 | 8.33 |

Table 4: Alignment of the modelled estimated runtime (left) and the maximum estimated runtime (hours) of the different traces (right).

As mentioned in Section 3, the LANL workload has a wider range than all the other workloads and so a Beta distribution is not suitable. Therefore, a modified procedure has been used for this workload. We have chosen a Gamma distribution [De86] to model the accuracy for this workload. Here the maximum estimated runtime for the alignment process is selected by 8.33 hours (Table 4). The process of parameterizing the Gamma distribution is similar to the method as described above; again, two parameters need to be extracted, this time $\gamma$ and $\beta$ instead of $p$ and $q$. Also the final selection of a reasonable estimated runtime is done by using the same algorithm as described for the other three workloads.

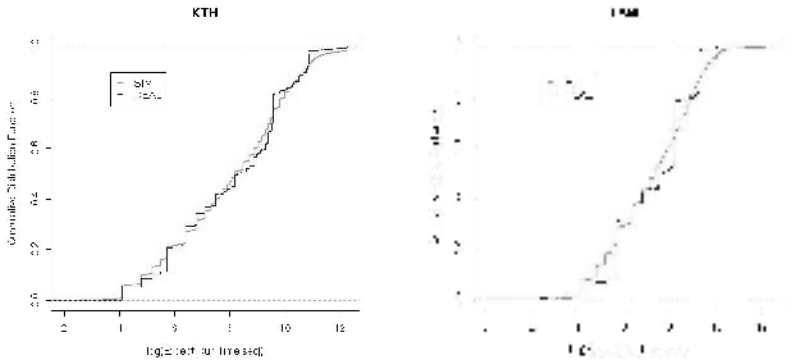


Figure 4: The comparison of synthetic estimated runtime (SIM) and real estimated runtime (REAL) for KTH (left) and LANL (right).

## 5 Evaluation

To measure the distance between the synthetic and original estimated runtime, we use the Kolmogorov-Smirnov [Pa84] test (KS). Furthermore, we calculate the difference of squashed area (SA) by

$$d_{\text{SA}} = \frac{\text{synthetic SA} - \text{original SA}}{\text{original SA}}.$$

In our experimental workloads, we have examined the KTH, SP2 and CTC. We used the traces to train the model and feed the model with their own real runtime and compare the estimated runtime of the same traces. The cumulative distribution functions (CDF) of the synthetic and real estimated runtimes are plotted in Figure 4. The results of the KS tests and the comparison of the squashed areas of the workloads are shown in Table 5.

| | KTH | SP2 | CTC |
|---|---|---|---|
| KS Test | 0.06 | 0.14 | 0.15 |
| $d_{\text{SA}}$ | 15% | -9% | 34% |

Table 5: Comparison of KS test results and difference of squashed area (SA).

The KS test shows that our synthetic and original estimated runtimes are similar. However, the results based on the comparison of the squashed areas are different for CTC. Here, the generated workload has a squashed area which is 34% higher than for the original workload. This is caused by our assumption of an upper bound for the estimated runtime of a job. As can be seen from the original workload trace of CTC in Table 4, the upper bound was much lower than 60 hours. In order to increase the precision of the estimation a tighter upper bound is needed for the estimated runtime in the alignment process.

Next, we examine whether a general model can be derived to recover the estimated runtimes for traces in which they are missing. That is, we consider the case where we cannot train the model and subsequently compare the results for a particular workload trace with itself. Therefore, we trained the model with KTH, SP2 and CTC and combinations of them. Finally, the synthetic estimated runtime is compared with the original estimated runtime of the workloads, as shown in Tables 6. The KS test shows that the combination of KTH, SP2 and CTC is suitable to train the model for all 3 workload traces. Independently of the training workload, the original squashed area of the estimated workload is much smaller for KTH. This is due to the fact that the overall accuracy of the estimated runtime is con-

siderably better than in all other workloads. Based on the averaging effect by training with other workload traces, the lower modelled accuracy yields a higher estimated runtime.

