Scaling of Workload Traces

Carsten Ernemann, Baiyi Song, and Ramin Yahyapour

Computer Engineering Institute, University Dortmund
44221 Dortmund, Germany
e-mail: {carsten.ernemann, song.baiyi, ramin.yahyapour}@udo.edu

Abstract. The design and evaluation of job scheduling strategies often require simulations with workload data or models. Usually workload traces are the most realistic data source as they include all explicit and implicit job patterns which are not always considered in a model. In this paper, a method is presented to enlarge and/or duplicate jobs in a given workload. This allows the scaling of workloads for later use on parallel machine configurations with a different number of processors. As quality criteria the scheduling results by common algorithms have been examined. The results show high sensitivity of schedule attributes to modifications of the workload. To this end, different strategies of scaling number of job copies and/or job size have been examined. The best results had been achieved by adjusting the scaling factors to be higher than the precise relation between the new scaled machine size and the original source configuration.

1 Introduction

The scheduling system is an important component of a parallel computer. Here, the applied scheduling strategy has direct impact to the overall performance of the computer system with respect to the scheduling policy and objective. The design of such a scheduling system is a complex task which requires several steps, see [13]. The evaluation of scheduling algorithms is important to identify the appropriate algorithm and the corresponding parameter settings. The results of theoretical worst-case analysis are only of limited help as typical workloads on production machines do normally not exhibit the specific structure that will create a really bad case. In addition, theoretical analysis is often very difficult to apply to many scheduling strategies. Further, there is no random distribution of job parameter values, see e.g. Feitelson and Nitzberg [9]. Instead, the job parameters depend on several patterns, relations, and dependencies. Hence, a theoretical analysis of random workloads will not provide the desired information either. A trial and error approach on a commercial machine is tedious and significantly affects the system performance. Thus, it is usually not practicable to use a production machine for the evaluation except for the final testing. This just leaves simulation for all other cases.

Simulations may either be based on real trace data or on a workload model. Workload models, see e.g. Jann et al. [12] or Feitelson and Nitzberg [9], enable a wide range of simulations by allowing job modifications, like a varying amount of assigned processor resources. However, many unknown dependencies and patterns may cause the actual workload of a real system. This is especially true as the characteristics of an workload usually change over time; beginning from daily or weekly cycles to changes in the job submissions during a year and the lifetime of a parallel machine. Here, the consistence of a statistical generated workload model with real workloads is difficult to guarantee. On the other hand, trace data restrict the freedom of selecting different configurations and scheduling strategies as a specific job submission depends on the original circumstances. The trace is only valid on a similar machine configuration and the same scheduling strategy. For instance, trace data taken from a 128 processor parallel machine will lead to unrealistic results on a 256 processor machine. Therefore, the selection of the underlying data for the simulation depends on the circumstances determined by the MPP architecture as well as the scheduling strategy. A variety of examples already exists for evaluations via simulation based on a workload model, see e.g. Feitelson [5], Feitelson and Jette [8] or on trace data, see e.g. Ernemann et al. [4].

Our research on job scheduling strategies for parallel computers as well as for computational Grid environments led to the requirement of considering different resource configurations. As the individual scheduling objectives of users and owners is of high importance in this research, we have to ensure that the workload is very consistent with real demand. To this end, statistical distribution of the various parameters without the detailed dependencies between them cannot be applied. Therefore, real workload traces have been chosen as the source for our evaluations. In this paper, we address the question how workload traces can be transformed to be used on different resource configurations while retaining important specifics. In Section 2 we give a brief overview on previous works in workload modelling and analysis. In addition, we discuss our considerations for choosing a workload for evaluation. Our approach and the corresponding results are presented in Section 3. Finally, we conclude this paper with a brief discussion on the important key observations in Section 4.

2 Background

We consider on-line parallel job scheduling in which a stream of jobs is submitted to a job scheduler by individual users. The jobs are executed in a space-sharing fashion for which a job scheduling system is responsible to decide when and on which resource set the jobs are actually started. A job is first known by the system at its submission time. The job description contains information on its requirements as e.g. number of processing nodes, memory or the estimated execution length.

For the evaluation of scheduling methods it is a typical task to choose one or several workloads for simulations. The designer of a scheduling algorithm must ensure that the workload is close to a real user demand in the examined scenario. Workload traces are recorded on real systems and contain information on the job requests including the actual start and execution time of the job. Extensive research has been done to analyze workloads as well as to propose corresponding workload models, see e.g. [7, 3, 2, 1].

Generally, statistical models use distributions or a collection of distributions to describe the important features of real workload attributes and the correlations among them. Then synthetic workloads are generated by sampling from the probability distributions [12, 7]. Statistical workload models have the advantage that new sets of job submissions can be generated easily. The consistence with real traces depends on the knowledge about the different examined parameters in the original workload. Many factors contribute to the actual process of workload generation on a real machine. Some of them are known, some are hidden and hard to deduce. It is difficult to find rules for job submissions by individual users. The analysis of workloads shows several correlations and patterns of the workload statistics. For example, jobs on many parallel computers require job sizes of a power of two [15, 5, 16]. Other examples are the job distribution during the daily cycle obviously caused by the individual working hours of the users, or the job distribution of different week days. Most approaches consider the different statistical moments isolated. Some correlations are included in several methods. However, it is very difficult to identify whether the important rules and patterns are extracted. In the same way it is difficult to tell whether the inclusion of the result is actually relevant to the the evaluation and therefore also relevant for the design of an algorithm.

