Minimisation of the Average Response Time in a Cluster of Servers

Valeriy Naumov

Abstract: *In this paper, we consider task assignment problem in a cluster of servers. We show that optimal static task assignment is tantamount to equalizing an appropriate cost functions associated with the servers. We also propose an improvement of dynamic Shortest Expected Delay (SED) task assignment policy.*

Key words: *Distributed System, Task Assignment, Average Response Time*.

1. INTRODUCTION

Task assignment policy is an important factor affecting the performance of a distributed system because it coordinates the use of processing capacity of servers. High-speed Web clusters [1] and Internet routers [2] implement various task assignment policies. An important element of a task assignment policy is the information it requires to operate. In general, dynamic policies operate under time dependent information, whereas static policies operate under time independent characteristics of the system [3].

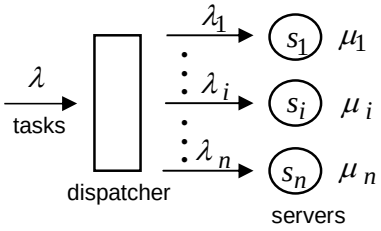

Figure 1. Task assignment in a cluster of servers.

We consider the task assignment problem in a cluster of servers where different servers exhibit different task processing times as shown in Figure 1. Tasks arrive to dispatcher, which is responsible for distributing them among servers according to a task assignment policy taking into account availability of resources at servers. We are interested in policies, which minimize the average response time and distinguish two optimisation problems. Optimization criterion in the Minimum Average Response Time (MART) problem is to minimize the average response time taken over all processed task. In the Minimum of the Maximum Response Times (MMRT) problem, the criterion of optimality is to minimize the maximum of average task response times at the servers, to which tasks are routed.

Static MART problem has been considered in several papers. Tantawi and Towsley [4] studied an arbitrarily connected distributed system. Tasks may arrive at any server and can be executed either locally or be sent to another server. They derived an iterative algorithm that determined the optimal static task assignment policy for a system with general response time and communication delay functions. Kim and Kameda improved this algorithm in [5]. Buzen and Chen [6] derived equations for optimal arrival rates λ_i at servers for a cluster of servers with Poisson task arrival process and generally distributed task processing times. Ni and Hwang [7], and Tang and Chanson [8], found closed form solution for particular case when task-processing times are exponentially distributed.

Minimization of the maximum of average task response times may be unfair since the average response time at the slower servers can be much higher than the average response time at the faster ones. Georgiadis et al. in [13] derived solution of the static MMRT problem for a cluster of servers with general response time functions. Optimal policy attains fairness by equalizing the average response times at active servers, at an increase in average response time taken over all processed tasks.

Dynamic task assignment policies usually outperform static policies [9], but optimal dynamic policy is unknown because of analytical difficulties [10]. Chow and Kohler [11] proposed simple dynamic Minimum Response Time policy. In Minimum Response Time,

also known as Shortest Expected Delay (SED), an arriving task is routed to the server with the least expected response time, i.e. the server i, for which $(s_i+1)/\mu_i$ is minimal, where s_i is the number of tasks at server i including the one in service, and μ_i is its task processing rate.

In this paper, we analyse optimal solutions for static task assignment policies. Based on this analysis we propose new dynamic task assignment policy - Modified SED, which is as simple as SED but results in smaller average response time.

2. OPTIMAL STATIC POLICIES

Consider a cluster of n servers. We denote λ the rate of tasks arriving at the system, λ_i the rate of tasks arriving at server i, θ_i the maximum task arrival rate that server i can sustain without becoming saturated, and $\Theta = \theta_1 + \ldots + \theta_n$ the maximum throughput of the system. The mean processing time needed to execute the task at server i at the absence of other tasks arrived from dispatcher, is denoted by β_i. In what follows we assume that servers are numbered in nondecreasing order of the mean processing times β_i, i.e. $\beta_1 \le \beta_2 \le \ldots \le \beta_n$, and it will be convenient to define an additional quantity, $\beta_{n+1} = \infty$. The ith server will be called faster than the jth server if $\beta_i \le \beta_j$.

2.1. Equalization Problem

Assume that the ith server has a cost function $c_i(\lambda_i)$ associated with it, and the following conditions hold:

1) $c_i(x)$ is strictly increasing and continuous for $x \in (0, \theta_i)$,
2) $\lim_{x \uparrow \theta_i} c_i(x) = \infty$,
3) $\lim_{x \downarrow 0} c_i(x) = \beta_i > 0$.

