Mathematical Problems of Computer Science 46, 55--58, 2016.

About Some Queueing Models for Computational Grid Systems

Vladimir G. Sahakyan, Yuri H. Shoukourian and Hrachya V. Astsatryan

Institute for Informatics and Automation Problems of NAS RA
e-mail: vladimir.sahakyan@sci.am, shouk@sci.am, hrach@sci.am

Abstract

In this paper parametric models of the queues are proposed in grid systems. The models take into account the restrictions on the waiting and interval of execution time of the tasks.

Keywords: Grid Computing, Queueing Systems, Resource allocation systems.

The optimal usage of CPU times in multi-core computational Grid infrastructures depends on several factors, such as the job scheduling, the possibility of dynamic allocation of computational resources, the possibility of job migration to the different phases of implementation in optimal requested environment, performing a stop of the job with the possibility of continuing, etc. The use of grid infrastructures to handle the information stream requires a flexible approach to the allocation of resources as well as for the timely performance of jobs.

The job receiving process in the queue for further executing in a grid infrastructure plays an important role in the organization of the whole process. At this stage the incoming job service script is generated based on a maintenance data bursts. This requires synchronization of distributed processes to manage the resources of a grid system. A set of GRAM (Grid Resource Allocation and Management) services allows to manage the resources and jobs, job queues handling, ensuring their execution on the required computational resources.

In addition to GRAM services, the schedulers play special roles. The adoption of the job queue for a service applies to the scheduler responsible for ensuring its timely implementation. Nowadays the usage of modern schedulers, such as Maui, Condor, PBS (Portable Batch System) has opportunities to perform the planning tasks on specified priorities [1,2,3]. However, in complex cases (for instance, to run the job at a specified time and in specified resources, or to manage multi-stream queues) we need to implement a flexible approach for scheduling.

The optimal run of the programs in multi-core environments involve selection of service discipline, delivering at least a "loss function", which characterizes the quality of the functioning of the process. As such a function can be considered some functional depending on the average waiting time, utilization of time, the number of denial of service and others. Many authors have shown that the discipline of the relative priority is optimal for linear functional without service interruption. If prevent the implementation of programs interrupted, sometimes absolute priority disciplines are better.

However, in systems with many devices maintenance procedure can be disrupted, while retaining the properties of known subjects. For example, in case of a FIFO discipline, if a job is received after the service does not affect the incoming job before him, it may be served. This can be done using Batchfill mechanism (check available periods for low-priority tasks). Often it is required to provide manage the job not later or earlier, or in a specified period of time after entering the system. For a description of these conditions we consider the classification of the job queues on systems diagrams defining the service priorities.

Schemes depending on the order of receipt

- MFIFO (Modified FIFO) – received to the job system, it is queued in order of receipt and can serve, if it does not affect the start time of service to all previously received assignments;
- MLIFO (Modified LIFO) - received to the job system, it is queued in order of receipt and can serve, if it does not affect the start of the service later he received jobs from the queue.
- MRS (Modified Random Service) - received to the job system, it is queued and can cater for random selection depending on specific task parameters.

Schemes based on the level of required resources

-JU (Job Up): received to the job system, it is queued in order of required resources for the service ([service time] * [number of required cores]) and can serve, if it does not affect the start of the maintenance of all available jobs in the queue ahead of its assignments. Previously, all will go on maintenance task requires the most resources.

-JD (Job Down): received to the job system, it is queued in order of increasing required for maintenance resources ([service time] * [number of required cores]) and can serve, if it does not affect the start of the maintenance of all available jobs in line behind its assignments. Previously, all will go on maintenance task requires the least resources.

Schemes based on the number of processors required

-PU (Processor Up): received to the job system, it is queued in order of processors required for service and can serve if it does not affect the start of the maintenance of all available jobs in the queue ahead of its assignments. Previously, all will go on maintenance task requires the most resources.

-PD (P Down): received to the job system, it is queued in order of increasing required maintenance for the processor and can serve, if it does not affect the start of service available queued jobs behind it. Previously, all will go on maintenance task requires the least resources.

Schemes, with a restriction on the waiting time

-WTR (Waiting Time Restriction): received to the job system, it is queued in order acceptable waiting time, i.e., the higher priority admission service has a job with less latency.