In general, only a limited number of users are active on a parallel computer, for instance, several dozens. Therefore, for some purposes it is not clear if a given statistical model comes reasonable close to a real system. For example, some workload traces include singular outliers which significantly influence the overall scheduling result. In this case, a statistical modelling without this outlier might significantly deviate from the real world result. In the same way, it may make a vast difference to have several outliers of the same or similar kind. The relevance to the corresponding evaluation is difficult to judge, but this also renders the validity of the results undefined.

Due to the above mentioned reasons, it emerged to be difficult to use statistical workload models for our research work. Therefore, we decided to use workload traces for our evaluations. The standard parallel workload archive [19] is a good source for job traces. However, the number of available traces is limited. Most of the workloads are observed on different supercomputers. Mainly, the total number of available processors differs in those workloads. Therefore, our aim was to find a reasonable method to scale workload traces to fit on a standard supercomputer. However, special care must be taken to keep the new workload as consistent as possible to the original trace. To this end, criteria for measuring the validity had to be chosen for the examined methods for scaling the workload.

The following well-known workloads have been used: of the CTC [11], the NASA [9], the LANL [6], the KTH [17] and three workloads from the SCSD [20]. All traces are available from the Parallel Workload Archive, see [19]. As shown in

Table 1, the supercomputer from the LANL has the highest number of processors from the given computers and so this number of processors was chosen as the standard configuration. Therefore the given workload from the LANL does not need to be modified and as a result the following modification will only be applied to the other given workloads.

Workload	CTC	NASA	KTH	LANL	SDSC95	SDSC96	SDSC00
Number of jobs	79302	42264	28490	201387	76872	38719	67667
Number of nodes	430	128	100	1024	416	416	128
Size of the biggest job	336	128	100	1024	400	320	128
Static factor f	3	8	10	1	3	3	8

Table 1. The Examined Original Workload Traces.

In comparison to statistical workload models, the use of actual workload traces is simpler as they inherently include all submission patterns and underlying mechanisms. The traces reflect the real workload exactly. However, it is difficult to perform several simulations as the data basis is usually limited. In addition, the applicability of workload traces to other resource configuration with a different number of processors is complicated. For instance, this could result in a too high workload and an unrealistic long wait time for a job. Or, contrary, the machine is not fully utilized if the amount of computational work is too low. However, it is difficult to change any parameter of the original workload trace as it has an influence on its overall validity. For example, the reduction of the inter-arrival time destroys the distribution of the daily cycle. Therefore, modifications on the job length are inappropriate. Modifications of the requested processor number of a job change the original job size distribution. For instance, we might invalid an existing preference of jobs with a power of 2 processor requirement. In the same way, an alternative scaling of the number of requested processors by a job would lead to an unrealistic job size submission pattern. For example, scaling a trace taken from a 128 node MPP system to 256 node system by just duplicating each job preserves the temporal distribution of job submissions. However, this transformation leads also to an unrealistic distribution as no larger jobs are submitted.

Note, that the scaling of a workload to match a different machine configuration always alters the original distribution whatsoever. Therefore, as a trade-off special care must be taken to preserve original time correlations and job size distribution.

3 Scaling Workloads to a Different Machine Size

The following 3 sections present the examined methods to scale the workload. We briefly discuss the different methods as the results of each step motivated the next.

First, it is necessary to select quality criteria for comparing the workload modifications. Distribution functions could be used to compare the similarity of the modified with the corresponding original workloads. This method might be valid, however, it is unknown whether the new workload has a similar effect on the resulting schedule as the original workload.

As mentioned above, the scheduling algorithm that has been used on the original parallel machine also influences the submission behavior of the users. If a different scheduling system is applied and causes different response times, this will most certainly influence the submission pattern of later arriving jobs. This is a general problem [3, 1] that has to be kept in mind if workload traces or statistical models are used to evaluate new scheduling systems. This problem can be solved if the feedback mechanisms of prior scheduling results on new job submissions is known. However, such a feedback modelling is a difficult topic as the underlying mechanisms vary between individual users and between single jobs.

For our evaluation, we have chosen the Average Weighted Response Time (AWRT) and the Average Weighted Wait Time (AWWT) generated by the scheduling process. Several other scheduling criteria, for instance the slowdown, can be derived from AWRT and AWWT. To match the original scheduling systems, we used First-Come-First-Serve [18] and EASY-Backfilling [17, 14] for generating the AWRT and AWWT. These scheduling methods are well known and used for most of the original workloads. Note, that the focus of this paper is not to compare the quality of both scheduling strategies. Instead, we use the results of each algorithm to compare the similarity of each modified workload with the corresponding original workload.

The definitions (1) to (3) apply whereas index j represents job j.

$$\text{Resource\_Consumption}_j = \Big(\text{requestedResources}_j \cdot (\text{endTime}_j - \text{startTime}_j)\Big) \tag{1}$$

$$\text{AWRT} = \frac{\sum\limits_{j \in \text{Jobs}} \Big(\text{Resource\_Consumption}_j \cdot (\text{endTime}_j - \text{submitTime}_j)\Big)}{\sum\limits_{j \in \text{Jobs}} \text{Resource\_Consumption}_j} \tag{2}$$

$$\text{AWWT} = \frac{\sum\limits_{j \in \text{Jobs}} \Big(\text{Resource\_Consumption}_j \cdot (\text{startTime}_j - \text{submitTime}_j)\Big)}{\sum\limits_{j \in \text{Jobs}} \text{Resource\_Consumption}_j} \tag{3}$$

In addition, the makespan is considered, which is the end time of the last job within the workload. The Squashed Area is given as a measurement for the amount of consumed processing power for the workloads which is defined in (4).

$$\text{Squashed\_Area} = \sum_{j \in \text{Jobs}} \text{Resource\_Consumption}_j \tag{4}$$

Note, that in the following we refer to jobs with a higher number of requested processor as *bigger* jobs, while calling jobs with a smaller demand in processor number as *smaller jobs* respectively.