The load can be distributed among the fastest servers, so that the values of cost functions $c_i(\lambda_i)$ at all these servers are equalized. In this case the arrival rates λ_i form a solution of the following equalization problem:

Equalization problem: For a given λ, $0 < \lambda < \Theta$, find an integer k, $1 \le k \le n$, and task arrival rates $0 < \lambda_i < \theta_i$, $i = 1,2 \ldots k$, so that $\lambda_1 + \lambda_2 + \ldots + \lambda_k = \lambda$ and $c_1(\lambda_1) = c_2(\lambda_2) = \cdots = c_k(\lambda_k) \le \beta_{k+1}$.

Georgiadis et al. present in [13] the solution of the equalization problem. Let $g_i(x), x > \beta_i$ be the inverse of $c_i(x)$, and set $g_i(x) = 0$ for $0 \le x \le \beta_i$. Server i is called activated if the arrival rate λ_i to this server is nonzero. As the task arrival rate λ increases from 0 to Θ, the number of activated servers increases from 1 to n. Server i is activated when λ exceeds a threshold A_i, called the activation rate for the server i. The server activation rates can be computed by

$$A_k = \sum_{i=1}^{k-1} g_i(\beta_k),\ 1 \le k \le n,$$

and have the following properties:

a) $A_1 = 0$,
b) $A_1 \le A_2 \le \ldots \le A_n < A_{n+1} = \Theta$,
c) $A_i = A_j$ when $\beta_i = \beta_j$.

When $A_k < \lambda < A_{k+1}$, then the arrival rates at servers satisfy $c_1(\lambda_1) = c_2(\lambda_2) = \cdots = c_k(\lambda_k) = E(\lambda)$, where $\beta_k < E(\lambda) < \beta_{k+1}$, and $\lambda_i = 0$ for $i > k$. Details of the algorithm for calculation of the arrival rates λ_i for arbitrary λ can be found in [13].

2.2. Minimization of the Average Response Time

The response time function $R_i(x)$ specifies the average response time of a task at server i for a task arrival rate x to that server. It is well known that for server i modelled as M/M/1 queue $R_i(x)$ is given by simple formula $R_i(x) = (\mu_i - x)^{-1}$ [12]. If the service times are not exponentially distributed and/or the server processes additional dedicated arrival stream then the response time function would be more complex function. We assume that the response time functions satisfy the following conditions:

1) $R_i(x)$ is strictly increasing and continuous for $x \in (0, \theta_i)$,
2) $\lim_{x \uparrow \theta_i} R_i(x) = \infty$,
3) $\lim_{x \downarrow 0} R_i(x) = \beta_i > 0$.

Minimum of the Maximum Response Time and Minimum Average Response Time problems for a cluster of servers can be formulated as follows.

Minimum of the Maximum Response Times (MMRT) problem: For a given λ, $0 < \lambda < \Theta$, find task arrival rates at servers $0 \le \lambda_i < \theta_i$, $i = 1,2 \ldots n$, so that $\lambda_1 + \lambda_2 + \ldots + \lambda_n = \lambda$ and the maximum average response time $\max_{\lambda_i > 0}\{R_i(\lambda_i)\}$ is minimized.

Minimum Average Response Time (MART) problem: For a given λ, $0 < \lambda < \Theta$, find task arrival rates at servers $0 \le \lambda_i < \theta_i$, $i = 1,2 \ldots n$, so that the $\lambda_1 + \lambda_2 + \ldots + \lambda_n = \lambda$ and average overall response time $R(\lambda) = \sum_{i=1}^{n} \frac{\lambda_i}{\lambda} R_i(\lambda_i)$ is minimized.

Both problems are tantamount to the equalization problem. Georgiadis et al. show in [13] that the solution of the MMRT problem can be found as a solution of the equalization problem with the cost function given by $c_i(x) = R_i(x)$. Tantawi and Towsley studied in [4] the problem of minimization of the average response time for an arbitrarily connected distributed system assuming that the response time functions $R_i(x)$ are differentiable and convex for $x \in (0, \theta_i)$. It follows from [4] that the solution of the MART problem can be found as a solution of the equalization problem with the cost function given by $c_i(x) = \frac{d(xR_i(x))}{dx}$.