-ITR (Interval Time Restriction): received to the job system, it is queued and must start the handling after a specified time, but no later than the specified timeout. The first goes on a mission handling shall start the service before anyone else.

All queuing schemes permitted job handling from the queue, if its handling does not affect the maintenance of higher-priority jobs.

Schemes are mutually exclusive and only one of the circuits can be selected and when queuing. However, restrictions on the waiting time can be in all schemes.

In systems allowing job migrations, these schemes can be described as a service processes without interruption (relative priority) and with interruption (absolute priority).

The job interruption may take place in case of entering higher priority jobs. We assume that the job under the new entry to the service will be continued (not started first). Setting with interrupted service it is returned to the queue with the adjusted design parameters (service time, waiting time, etc.).

For a more detailed formulation let's consider a computational consisting of m ($m \geq 1$) processors. When using systems based on virtual clusters, limiting the number of processors is arbitrary and depends on the performance of the basic configuration. We assume that the number of jobs that can be in the queue is not limited. The grounds of refusal of service can only serve as the impossibility of his service with the user-defined constraints (time, number of processors, etc.).

Let job stream be supplied. Each job can be characterized by parameters (ν, β, ω, γ), where ν - the number of processors required for the job, β - the time required to run, ω - permissible total residence time job in the queue, γ - the time from the admission to the system after it is allowed to start the service.

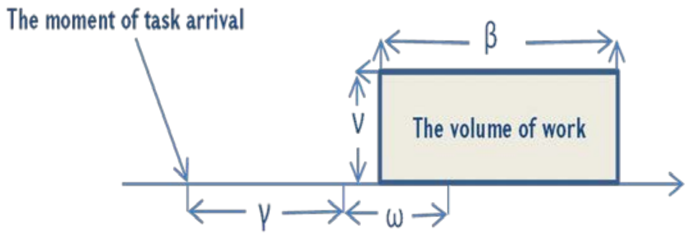

Fig. 1. The example of task an acceptable execution.

If $\gamma = 0$, the value of this parameter can be omitted. In systems with no limit on the waiting time the value of the parameter ω is also lowered.

For jobs with a restriction on the waiting time for admission to the system is checked on the ability to perform a task and then either taken in turn to perform, or is refused by the implementation.

The time required for maintenance is in some sense arbitrary, which means maximum allowable. In reality, it is random and may be less than the predetermined one. Therefore, the order may vary both in case of receiving the jobs and finishing the maintenance of jobs.

Service denial gets jobs, if at the time of admission to the system it finds that it cannot serve the specified parameters. For example, to start services at a specified time. The system is considered at discrete points in time, the job proceeds to the queue or completion of service. We call these moments 0-momentums.

In each 0-momentum it carries out queue recalculation and a new order of the queue in the computational system.

For various schemes a multithreading queuing model is proposed. Between the streams are set priorities, and inside threads can be used as one of the described schemes of service.

For the above mentioned schemes, new algorithms have been developed to schedule jobs with mixed priorities. Note that the algorithms can be used both in homogeneous and in lines with mixed schemas with a restriction on waiting time.

References

[1] I. Foster, *Computational Grid*, Morgan-Kufman, 1999.
[2] I. Foster, C. Kesseman and S. Tuecke, “The Anatomy of the Grid”, *International Journal of HPC Applications*, vol. 15, no. 3, pp. 25-37, 2001.
[3] F. Xhafa, “Batch mode scheduling in grid systems”, *Int.J. Web and Grid Services*, vol.3, no. 1, pp.19-37, 2007.

Submitted 04.07.2016, accepted 20.10.2016.

Հերթերի որոշ մոդելների մասին հաշվողական գրիդ–համակարգերում

Վ.Սահակյան, Յու.Շուքուրյան և Հ. Ասցատրյան

Ամփոփում

Հոդվածում առաջարկվում է հաշվողական գրիդ–համակարգի համար հերթի պարամետրիկ մոդելը: Մոդելը հաշվի է առնում առաջադրանքի հերթում սպասման և կատարման թույլատրելի ժամանակները։

О некоторых моделях очередей в вычислительных грид-системах

В. Саакян, Ю. Шукурян и Г. Асцатрян

Аннотация

В статье предложена параметрическая модель очереди для вычислительной грид-системы. Модель учитывает ограничения на допустимые времена ожиданий и промежутков выполнения заданий.