Scaling only the number of requested processors of a job results in the problem that the whole workload distribution is transformed by a factor. In this case the modified workload might not contain jobs requesting 1 or a small number of processors. In addition, the favor of jobs requesting a power of 2 processors is not modelled correctly for most scaling factors. Alternatively, the number of jobs can be scaled. Each original job is duplicated to several jobs in the new workload. Using only this approach has the disadvantage that the new workload has more smaller jobs in relation to the original workload. For instance, if the biggest job in the original workload uses the whole machine, a duplication of each job for a machine with twice the number of processors leads to a new workload in which no job requests the maximum number of processors at all.

3.1 Precise Scaling of Job Size

Based on the considerations above, a factor f is calculated for combining the scaling of the requested processor number of each job with the scaling of the total number of jobs. In Table 1 the requested maximum number of processors requested by a job is given as well as the total number number of available processors.

As explained above multiplying solely the number of processors of a job or the number of jobs by a constant factor is not reasonable. Therefore, the following combination of both strategies has been applied. In order to analyze the influence of both possibilities the workloads were modified by using a probabilistic approach: a probability factor p is used to specify whether the requested number of processors is multiplied for a job or copies of this job are created. During the scaling process each job of the original workload is modified by only one of the given alternatives. A random value between 0 and 100 is generated for probability p. A decision value d is used to discriminate which alternative is applied for a job. If p produced by the probabilistic generator is greater d the number of processors is scaled for the job. Otherwise, f identical, new job are included in the new workload. So, if d has a greater value, the system prefers the creation of smaller jobs while resulting in less bigger jobs otherwise.

As a first approach, integer scaling factors had been chosen based on the relation to a 1024 processor machine. We restricted ourselves to integer factors as it would require additional considerations to model fractional job parts. For the KTH a factor f of 10 is chosen, for the NASA and the SDSC00 workloads a factor of 8 and for all other workloads a factor of 3. Note, that for the SDSC95 workload one job yields more than 1024 processors if multiplied by 3. Therefore, this single job is reduced to 1024.

For the examination of the influence of d, we created 100 modified workloads for each original workload with d between 0 and 100. However, with exception to the NASA traces, our method did not produce satisfying results for the workload scaling. The imprecise factors increased the overall amount of workload at most 26% which lead to a jump of several factors for AWRT and AWWT. This shows how important the precise scaling of the overall amount of workload is. Second, if the chosen factor f is smaller than the precise scaling factor the workloads which prefer smaller jobs scale better than the workloads with bigger jobs. If f is smaller or equal to the precise scaling factor, the modified workloads scale better for smaller values of d.

Based on these results, we introduced a precise scaling for the job size. As the scaling factors for the workloads CTC, KTH, SDSC95 and SDSC96 are not integer values an extension to the previous method was necessary. In the case that a single large job is being created the number of jobs is multiplied by the precise scaling factor and rounded.

The scheduling results for the modified workloads are presented in Table 2. Only the results for the original workload (ref) and the modified workloads with the parameter settings of $d = \{1, 50, 99\}$ are shown. Now the modified CTC based workloads are close to the original workloads in terms of AWWT, AWRT and utilization if only bigger jobs are created ($d = 1$). For increasing values for d, also AWRT, AWWT and utilization increase. Overall, the results are closer to the original results in comparison to using an integer factor. A similar behavior can be found for the SDSC95 and SDSC96 workload modifications. For KTH the results are similar with the exception that we converge to the original workload for decreasing d.

The results for the modified NASA workloads present very similar results for the AWRT and AWWT for the derived and original workloads independently from the used scheduling algorithm. Note, that the NASA workload itself is quite different in comparison to the other workloads as it includes a high percentage of interactive jobs.

In general, the results for this method are still not satisfying. Using a factor of $d = 1$ is not realistic as mentioned in Section 2 because small jobs are missing in relation to the original workload.

3.2 Precise Scaling of Number and Size of Jobs

Consequently, the precise factor is also used for the duplication of jobs. However, as mentioned above, it is not trivial to create fractions of jobs. To this end, a second random variable p_1 was introduced with values between 0 and 100. The variable p_1 is used to decide whether the lower or upper integer bound of the precise scaling factor is considered. For instance, the precise scaling factor for the CTC workload is 2.3814 we used the value of p_1 to decide whether to use the scaling factor of 2 or 3. If p_1 is smaller than 38.14 the factor of 2 will be used, 3 otherwise. The average should result in a scaling factor of around 2.3814. For the other workloads we used the same scaling strategy with the decision values of 24.00 for the KTH workload and with 46.15 for the SDSC95 and SDSC96 workloads.

This enhanced method improves the results significantly. In Table 3 the main results are summarized. Except for the simulations with the SDSC00 workload all other results show a clear improvement in terms of similar utilization for each of the according workloads. The results for the CTC show again that only small values of d lead to convergence of AWRT and AWWT to the original workload. The same qualitative behavior can be observed for the workloads which are derived from the KTH and SDSC00 workloads.

