

# Parallel Workload Models

This page points to detailed workload models which are based on workload logs collected from large scale parallel systems in production use. As the models do not necessarily include the same features, a short description of each is also provided.

Some of the models include source code of programs to generate workloads according to this model. An effort is made to create the workloads according to the [standard workload format](#).

The following directory attempts to compare the scope of the various models:

<i>model</i>	<i>jobs</i>	<i>work</i>	<i>parallelism</i>	<i>runtime</i>	<i>speedup</i>	<i>arrivals</i>	<i>user runtime estimates</i>
<a href="#">Calzarossa85</a>	Unix	no	no	no	no	yes	no
<a href="#">Leland86</a>	Unix	yes	no	yes	no	no	no
<a href="#">Sevcik94</a>	moldable	no	no	yes	yes	no	no
<a href="#">Feitelson96</a>	rigid	no	yes	yes	no	partial	no
<a href="#">Downey97</a>	moldable	yes	yes	yes	yes	partial	no
<a href="#">Jann97</a>	rigid	no	partial	yes	no	yes	no
<a href="#">Feitelson98</a>	varied	yes	partial	partial	implied	no	no
<a href="#">Lublin99</a>	rigid	no	yes	yes	no	yes	no
<a href="#">Cirne01</a>	moldable	yes	yes	yes	yes	yes	no
<a href="#">Tsafrir05</a>	no	no	no	no	no	no	yes

Rigid jobs are jobs that specify the number of processors they need, and run for a certain time using this number of processors. Moldable jobs specify the amount of total computational work they need to perform, and this can be done by different numbers of processors. The runtime on a specific number of processors depends on the speedup function.

Please send comments and additional information to [feit@cs.huji.ac.il](mailto:feit@cs.huji.ac.il).

## Calzarossa and Serazzi, 1985

This is actually not a model of a parallel workload, but rather a model of the arrival process of interactive jobs in a multiuser environment. It gives the arrival rate as a function of the time of day. It is included because such cyclic arrival patterns do not appear in other models.

### [Detailed description](#)

There is no available code for this model. If you would like to contribute such code, please contact us.

If you use this model in your work, please acknowledge it by citing the following reference:

Maria Calzarossa and Giuseppe Serazzi, "A Characterization of the Variation in Time of Workload Arrival Patterns". *IEEE Trans. Comput.* **C-34(2)**, pp. 156-162, Feb 1985.

This model was used in the following papers: [\[feitelson98b\]](#) [\[gehring99\]](#)

## Leland and Ott, 1986

This is actually not a model of a parallel workload, but rather a model of the runtimes of processes in an (interactive) Unix environment. It is included because it may be relevant for interactive parallel workloads as well.

### [Detailed description](#)

There is no available code for this model. If you would like to contribute such code, please contact us.

If you use this model in your work, please acknowledge it by citing the following reference:  
W. E. Leland and T. J. Ott, "Load-Balancing Heuristics and Process Behavior". *SIGMETRICS Conf. Measurement & Modeling of Comput. Syst.*, pp. 54-69, 1986.

This model has been used and re-affirmed in [[harcholb96](#)] and also used in [[feitelson98b](#)] [[gehring99](#)]

---

## Sevcik, 1994

This model attempts to capture the speedup characteristics of parallel applications, including phenomena such as imbalance, inherent serial work, and parallel overhead. Such a model is useful for evaluating systems where the degree of parallelism is changed dynamically.

### [Detailed description](#)

There is no available code for this model. If you would like to contribute such code, please contact us.

If you use this model in your work, please acknowledge it by citing the following reference:  
K. C. Sevcik, "Application Scheduling and Processor Allocation in Multiprogrammed Parallel Processing Systems". *Performance Evaluation* **19(2-3)**, pp. 107-140, Mar 1994.