| | | SA comparison | | | KS test results | | |
|---|---|---|---|---|---|---|---|
| | | KTH | SP2 | CTC | KTH | SP2 | CTC |
| Training Workloads | KTH | 15% | −28% | −30% | 0.06 | 0.18 | 0.18 |
| | SP2 | 40% | −9% | −10% | 0.10 | 0.14 | 0.15 |
| | CTC | 144% | 40% | 35% | 0.22 | 0.13 | 0.15 |
| | KTH,SP2 | 49% | −12% | −16% | 0.11 | 0.13 | 0.14 |
| | KTH,CTC | 56% | −1% | −5% | 0.13 | 0.11 | 0.13 |
| | SP2,CTC | 96% | 15% | 7% | 0.17 | 0.10 | 0.11 |
| | KTH,SP2,CTC | 61% | −2% | −7% | 0.12 | 0.12 | 0.13 |

Table 6: The comparison of synthetic estimated runtime and real estimated runtime

In our example, a decision must be made whether a workload resembles similarities with a certain other workload and train the model accordingly. For instance, whether KTH or SP2 or CTC is better suited to examine a workload and train the model accordingly to recover the missing estimated runtimes.

For LANL, about 20% of the jobs have estimated runtime information available, while it is missing for all other jobs. Therefore we trained our model with those 20% of entries and used the model to recover the others. The CDF of synthetic and real estimated runtime are plotted in Figure 4. We see acceptable results in regards to a KS test of 0.13, and a difference of the squashed area of -7%.

The workloads generated by the presented model are online available [Pas04].

## 6 Conclusion

In this paper, we presented a model to recover an estimated job execution time for workload traces without such information. This value is often used in scheduling strategies and therefore necessary in the evaluation process. Statistical criteria as e.g. given by the Kolmogorov-Smirnov test showed that the model produces good results in comparison to the original traces. The evaluation also showed that the parameterization of the model varies for different workloads. While some parameter sets could be found to produce good results for several workloads, a general model could not be found. The quality of the modelling depends on the similarity of the workload used for training and the workload scenario for which estimated runtimes are modelled. The presented method to model the estimated runtimes is an example for recovering missing workload parameters. Similarly, the method can be applied to other parameters.

## References


[CB01] Cirne, W. und Berman, F.: A comprehensive model of the supercomputer workload. In: *4th Workshop on Workload Characterization*. Dec 2001.

[De86] DeGroot, M. H.: *Probability and Statistics*. Addison-Wesley Series in Statistics. Addison-Wesley. Reading, MA, USA. 2nd. 1986.

[Do97a] Downey, A. B.: A parallel workload model and its implications for processor allocation. In: *6th Intl. Symp. High Performance Distributed Comput.* Aug 1997.

[Do97b] Downey, A. B.: Using queue time predictions for processor allocation. In: Feitelson, D. G. und Rudolph, L. (Hrsg.), *Job Scheduling Strategies for Parallel Processing*. S. 35–57. Springer Verlag. 1997. Lect. Notes Comput. Sci. vol. 1291.

[ESY03] Ernemann, C., Song, B., und Yahyapour, R.: Scaling of workload traces. In: Feitelson, D. G., Rudolph, L., und Schwiegelshohn, U. (Hrsg.), *Job Scheduling Strategies for Parallel Processing: 9th International Workshop, JSSPP 2003 Seattle, WA, USA, June 24, 2003*. volume 2862 of *Lecture Notes in Computer Science (LNCS)*. S. 166–183. Springer-Verlag Heidelberg. October 2003.

[Fe96] Feitelson, D. G.: Packing schemes for gang scheduling. In: Feitelson, D. G. und Rudolph, L. (Hrsg.), *Job Scheduling Strategies for Parallel Processing*. S. 89–110. Springer-Verlag. 1996. Lect. Notes Comput. Sci. vol. 1162.

[FW98] Feitelson, D. G. und Weil, A. M.: Utilization and predictability in scheduling the ibm sp2 with backfilling. In: *12th Intl. Parallel Processing Symp.* S. 542–546. Apr 1998.

[JPF+97] Jann, J., Pattnaik, P., Franke, H., Wang, F., Skovira, J., und Riodan, J.: Modeling of workload in MPPs. In: Feitelson, D. G. und Rudolph, L. (Hrsg.), *Job Scheduling Strategies for Parallel Processing*. S. 95–116. Springer Verlag. 1997. Lect. Notes Comput. Sci. vol. 1291.

[LF01] Lublin, U. und Feitelson, D. G.: The workload on parallel supercomputers: Modeling the characteristics of rigid jobs. Technical Report 2001-12. Hebrew University. Oct 2001.

[Li95] Lifka, D.: The ANL/IBM SP Scheduling System. In: Feitelson, D. und Rudolph, L. (Hrsg.), *IPPS'95 Workshop: Job Scheduling Strategies for Parallel Processing*. S. 295–303. Springer–Verlag, Lecture Notes in Computer Science LNCS 949. 1995.