The results for the NASA workload show that AWRT and AWWT do not change between the presented methods. This leads to the assumption that this specific NASA workload does not contain enough workload to produce job delays. The results of the modifications for the SDSC9* derived workloads are already acceptable as the AWRT and AWWT between the original workloads and the modified workloads with a mixture of smaller and bigger jobs ($d = 50$) are already very close. For this two workloads the scaling is acceptable.

In general, it can be summarized that the modification still do not produce matching results for all original workloads. Although we use precise factors for scaling job number and job width, some of the scaled workloads yield better results than the original workload. This is probably caused due to the fact that according to the factor d the scaled workload is distributed over either more but smaller ($d = 99$) or less but bigger jobs ($d = 1$). As mentioned before, the existence of more smaller jobs in a workload usually improves the scheduling result. The results show that a larger machine leads to smaller AWRT and AWWT values. Or contrary, a larger machine can execute relatively more workload than an according number of smaller machine for the same AWRT or AWWT. However, this applies only for the described workload modifications. Here, we generate relatively more smaller jobs in relation to the original workload.

workload	resources	d	Policy	number of jobs	makespan in seconds	util. in %	AWWT in seconds	AWRT in seconds	Squashed Area
CTC	430	ref	EASY	79285	29306750	66	13905	53442	8335013015
	1024	1		82509	29306750	66	13851	53377	19798151305
		50		158681	29306750	75	21567	61117	22259040765
		99		236269	29306750	83	30555	70083	24960709755
	430	ref	FCFS	79285	29306750	66	19460	58996	8335013015
	1024	1		82509	29306750	66	19579	59105	19798151305
		50		158681	29306750	75	28116	67666	22259040765
		99		236269	29306750	83	35724	75253	24960709755

workload	resources	d	Policy	number of jobs	makespan in seconds	util. in %	AWWT in seconds	AWRT in seconds	Squashed Area
KTH	100	ref	EASY	28482	29363625	69	24677	75805	2024854282
KTH	1024	1	EASY	30984	29363625	69	25002	76102	20698771517
KTH	1024	50	EASY	157614	29363625	68	17786	68877	20485558974
KTH	1024	99	EASY	282228	29363625	67	10820	61948	20258322777
KTH	100	ref	FCFS	28482	29381343	69	400649	451777	2024854282
KTH	1024	1	FCFS	30984	29373429	69	386539	437640	20698771517
KTH	1024	50	FCFS	157614	29376374	68	38411	89503	20485558974
KTH	1024	99	FCFS	282228	29363625	67	11645	62773	20258322777
NASA	128	ref	EASY	42049	7945421	47	6	9482	474928903
NASA	1024	1	EASY	44926	7945421	47	6	9482	3799431224
NASA	1024	50	EASY	190022	7945421	47	5	9481	3799431224
NASA	1024	99	EASY	333571	7945421	47	1	9477	3799431224
NASA	128	ref	FCFS	42049	7945421	47	6	9482	474928903
NASA	1024	1	FCFS	44926	7945421	47	6	9482	3799431224
NASA	1024	50	FCFS	190022	7945421	47	5	9481	3799431224
NASA	1024	99	FCFS	333571	7945421	47	1	9477	3799431224
SDSC00	128	ref	EASY	67655	63192267	83	76059	116516	6749918264
SDSC00	1024	1	EASY	72492	63201878	83	74241	114698	53999346112
SDSC00	1024	50	EASY	305879	63189633	83	54728	95185	53999346112
SDSC00	1024	99	EASY	536403	63189633	83	35683	76140	53999346112
SDSC00	128	ref	FCFS	67655	68623991	77	2182091	2222548	6749918264
SDSC00	1024	1	FCFS	72492	68569657	77	2165698	2206155	53999346112
SDSC00	1024	50	FCFS	305879	64177724	82	516788	557245	53999346112
SDSC00	1024	99	FCFS	536403	63189633	83	38787	79244	53999346112
SDSC95	416	ref	EASY	75730	31662080	63	13723	46907	8284847126
SDSC95	1024	1	EASY	77266	31662080	63	14505	47685	20439580820
SDSC95	1024	50	EASY	151384	31662080	70	19454	52652	22595059348
SDSC95	1024	99	EASY	225684	31662080	77	25183	58367	24805524723
SDSC95	416	ref	FCFS	75730	31662080	63	17474	50658	8284847126
SDSC95	1024	1	FCFS	77266	31662080	63	18735	51914	20439580820
SDSC95	1024	50	FCFS	151384	31662080	70	24159	57357	22595059348
SDSC95	1024	99	FCFS	225684	31662080	77	28474	61659	24805524723
SDSC96	416	ref	EASY	37910	31842431	62	9134	48732	8163457982
SDSC96	1024	1	EASY	38678	31842431	62	9503	49070	20140010107
SDSC96	1024	50	EASY	75562	31842431	68	14858	54305	22307362421
SDSC96	1024	99	EASY	112200	31842431	75	22966	62540	24410540372
SDSC96	416	ref	FCFS	37910	31842431	62	10594	50192	8163457982
SDSC96	1024	1	FCFS	38678	31842431	62	11175	50741	20140010107
SDSC96	1024	50	FCFS	75562	31842431	68	18448	57896	22307362421
SDSC96	1024	99	FCFS	112200	31842431	75	26058	65632	24410540372

Table 2: Results for Precise Scaling for the Job Size and Estimated Scaling for Job Number.