This model was used in the following papers: [[parsons95](#)]

---

## Feitelson, 1996

This model characterizes rigid jobs based on observations from 6 logs. It includes the distribution of job sizes in terms of number of processors, the correlation of runtime with parallelism, and repeated runs of the same job.

### [Detailed description](#)

[Download code \(C program\)](#) (10 KB)

If you use this model in your work, please acknowledge it by citing the following reference:  
D. G. Feitelson, "Packing schemes for gang scheduling". In *Job Scheduling Strategies for Parallel Processing*, D. G. Feitelson and L. Rudolph (Eds.), Springer-Verlag, 1996, Lect. Notes Comput. Sci. vol. 1162, pp. 89-110.

This model (or variations of it) were used in the following papers: [[feitelson97a](#)] [[feitelson98](#)] [[feitelson98b](#)] [[lo98](#)] [[ghare99](#)] [[talby99b](#)] [[aida00](#)] [[mualem01](#)] [[feitelson01](#)] [[feitelson03a](#)] [[liux12](#)] [[shih13](#)]

---

## Downey, 1997

This model is based on observations from the SDSC Paragon logs and the CTC SP2 log. Its two main innovations are in the modeling of job runtimes in a way that allows the remaining runtime to be estimated

conditioned on how long the job has already run, and modeling moldable jobs where the number of processors used is not set by the model but can be chosen by the scheduler.

### [Detailed description](#)

[Download code for workload generation \(C program\)](#) (6 KB)

[Download code for complete simulation \(C program\)](#) (19 KB)

If you use this model in your work, please acknowledge it by citing the following reference:

Allen B. Downey, ``[A Parallel Workload Model and Its Implications for Processor Allocation](#)". 6th Intl. Symp. High Performance Distributed Comput., Aug 1997.

This model was used in the following papers: [[downey97c](#)] [[downey97a](#)] [[lo98](#)] [[gehring99](#)] [[talby99b](#)] [[cirne00](#)] [[zhou00](#)] [[zhou01](#)] [[feitelson01](#)] [[feitelson03a](#)] [[sabin06](#)] [[huang13a](#)] [[huang13b](#)] [[huang13c](#)]

---

## **Jann et al, 1997**

This is a detailed model of part of the CTC SP2 log. It handles rigid jobs, and provides information about the distributions of runtimes and interarrival times.

New model parameters were later provided for the workload on the ASCI Blue machine (while parameters are there, they seem to be unusable).

### [Detailed description](#)

[Download original sample code \(C program\)](#) (10 KB)

[Download extended code for complete model, with both parameter sets \(C program\)](#) (17 KB)

If you use this model in your work, please acknowledge it by citing the following reference:

Joefon Jann, Pratap Pattnaik, Hubertus Franke, Fang Wang, Joseph Skovira, and Joseph Riordan, ``[Modeling of Workload in MPPs](#)". In *Job Scheduling Strategies for Parallel Processing*, D. G. Feitelson and L. Rudolph (Eds.), Springer-Verlag, 1997, Lect. Notes Comput. Sci. vol. 1291, pp. 95-116.

The parameters for the ASCI Blue workload were given in:

H. Franke, J. Jann, J. E. Moreira, P. Pattnaik, and M. A. Jette, ``[An Evaluation of Parallel Job Scheduling for ASCI Blue-Pacific](#)". In *Supercomputing '99*, Nov 1999.

Regrettably, these parameters seem to be erroneous.

This model (in either version) was used in the following papers: [[talby99b](#)] [[dasilva00](#)] [[mualet01](#)] [[zhang01](#)] [[feitelson01](#)] [[zhang03b](#)] [[feitelson03a](#)] [[feitelson05b](#)] [[liux12](#)] [[shih13](#)]

---

## **Feitelson and Rudolph, 1998**

This is actually more of a framework to create models of the internal structure of parallel application, in order to be able to investigate the connections between application behavior and scheduling.

### [Detailed description](#)

There is no available code (or even parameter values!) for this model. If you would like to contribute such code, please contact us.