[LS02] Lawson, B. G. und Smirni, E.: Multiple-queue backfilling scheduling with priorities and reservations for parallel systems. In: Feitelson, D. G., Rudolph, L., und Schwiegelshohn, U. (Hrsg.), *Job Scheduling Strategies for Parallel Processing*. S. 72–87. Springer Verlag. 2002. Lect. Notes Comput. Sci. vol. 2537.

[Pa84] Papoulis, A.: *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill. New York, NY, USA. 2nd. 1984.

[Pas04] Webpage with workload traces including the modelled estimated runtime used in this paper. http://www-ds.e-technik.uni-dortmund.de/∼rmg/Pasa2004/. March 2004.

[SWF04] Parallel Workloads Archive. http://www.cs.huji.ac.il/labs/parallel/workload/. March 2004.

# A Appendix: Complete Results

| Group | LANL | | KTH | |
|---|---|---|---|---|
| | Requested runtime | Percentage of all jobs | Requested runtime | Percentage of all jobs |
| 1 | 1 min | 4.6 | 1 min | 5.3 |
| 2 | 2 mins | 0.9 | 2 mins | 3.3 |
| 3 | 3 mins | 1.0 | 3 mins | 1.7 |
| 4 | 5 mins | 23.3 | 5 mins | 9.3 |
| 5 | 6 mins | 0.3 | 10 mins | 7.8 |
| 6 | 10 mins | 5.6 | 15 mins | 4.3 |
| 7 | 15 mins | 3.8 | 20 mins | 2.5 |
| 8 | 20 mins | 2.0 | 30 mins | 4.5 |
| 9 | 29 mins | 0.9 | 1 hour | 4.7 |
| 10 | 30 mins | 4.7 | 1.67 hours | 1.1 |
| 11 | 40 mins | 0.9 | 2 hours | 3.7 |
| 12 | 50 mins | 1.1 | 2.5 hours | 1.1 |
| 13 | 59 mins | 0.8 | 3 hours | 2.0 |
| 14 | 1 hour | 25.8 | 3.33 hours | 4.1 |
| 15 | 2 hours | 0.7 | 3.83 hours | 2.6 |
| 16 | 2.33 hours | 0.6 | 4 hours | 10.1 |
| 17 | 2.5 hours | 0.9 | | |
| 18 | 3 hours | 16.5 | | |
| 19 | 4 hours | 0.3 | | |
| 20 | 6 hours | 1.0 | | |
| Total alignment | 96 % | | 74.9% | |

| Group | CTC | | SDSC SP2 | |
|---|---|---|---|---|
| | Requested runtime | Percentage of all jobs | Requested runtime | Percentage of all jobs |
| 1 | 5 mins | 8.6 | 5 mins | 11.6 |
| 2 | 10 mins | 6.2 | 10 mins | 7.8 |
| 3 | 15 mins | 10.4 | 12 mins | 1.1 |
| 4 | 20 mins | 1.9 | 15 mins | 2.9 |
| 5 | 30 mins | 3.4 | 20 mins | 5.5 |
| 6 | 1 hour | 4.1 | 30 mins | 4.6 |
| 7 | 1.5 hours | 0.8 | 45 mins | 1.0 |
| 8 | 2 hours | 5.3 | 1 hour | 10.3 |
| 9 | 3 hours | 4.8 | 2 hours | 5.1 |
| 10 | 4 hours | 2.1 | 2.5 hours | 1.2 |
| 11 | 4.83 hours | 0.6 | 3 hours | 3.7 |
| 12 | 5 hours | 1.1 | 4 hours | 6.4 |
| 13 | 6 hours | 8.5 | 5 hours | 1.4 |
| 14 | 8 hours | 1.5 | 6 hours | 1.9 |
| 15 | 10 hours | 1.7 | 7 hours | 0.9 |
| 16 | 12 hours | 2.2 | 8 hours | 3.3 |
| 17 | 15 hours | 1.4 | 10 hours | 3.2 |
| 18 | 16 hours | 1.0 | 12 hours | 3.9 |
| 19 | 17 hours | 0.6 | 15 hours | 0.8 |
| 20 | 18 hours | 23.1 | 18 hours | 9.4 |
| Total alignment | 89.15% | | 86.22% | |

Table 7: Alignment of all job within the examined workloads.