workload	resources	d	Policy	number of jobs	makespan in seconds	util. in %	AWWT in seconds	AWRT in seconds	Squashed Area
CTC	430	ref	EASY	79285	29306750	66	13905	53442	8335013015
	1024	1		80407	29306750	66	13695	53250	19679217185
		50		133981	29306750	66	12422	51890	19734862061
		99		187605	29306750	66	10527	50033	19930294802
	430	ref	FCFS	79285	29306750	66	19460	58996	8335013015
	1024	1		80407	29306750	66	18706	58261	19679217185
		50		133981	29306750	66	15256	54724	19734862061
		99		187605	29306750	66	12014	51519	19930294802
KTH	100	ref	EASY	28482	29363625	69	24677	75805	2024854282
	1024	1		31160	29363625	69	24457	75562	20702184590
		50		159096	29363625	69	18868	70002	20725223128
		99		289030	29363625	69	11903	62981	20737513457
	100	ref	FCFS	28482	29381343	69	400649	451777	2024854282
	1024	1		31160	29381343	69	383217	434322	20702184590
		50		159096	29371792	69	41962	93097	20725223128
		99		289030	29363625	69	12935	64013	20737513457
NASA	128	ref	EASY	42049	7945421	47	6	9482	474928903
	1024	1		44870	7945421	47	6	9482	3799431224
		50		188706	7945421	47	2	9478	3799431224
		99		333774	7945421	47	1	9477	3799431224
	128	ref	FCFS	42049	7945421	47	6	9482	474928903
	1024	1		44870	7945421	47	6	9482	3799431224
		50		188706	7945421	47	3	9479	3799431224
		99		333774	7945421	47	1	9477	3799431224
SDSC00	128	ref	EASY	67655	63192267	83	76059	116516	6749918264
	1024	1		77462	63192267	83	75056	115513	53999346112
		50		305802	63189633	83	61472	101929	53999346112
		99		536564	63189633	83	35881	76338	53999346112
	128	ref	FCFS	67655	68623991	77	2182091	2222548	6749918264
	1024	1		77462	68486537	77	2141633	2182090	53999346112
		50		305802	64341025	82	585902	626359	53999346112
		99		536564	63189633	83	38729	79186	53999346112
SDSC95	416	ref	EASY	75730	31662080	63	13723	46907	8284847126
	1024	1		76850	31662080	63	14453	47641	20411681280
		50		131013	31662080	63	13215	46319	20466656625
		99		185126	31662080	62	11635	44739	20446439351
	416	ref	FCFS	75730	31662080	63	17474	50658	8284847126
	1024	1		76850	31662080	63	18511	51698	20411681280
		50		131013	31662080	63	15580	48684	20466656625
		99		185126	31662080	62	12764	45867	20446439351

workload	resources	d	Policy	number of jobs	makespan in seconds	util. in %	AWWT in seconds	AWRT in seconds	Squashed Area
SDSC96	416	ref	EASY	37910	31842431	62	9134	48732	8163457982
	1024	1		38459	31842431	62	9504	49084	20100153862
		50		66059	31842431	62	9214	49087	20106192767
		99		92750	31842431	62	8040	47796	20171317735
	416	ref	FCFS	37910	31842431	62	10594	50192	8163457982
	1024	1		38459	31842431	62	11079	50658	20100153862
		50		65627	31842431	62	10126	49823	20106192767
		99		92750	31842431	62	8604	48360	20171317735

Table 3: Results using Precise Factors for Job Number and Size.

workload	resources	f	Policy	number of jobs	makespan in seconds	utilization in %	AWWT in seconds	AWRT in seconds	Squashed Area
CTC	430	ref	EASY	79285	29306750	66	13905	53442	8335013015
	1024	2.45		136922	29306750	68	14480	54036	20322861231
	430	ref	FCFS	79285	29306750	66	19460	58996	8335013015
	1024	2.45		136922	29306750	68	19503	59058	20322861231
KTH	100	ref	EASY	28482	29363625	69	24677	75805	2024854282
	1024	10.71		165396	29363625	72	24672	75826	21708443586
	100	ref	FCFS	28482	29381343	69	400649	451777	2024854282
	1024	10.71		165396	29379434	72	167185	218339	21708443586
NASA	128	ref	EASY	42049	7945421	47	6	9482	474928903
	1024	8.00		188706	7945421	47	2	9478	3799431224
	128	ref	FCFS	42049	7945421	47	6	9482	474928903
	1024	8.00		188258	7945421	47	4	9480	3799431224
SDSC00	128	ref	EASY	67655	63192267	83	76059	116516	6749918264
	1024	8.21		312219	63204664	86	75787	116408	55369411171
	128	ref	FCFS	67655	68623991	77	2182091	2222548	6749918264
	1024	8.58		323903	69074629	82	2180614	2221139	58020939264
SDSC95	416	ref	EASY	75730	31662080	63	13723	46907	8284847126
	1024	2.48		131884	31662080	63	13840	46985	20534988559
	416	ref	FCFS	75730	31662080	63	17474	50658	8284847126
	1024	2.48		131884	31662080	63	17327	50472	20534988559
SDSC96	416	ref	EASY	37910	31842431	62	9134	48732	8163457982
	1024	2.48		66007	31842431	62	8799	48357	20184805564
	416	ref	FCFS	37910	31842431	62	10594	50192	8163457982
	1024	2.48		66007	31842431	62	10008	49566	20184805564

Table 4: Results for Increased Scaling Factors with $d = 50$.

3.3 Adjusting the Scaling Factor

In order to compensate the above mentioned scheduling advantage of having more small jobs in relation to the original workload, the scaling factor f was modified to increase the overall amount of workload. The aim is to find a scaling factor f that the results in terms of the AWRT and AWWT match to the original workload for $d = 50$. In this way, a combination of bigger as well as more smaller jobs exists. To this end, additional simulations have been performed with small increments of f.