If you use this model in your work, please acknowledge it by citing the following reference:

Dror G. Feitelson and Larry Rudolph, ``[Metrics and Benchmarking for Parallel Job Scheduling](#)". In *Job*

*Scheduling Strategies for Parallel Processing*, D. G. Feitelson and L. Rudolph (Eds.), Springer-Verlag, 1998, Lect. Notes Comput. Sci. vol. 1459, pp. 1-24.

This model was used in the following papers: [[gehring99](#)]

---

## Lublin, 1999/2003

This is a very detailed model for rigid jobs, that includes an arrival pattern with a daily cycle, runtimes that are correlated with the number of nodes, and a distinction between interactive and batch jobs.

A detailed description is provided at the head of the program implementing this model.

[Download code \(C program\)](#) (38 KB)

If you use this model in your work, please acknowledge it by citing the following reference:

Uri Lublin and Dror G. Feitelson, *The Workload on Parallel Supercomputers: Modeling the Characteristics of Rigid Jobs*. *J. Parallel & Distributed Comput.* **63**(11), pp. 1105-1122, Nov 2003.

This model was used in the following papers: [[talby99b](#)] [[batat00](#)] [[feitelson01](#)] [[wiseman03](#)] [[frachtenberg03b](#)] [[feitelson03a](#)] [[barsanti06](#)] [[goh08](#)] [[zeng09](#)] [[sodan09](#)] [[sodan10](#)] [[abbes10](#)] [[minh11](#)] [[sodan11](#)] [[toosi11](#)] [[utreria12](#)] [[neves12](#)] [[shih13](#)]

---

## Cirne and Berman, 2001

This is a comprehensive model for generating moldable jobs. It is composed of two parts:

- A model for generating a stream of rigid jobs
- A model for turning the rigid jobs into moldable ones, by generating a set of alternative <partition size, runtime> options. This is partly based on Downey's speedup model, for which parameter distributions are proposed.

[Download code for complete model \(compressed tar file of C++ source\)](#) (37 KB)

[Download code for moldability model \(compressed tar file of C++ source\)](#) (39 KB)

If you use this model in your work, please acknowledge it by citing the following references:

Walfredo Cirne and Francine Berman, ``[A Comprehensive Model of the Supercomputer Workload](#)". 4th *Ann. Workshop Workload Characterization*, Dec 2001.

and/or

Walfredo Cirne and Francine Berman, ``[A Model for Moldable Supercomputer Jobs](#)". 15th *Intl. Parallel & Distributed Processing Symp.*, Apr 2001.

This model was used in the following papers: [[cao04](#)]

---

## Tsafrir, 2005

This is a very detailed model that generates realistic user runtime estimates, upon which backfill schedulers rely. The model targets the modal nature of user estimates (very few popular values; most popular is the maximal estimate). It is composed of two parts:

1. generating a realistic distribution of user runtime estimates, and
2. embedding this distribution within a real workload log or the output of a workload model.

[Detailed description and "how to"](#)

[Download model's code](#): Compressed tar file (614K) of C++ source, documentation, and examples or see [detailed listing of individual files](#).

If you use this model in your work, please acknowledge it by citing the following references:

1. Dan Tsafir, Yoav Etsion, and Dror G. Feitelson, ``[Modeling User Runtime Estimates](#)". 11th *Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP)*, pp. 1-35, Jun 2005. Lecture Notes in Computer Science Vol.3834 (528K).
2. Dan Tsafir, ``[A Model/Utility to Generate User Runtime Estimates and Append Them to a Standard Workload File](#)". URL [http://www.cs.huji.ac.il/labs/parallel/workload/m\\_tsafir05](http://www.cs.huji.ac.il/labs/parallel/workload/m_tsafir05)

---

[Back to the Parallel Workloads Archive home page](#)

DGF / Sep 13, 2012