3. MODIFIED SED POLICY

For particular case, when task arrival process is Poisson and task processing times are exponentially distributed, we have $\theta_i = \mu_i$, $\beta_i = 1/\mu_i$. The cost functions $c_i(x)$ and its inverse $g_i(x)$ are given by

$$c_i^1(x) = \frac{1}{\mu_i - x}, \qquad g_i^1(x) = \begin{cases} \mu_i - \frac{1}{x}, & x\mu_i > 1, \\ 0, & 0 < x\mu_i \le 1, \end{cases} \quad \text{for MMRT problem,}$$

$$c_i^2(x) = \frac{\mu_i}{(\mu_i - x)^2}, \qquad g_i^2(x) = \begin{cases} \mu_i - \sqrt{\frac{\mu_i}{x}}, & x\mu_i > 1, \\ 0, & 0 < x\mu_i \le 1, \end{cases} \quad \text{for MART problem.}$$

From results presented in the previous section we may conclude that equalization of functions $c_i^1(x)$, i.e. the average response times, results in minimization of the maximum response time incurred on any active server. While equalization of either functions $c_i^2(x)$ or $\sqrt{c_i^2(x)}$, which are the average response times multiplied by square roots of task

processing rates, results in minimization of the average response time taken over all processed tasks.

In SED policy, an arriving task is sent to the server i, for which $(s_i+1)/\mu_i$ is minimal, where s_i is the number of tasks at server i. In other words, SED policy equalizes response times at active servers, that leads to minimization of the maximum response time incurred on any server. We modify SED so, that in Modified SED (MSED), an arriving task is sent to the server i, for which $(s_i+1)/\sqrt{\mu_i}$ is minimal. MSED policy equalizes response times multiplied by square roots of task processing rates, and thus, according to previous considerations, would minimize the average response time taken over all processed tasks.

We simulate three different systems, which were used in [9], and compare SED and MSED. The first system, System 1, has 10 nodes. The task processing rates of the nodes are $\mu_1 = 6$, $\mu_2 = \mu_3 = \ldots = \mu_{10} = 1$. The second system, System 2, also has 10 nodes. The task processing rates of the nodes form an arithmetic series, that is $\mu_i = 3\left(1 - \frac{i}{11}\right)$, $i = 1,2\ldots10$. Each of these two systems has an aggregate processing rate of 15. The third system, System 3, has 8 nodes, and the task processing rates of the nodes form the geometric series $\mu_i = 2^{10-i}$, $i = 1,2\ldots10$. In all cases, the arrival process is assumed to be Poisson.

Figures 2 and 3 show the average response times taken over all processed tasks under SED (solid lines) and MSED (dash lines) versus task arrival rate λ.

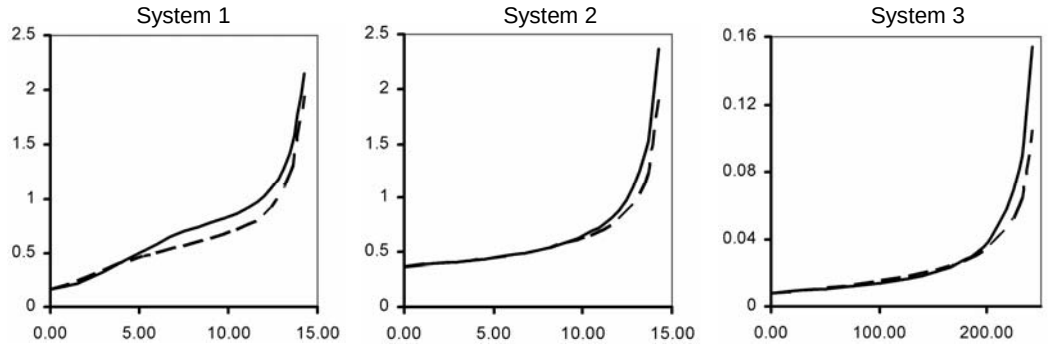

Figure 2. Average response times,
when task processing times are exponentially distributed.

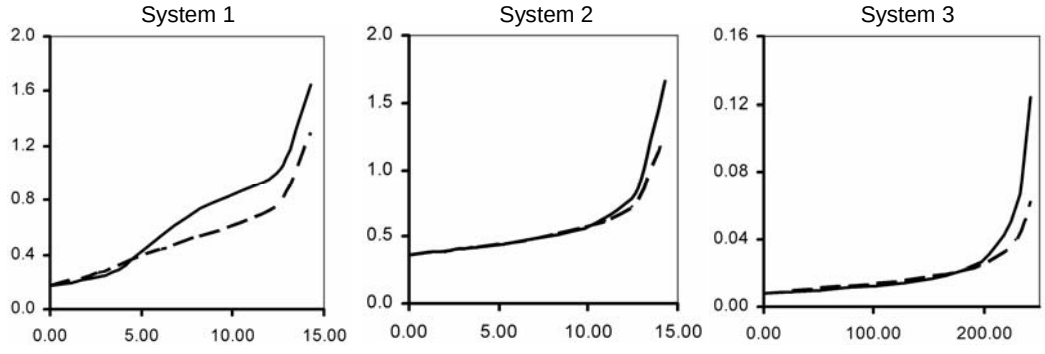

Figure 3. Average response times, when task processing times are constant.