In Table 4 the corresponding results are summarized, more extended results are shown in Table 5 in the appendix. It can be observed that the scheduling behavior is not strict linear corresponding to the incremented scaling factor f. The precise scaling factor for the CTC workload is 2.3814, whereas a slightly higher scaling factor corresponds to a AWRT and AWWT close to the original workload results. The actual values slightly differ e.g. for the EASY ($f = 2.43$) and the FCFS strategy ($f = 2.45$). Note, that the makespan stays constant for different scaling factors. Obviously the makespan is dominated by a later job and is therefore independent of the increasing amount of computational tasks (squashed area, utilization and the number of jobs). This underlines that the makespan is predominantly an off-line scheduling criterion [10]. In an on-line scenario new jobs are submitted to the system where the last submitted jobs influence the makespan without regard to the overall scheduling performance of the whole workload. An analogous procedure can be applied to the KTH, SDSC95 and SDSC96 workloads. The achieved results are very similar.

The increment of the scaling factor f for the NASA workloads leads to different effects. A marginal increase causes a significant change of the scheduling behavior. The values of the AWRT and AWWT are drastically increasing. However, the makespan, the utilization and the workload stay almost constant. This indicates that the original NASA workload has almost no wait time while a new job is started when the previous job is finished.

The approximation of an appropriate scaling factor for the SDSC00 workload differs from the previous described process as the results for the EASY and FCFS strategies differ much. Here the AWRT and the AWWT of the FCFS are more than a magnitude higher than by using EASY-Backfilling. Obviously, the SDSC00 workload contains highly parallel jobs as this causes FCFS to suffer in comparison to EASY backfilling. In our opinion, it is more reasonable to use the results of the EASY strategy for the workload scaling, because the EASY strategy is more representative for many current systems and for the observed workloads. However, as discussed above, if the presented scaling methods are applied to other traces, it is necessary to use the original scheduling method that caused the workload trace.

4 Conclusion

In this paper we proposed a procedure for scaling different workloads to a uniform supercomputer. To this end, the different development steps have been presented as each motivated the corresponding next step. We used combinations of duplicating jobs and/or modifying the requested processor numbers. The results showed again how sensitive workloads react to modifications. Therefore, several steps were necessary to ensure that the scaled workload showed similar scheduling behavior. Resulting schedule attributes as e.g. average weighted response or wait time have been used as quality criteria. The significant differences between the intermediate results for modified workloads indicate the general difficulties to generate realistic workload models. The presented method is motivated as the development of more complex scheduling strategies requires workloads with a careful reproduction of real workloads. Only workload traces include all such explicit and implicit dependencies. As simulations are commonly used for evaluating scheduling strategies, there is demand for a sufficient database of workload traces. However, there is only a limited number of traces available which originate from different systems. The presented method can be used to scale such workload traces to a uniform resource configuration for further evaluations.

Note, we do not propose that our method actually extrapolates an actual user behavior for a specific larger machine. Moreover, we scale the real workload traces to fit on a larger machine while maintaining original workload properties. To this end, our method includes a combination of generating additional job copies and extending the job width. In this way, we ensure that some jobs utilize the same relative number of processors as in the original traces, while original jobs still occur in the workload. For instance, an existing preference of power of 2 jobs in the original workload is still included in the scaled workload. Similarly, other preferences or certain job patterns maintain intact even if they are not explicitly known.

The presented model can be extended to scale other job parameters in the same fashion. Preliminary work has been done to include memory requirements or requested processor ranges. This list can be extended by applying additional rules and policies for the scaling operation.

References

1. Allen B. Downey. A parallel workload model and its implications for processor allocation. In *6th Intl. Symp. High Performance Distributed Comput.*, Aug 1997.
2. Allen B. Downey. Using queue time predictions for processor allocation. In Dror G. Feitelson and Larry Rudolph, editors, *Job Scheduling Strategies for Parallel Processing*, pages 35–57. Springer Verlag, 1997. Lect. Notes Comput. Sci. vol. 1291.
3. Allen B. Downey and Dror G. Feitelson. The elusive goal of workload characterization. *Perf. Eval. Rev.*, 26(4):14–29, Mar 1999.
4. Carsten Ernemann, Volker Hamscher, and Ramin Yahyapour. Economic scheduling in grid computing. In Dror G. Feitelson, Larry Rudolph, and Uwe Schwiegelshohn, editors, *Job Scheduling Strategies for Parallel Processing*, pages 128–152. Springer Verlag, 2002. Lect. Notes Comput. Sci. vol. 2537.
5. Dror G. Feitelson. Packing schemes for gang scheduling. In Dror G. Feitelson and Larry Rudolph, editors, *Job Scheduling Strategies for Parallel Processing*, pages 89–110. Springer-Verlag, 1996. Lect. Notes Comput. Sci. vol. 1162.

