

# Improved Model of Scheduling Algorithm

Artur P. Vardanyan

Institute for Informatics and Automation Problems of NAS RA  
e-mail: artvardanyan@asnet.am

## Abstract

Cluster computing is becoming increasingly practical for high performance computing research and development. A computer cluster is a set of connected computers that work together so that, they can be viewed as a single system. Clusters offer a scalable means of linking computers together to provide an expansive environment for hosting enterprise applications. As the number of nodes in cluster configurations grows, the cluster administration becomes more challenging. We need to study the challenges of cluster management and to provide a solution. To have an effective cluster management we need to have an effective task scheduling algorithm. With the explosive growth of information, the demand on computing is sharply increasing. Due to a large number of computing tasks, the scheduling algorithm is an important part of cluster computing and has a great influence on the quality of cluster service. In cluster computing, some large tasks may occupy too many resources and some small tasks may wait for a long time based on First-In-First-Out (FIFO) scheduling algorithm. This paper provides an overview of an improved scheduling algorithm that shortens the execution time of tasks and increases the resource utilization.

**Keywords:** Cluster computing, Task scheduling, Scheduling algorithm.

## 1. Introduction

Task Scheduling is to assigned a task to an available resource for its execution. Task scheduling is an important component of cluster computing and has a great impact on the quality of cluster service. The basic problem of task scheduling that needs to be solved is assigning tasks to resources. A scheduler is what carries out the scheduling activity. Schedulers are often implemented so they keep all cluster resources busy (as in load balancing), allow multiple users to share cluster resources effectively, or to achieve a target quality of service. FIFO scheduler is a common scheduler, and it chooses the task to execute according to task arrival time. By default, FIFO scheduler does not consider a tasks priority.

Usually, in task scheduling algorithms, tasks are described by two parameters: the resource utilization parameter and task's execution time. Task's resource utilization parameter is the number of cluster nodes required to perform this task. Task's execution time is the maximum time necessary to complete this task. But in this paper we propose a task scheduling algorithm, in which tasks are described by three parameters: the resource utilization parameter, task's execution time and task's waiting time. What is a tasks waiting time? Tasks waiting time is the maximum time that this task can wait before assigning to run. To describe each

task we used  $(\nu, \beta, \omega)$  triplet, where  $\nu$  is the number of cluster nodes required to perform this task,  $\beta$  is the maximum time necessary to complete this task and  $\omega$  is the maximum time that this task can wait before assigning to run. For the value to the number of the cluster nodes we took  $m$ . The first moment of the time we have a queue, in which there are  $n$  tasks, that is we have a queue with  $n$  three-dimensional vectors.

## 2. Model of Scheduling Algorithm for FIFO

Improved algorithm description:

Input:

$n$  tree-dementional vectors:  $\{(\nu_i, \beta_i, \omega_i); i = 1, 2, \dots, n\}$ ;  
the number of cluster nodes:  $m$ ;

Process:

Step1: declare the variables

the number of assigned tasks:  $k = 0$ ;  
the number of cluster occupied nodes:  $s = 0$   
the number of cluster idle nodes:  $f$ ;

Step 2: while  $s + \nu_{k+1} \leq 0$  then  $k = k + 1, s = s + \nu_k$  and serve  $k^{th}$  task;

Step 3: set  $f = m - s$ ;

Step 4: sort  $\beta_1, \beta_2, \dots, \beta_k$  in non-decreasing order:  $\beta_{i_1}, \beta_{i_2}, \dots, \beta_{i_k}$ ;

Step 5: declare the variables

$\tau_i^{th}$  moment of the time:  $\tau_i, i = 1, 2, \dots, k$ ;  
the number of cluster idle nodes at  $\tau_i^{th}$  moment of the time:  
 $m_i, i = 1, 2, \dots, k$ ;

Step 6: for  $j = 1$  to  $k$  set  $\tau_i = \beta_{i_j}$  and  $m_j = f + \sum_{i=1}^j m_i$ ;

Step 7: go through the queue and find which  $(\nu_j, \beta_j, \omega_j)$  so that  $\nu_j \leq f, \beta_j < \tau_1$   
for  $j = k + 1$  to  $n$   
if  $\nu_j \leq f, \beta_j < \tau_1$  then serve  $j^{th}$  task and break for;

Step 8: check if there are tasks in queue that have the waiting time less than  $\tau_1$   
for  $j = k + 1$  to  $n$

if  $\omega_j < \tau_1$  then delete  $j^{th}$  task from the queue and rearrange the queue;

Step 9: while the queue is not empty at each  $\tau_i, i = 1, \dots, k$  point do these checks  
for  $j = k + i + 1$  to  $n$

if  $\omega_j < \tau_1$  then delete  $j^{th}$  task from the queue and rearrange the queue;

if  $\nu_{k+i} \leq m_i$  then serve  $(k + i)^{th}$  task;

else

for  $j = k + i + 1$  to  $n$

if  $\nu_j \leq m_i$  and  $\tau_i + \beta_j < \tau_{i+1}$  then serve  $j^{th}$  task and break for.

## 3. Conclusion

We presented the improved model of the scheduling algorithm, which decreases the execution time of tasks and increases the resource utilization of the system. In the above mentioned algorithm each task is characterized by parameters  $(\nu, \beta, \omega)$ , where  $\nu$  is the number of cluster nodes required to perform this task,  $\beta$  is the maximum time necessary to complete this task and  $\omega$  is the maximum time that this task can wait before assigning to run. Note that in

the improved algorithm we have used the backfill scheduling model. Backfill is a scheduling optimization, which allows the scheduler to make better use of available resources by running jobs out of order.

## References

- [1] V. G. Sahakyan, Y. H. Shoukourian, H. V. Astsatryan, "About some queueing models for computational grid systems", *Mathematical Problems of Computer Science*, Yerevan, Armenia, 2016.
- [2] M. L. Pinedo, Scheduling: Theory, Algorithms, and Systems, *Springer International Publishing*, 5th ed., 2016.
- [3] P. Brucker, Scheduling Algorithms, *Springer International Publishing*, 5th ed., 2007.

Submitted 10.04.2018, accepted 28.11.2018.

## Պլանավորման ալգորիթմի բարելավված մոդել

Ա. Վարդանյան

### Անփոփում

Հոդվածում ներկայացվում պլանավորման ալգորիթմի մոդել, որը նվազեցնում է առաջադրանքի կատարման ժամանակը և մեծացնում համակարգի ռեսուրսների օգտագործումը: Ալգորիթմում յուրաքանչյուր առաջադրանք բնութագրվում է  $(\nu, \beta, \omega)$  պարամետրերով, որտեղ  $\nu$ -ն այդ խնդիրը սպասարկելու համար պահանջվող կլաստերային հանգույցների քանակն է,  $\beta$ -ն այդ խնդիրը սպասարկելու համար պահանջվող առավելագույն ժամանակն է, իսկ  $\omega$  - ն այն առավելագույն ժամանակն է, որ այդ առաջադրանքը կարող է սպասել մինչև սպասարկման ուղարկվելը:

## Улучшенная модель алгоритма планирования

А. Варданян

### Аннотация

В статье вводится улучшенную модель алгоритма планирования, которая уменьшает время выполнения задач и увеличивает использование ресурсов системы. В вышеупомянутом алгоритме каждая задача характеризуется параметрами  $(\nu, \beta, \omega)$ , где  $\nu$ -количество узлов кластера, необходимых для выполнения этой задачи,  $\beta$ -максимальное время, необходимое для выполнения этой задачи, а  $\omega$  -максимальное время, что эта задача может подождать до назначения для запуска.