Results shown in Figure 2 demonstrate that MSED policy outperforms SED when task-processing times have exponential distributions. Figure 3 shows that MSED policy also outperforms SED for systems with constant task processing times. In this case, the differences in average response times are even higher then in the case of exponential distributions.

3. CONCLUSION

We show that the static policy, that equalizes task response times at active servers, does not minimize average overall response time. Well-known dynamic Shortest Expected Delay assignment policy tries to minimise the average overall response time by sending new task to a server with minimal expected response time. We propose modification to SED policy. In Modified SED, an arriving task is sent to the server, for which expected response times multiplied by square roots of task processing rates is minimal. Modified SED is as simple as SED but results in smaller the average overall response time at high load.

REFERENCES

[1] Cardellini V., E. Casalicchio. The State of the Art in Locally Distributed Web-Server Systems, *ACM Computing Surveys*, Vol. 34, No. 2, June 2002, pp. 263–311.

[2] Younis O., S. Fahmy. Constraint-Based Assignment in the Internet: Basic Principles and Recent Research, *IEEE Communications Surveys*, Vol. 5, No. 1, 2003, pp. 2-13.

[3] Casavant T.L., J.G. Kuhl. A Taxonomy of Scheduling in General-Purpose Distributed Computing Systems, *IEEE Transactions on Software Engineering*, Vol. 14, No. 2, February 1988, pp. 141-154.

[4] Tantawi A.N., D. Towsley. Optimal Static Load Balancing in Distributed Computer Systems, *Journal of the ACM*, Vol. 32, No. 2, Apr. 1985, pp. 445-465.

[5] Kim C., H. Kameda. An Algorithm for Optimal Static Load Balancing in Distributed Computer Systems, *IEEE Transactions on Computers*, Vol. 41, No. 3, March 1992, pp. 381-384.

[6] Buzen J.P., P.P.-S. Chen. Optimal Load Balancing in Memory Hierarchies, *Proc. IFIP 1974* (J.L. Rosenfeld ed.), North-Holland, Amsterdam, 1974, pp. 271–275.

[7] Ni L.M., K. Hwang. Optimal Load Balancing in a Multiple Processor System with Many Task Classes. *IEEE Transactions on Software Engineering*, Vol. 11, No. 5, 1985, pp. 491–496.

[8] Tang X., S.T. Chanson. Optimizing Static Task Scheduling in a Network of Heterogeneous Computers, *Proc. of ICPP 2000*, Aug. 2000, pp. 373--382.

[9] Banawan S.A., N.M. Zeidat. Comparative Study of Load Sharing in Heterogeneous Multicomputer Systems, *Proc. 25th Annual Simulation Symposium*, Orlando, Florida, USA, Apr. 6-9, 1992, pp. 23-31.

[10] Boxma O., G. Koole, Z. Liu. Queueing-Theoretic Solution Methods for Models of Parallel and Distributed Systems, *Performance Evaluation of Parallel and Distributed Systems - Solution Methods,* (O.J. Boxma and G.M. Koole eds.), CWI, Amsterdam, 1994, pp. 1-24.

[11] Chow Y.C., W.H. Kohler. Models of dynamic load balancing in a heterogeneous multiple processor system. *IEEE Transactions on Computers*, Vol. C-28, No. 5, May 1979, pp. 354-361.

[12] Kleinrock L., *Queueing Systems, Vol. 1, Theory*, Wiley Interscience, 1975.

[13] Georgiadis L., C. Nikolaou, A. Thomasian. A Fair Workload Allocation Policy for Heterogeneous Systems, J. Parallel Distrib. Comput., Vol. 64, No. 4, 2004, pp. 507-519.

ABOUT THE AUTHOR

Prof. Valeriy Naumov, Laboratory of Communication Engineering, Lappeenranta University of Technology, Finland, Phone: +358 5 6212873, E-mail: valeriy.naumov@lut.fi.