6. Dror G. Feitelson. Memory usage in the LANL CM-5 workload. In Dror G. Feitelson and Larry Rudolph, editors, *Job Scheduling Strategies for Parallel Processing*, pages 78–94. Springer Verlag, 1997. Lect. Notes Comput. Sci. vol. 1291.
7. Dror G. Feitelson. Workload modeling for performance evaluation. In M. C. Calzarossa and S. Tucci, editors, *Performance Evaluation of Complex Systems: Techniques and Tools*, pages 114–141. Springer Verlag, 2002. Lect. Notes Comput. Sci. vol. 2459.
8. Dror G. Feitelson and Morris A. Jette. Improved utilization and responsiveness with gang scheduling. In Dror G. Feitelson and Larry Rudolph, editors, *Job Scheduling Strategies for Parallel Processing*, pages 238–261. Springer Verlag, 1997. Lect. Notes Comput. Sci. vol. 1291.
9. Dror G. Feitelson and Bill Nitzberg. Job characteristics of a production parallel scientific workload on the NASA Ames iPSC/860. In Dror G. Feitelson and Larry Rudolph, editors, *Job Scheduling Strategies for Parallel Processing*, pages 337–360. Springer-Verlag, 1995. Lect. Notes Comput. Sci. vol. 949.
10. Dror G. Feitelson and Larry Rudolph. Metrics and benchmarking for parallel job scheduling. In Dror G. Feitelson and Larry Rudolph, editors, *Job Scheduling Strategies for Parallel Processing*, pages 1–24. Springer-Verlag, 1998. Lect. Notes Comput. Sci. vol. 1459.
11. Steven Hotovy. Workload evolution on the Cornell Theory Center IBM SP2. In Dror G. Feitelson and Larry Rudolph, editors, *Job Scheduling Strategies for Parallel Processing*, pages 27–40. Springer-Verlag, 1996. Lect. Notes Comput. Sci. vol. 1162.
12. Joefon Jann, Pratap Pattnaik, Hubertus Franke, Fang Wang, Joseph Skovira, and Joseph Riodan. Modeling of workload in MPPs. In Dror G. Feitelson and Larry Rudolph, editors, *Job Scheduling Strategies for Parallel Processing*, pages 95–116. Springer Verlag, 1997. Lect. Notes Comput. Sci. vol. 1291.
13. Jochen Krallmann, Uwe Schwiegelshohn, and Ramin Yahyapour. On the design and evaluation of job scheduling algorithms. In Dror G. Feitelson and Larry Rudolph, editors, *Job Scheduling Strategies for Parallel Processing*, pages 17–42. Springer Verlag, 1999. Lect. Notes Comput. Sci. vol. 1659.
14. Barry G. Lawson and Evgenia Smirni. Multiple-queue backfilling scheduling with priorities and reservations for parallel systems. In Dror G. Feitelson, Larry Rudolph, and Uwe Schwiegelshohn, editors, *Job Scheduling Strategies for Parallel Processing*, pages 72–87. Springer Verlag, 2002. Lect. Notes Comput. Sci. vol. 2537.
15. V. Lo, J. Mache, and K. Windisch. A comparative study of real workload traces and synthetic workload models for parallel job scheduling. In Dror G. Feitelson and Larry Rudolph, editors, *Job Scheduling Strategies for Parallel Processing*, pages 25–46. Springer-Verlag, 1998. Lect. Notes Comput. Sci. vol. 1459.
16. Uri Lublin and Dror G. Feitelson. The workload on parallel supercomputers: Modeling the characteristics of rigid jobs. *J. Parallel & Distributed Comput.*, (to appear).
17. Ahuva W. Mu'alem and Dror G. Feitelson. Utilization, predictability, workloads, and user runtime estimates in scheduling the IBM SP2 with backfilling. *IEEE Trans. Parallel & Distributed Syst.*, 12(6):529–543, Jun 2001.
18. Uwe Schwiegelshohn and Ramin Yahyapour. Improving first-come-first-serve job scheduling by gang scheduling. In Dror G. Feitelson and Larry Rudolph, editors, *Job Scheduling Strategies for Parallel Processing*, pages 180–198. Springer Verlag, 1998. Lect. Notes Comput. Sci. vol. 1459.
19. Parallel Workloads Archive. http://www.cs.huji.ac.il/labs/parallel/workload/, April 2003.

20. K. Windisch, V. Lo, R. Moore, D. Feitelson, and B. Nitzberg. A comparison of workload traces from two production parallel machines. In *6th Symp. Frontiers Massively Parallel Comput.*, pages 319–326, Oct 1996.

A Appendix: Complete Results

workload	resources	f	Policy	number of jobs	makespan in seconds	util. in %	AWWT in seconds	AWRT in seconds	Squashed Area
CTC	430	ref	EASY	79285	29306750	66	13905	53442	8335013015
	1024	2.41		135157	29306750	67	13242	52897	19890060461
		2.42		135358	29306750	67	13475	52979	20013239358
		2.43		135754	29306750	67	14267	53771	20130844161
		2.45		136922	29306750	68	14480	54036	20322861231
		2.46		136825	29306750	68	13751	53267	20455740107
		2.47		137664	29306750	69	15058	54540	20563974522
		2.48		137904	29306750	69	15071	54611	20486963613
	430	ref	FCFS	79285	29306750	66	19460	58996	8335013015
	1024	2.43		135754	29306750	67	17768	57272	20130844161
		2.44		136524	29306750	67	18818	58326	20291066216
		2.45		136922	29306750	68	19503	59058	20322861231
		2.46		136825	29306750	68	18233	57749	20455740107
		2.47		137664	29306750	69	19333	58815	20563974522
		2.48		137904	29306750	69	19058	58598	20486963613
		2.49		138547	29306750	69	19774	59291	20675400432
KTH	100	ref	EASY	28482	29363625	69	24677	75805	2024854282
	1024	10.68		166184	29363625	72	24756	75880	21649282727
		10.69		165766	29363625	72	24274	75233	21668432748
		10.70		166323	29363625	72	24344	75549	21665961992
		10.71		165396	29363625	72	24672	75826	21708443586
		10.72		166443	29363625	72	24648	75775	21663836681
		10.75		167581	29363625	72	24190	75273	21763427500
		10.78		170046	29363625	73	24417	75546	21829946042
		10.80		168153	29363625	73	25217	76284	21871159818
		10.83		168770	29363625	73	25510	76587	21904565195
	100	ref	FCFS	28482	29381343	69	400649	451777	2024854282
	1024	10.71		165396	29379434	72	167185	218339	21708443586
		10.72		166443	29380430	72	104541	155669	21663836681
		10.80		168153	29374047	73	291278	342345	21871159818
		10.85		167431	29366917	73	295568	346661	21968343948
		10.88		167681	29381624	73	404008	455149	22016195800
		10.89		167991	29366517	73	424255	475405	22051851208
		10.90		169405	29378230	73	281495	332646	22080508136
		10.92		168894	29371367	74	415358	466515	22127579593
		10.96		169370	29381584	74	539856	590999	22204787743
		10.99		170417	29380278	74	491738	542886	22263296356

workload	resources	f	Policy	number of jobs	makespan in seconds	util. in %	AWWT in seconds	AWRT in seconds	Squashed Area
NASA	128	ref	EASY	42049	7945421	47	6	9482	474928903
	1024	8.00		188706	7945421	47	2	9478	3799431224
		8.01		188659	7945421	47	436	9910	3805309069
		8.04		189104	7945421	47	370	9850	3813901379
		8.05		190463	7945421	47	466	9952	3815152286
		8.06		190221	7945421	47	527	10001	3825085688
		8.07		190897	7945421	47	380	9847	3829707646
		8.08		191454	7945421	47	483	9967	3829000061
		8.09		190514	7945507	47	736	10220	3838797287
		8.10		190580	7945421	47	243	9730	3835645184
	128	ref	FCFS	42049	7945421	47	6	9482	474928903
	1024	8.00		188258	7945421	47	4	9480	3799431224
		8.01		188659	7945421	47	562	10036	3805309069
		8.02		189563	7945421	47	629	10126	3806198375
		8.03		189864	7945421	47	427	9901	3810853391
		8.04		189104	7945421	47	534	10013	3813901379
		8.05		190463	7945421	47	562	10048	3815152286
		8.06		190221	7945421	47	721	10194	3825085688
		8.07		190897	7945421	47	531	9998	3829707646
		8.08		191454	7945421	47	587	10070	3829000061
		8.09		190514	7945507	47	605	10088	3838797287
SDSC00	128	ref	EASY	67655	63192267	83	76059	116516	6749918264
	1024	8.12		308872	63209190	85	70622	111043	54813430352
		8.14		309778	63189633	85	71757	112264	54908840905
		8.15		310917	63195547	85	78663	119080	55003341172
		8.16		310209	63189633	85	76235	116714	55030054463
		8.18		310513	63189633	85	74827	115312	55206637895
		8.19		310286	63247375	85	77472	118119	55258239565
		8.20		311976	63194139	86	78585	119254	55368328613
		8.21		312219	63204664	86	75787	116408	55369411171
		8.22		313024	63200276	86	75811	116267	55499902234
	128	ref	FCFS	67655	68623991	77	2182091	2222548	6749918264
	1024	8.55		321966	68877042	82	2133228	2173666	57703096198
		8.56		323298	69093787	82	2154991	2195442	57785593002
		8.58		323903	69074629	82	2180614	2221139	58020939264
		8.59		323908	69499787	82	2346320	2386846	57999342465
		8.60		325858	69428033	82	2338591	2379182	58011833809
		8.61		325467	69146937	82	2248848	2289373	58074546998
		8.63		325458	69258234	82	2219200	2259628	58211844138

workload	resources	f	Policy	number of jobs	makespan in seconds	util. in %	AWWT in seconds	AWRT in seconds	Squashed Area
SDSC95	416	ref	EASY	75730	31662080	63	13723	46907	8284847126
	1024	2.46		130380	31662080	63	13287	46492	20351822499
		2.47		131399	31662080	63	13144	46288	20464087105
		2.48		131884	31662080	63	13840	46985	20534988559
		2.49		131730	31662080	64	13957	47245	20722722130
		2.50		132536	31662080	64	14409	47682	20734539617
		2.52		133289	31662080	64	14432	47628	20794582470
	416	ref	FCFS	75730	31662080	63	17474	50658	8284847126
		2.48		131884	31662080	63	17327	50472	20534988559
		2.49		131730	31662080	64	17053	50341	20722722130
		2.50		132536	31662080	64	17624	50896	20734539617
		2.52		133289	31662080	64	17676	50872	20794582470
		2.53		133924	31662080	65	17639	50820	20955732920
SDSC96	416	ref	EASY	37910	31842431	62	9134	48732	8163457982
	1024	2.46		65498	31842431	62	9055	48736	20026074751
		2.48		66007	31842431	62	8799	48357	20184805564
		2.50		66457	31842431	63	9386	49134	20353508244
		2.51		66497	31842431	63	9874	49315	20502723327
		2.52		66653	31842431	63	9419	48715	20629070916
	416	ref	FCFS	37910	31842431	62	10594	50192	8163457982
	1024	2.47		65842	31842431	62	9674	49361	20120648801
		2.48		66007	31842431	62	10008	49566	20184805564
		2.49		66274	31842431	63	11312	51211	20374472890
		2.50		66457	31842431	63	11321	51069	20353508244
		2.52		66653	31842431	63	11089	50386	20629070916

Table 5: All Results for Increased Scaling Factors with $d = 50$.