Rational Server Selection for Mobile Agents: Problem Solution and Performance Analysis

Carsten Pils, Jan Kritzner, and Stefan Diepolder

Informatik 4 (Communications Systems)
RWTH Aachen, 52056 Aachen, Germany
{pils, kritzner, diepolder}@informatik.rwth-aachen.de

Abstract. Since agents have the ability to migrate to outperforming resources they can potentially balance the load of heterogeneous systems. However, to balance resources efficiently agents must take the load into account. Thus, to support the agent migration strategy the application of server selection systems has been proposed recently. Server selection systems keep track of the load of network and host resources and hence predict the performance of different migration strategies. Yet, server selection comes at a cost and therefore agents must take care when applying it. This paper presents a decision strategy for the agent's decision problem. The performance of the approach is analysed with the help of a simple queuing model.

1 Introduction

Basically, the advantage of mobile agent technology is that it allows application designers to decide where an agent is processed. In that sense, developers can optimise the performance of an application by carefully selecting processing resources. Yet, approaches proposing an automatic selection of destination systems [1] [2] [3] have not gained much attention. A similar problem is server selection in the Internet. The deployment of mirror sites has motivated users to select a mirror offering the best performance. With gaining significance of bulk document, audio, and video file transfer, smart mirror site selection has become a compelling task and thus motivated numerous automatic server selection approaches. Lately, we discussed in [4] the application of server selection to mobile agents. It has been pointed out that due to the rather small resource requirements of mobile agents, these must be careful when applying server selection. That is, since agents are idle while the server selection system processes their requests, server selection comes at a cost. Thus, agents will only access a server selection system if the utility function U is positive:

$$U = d_{\varnothing} - d_{min} - \delta \tag{1}$$

where $d_{\varnothing}$ is the average service time, d_{min} is the service time of a server recommended by a selection system, and δ is the agent idle time while the selection system processes its request. In the remainder of the paper, this problem will be

N. Mitrou et al. (Eds.): NETWORKING 2004, LNCS 3042, pp. 1440–1445, 2004.

referred to as the *rational server selection problem.* In [4] we presented a decision algorithm which solves the *rational server selection problem.* This paper briefly summarises our findings in [4] and analyses the performance of the algorithm with the help of a queuing model.

2 Rational Server Selection

The problem in developing a decision algorithm for the *server selection problem* is that the agents neither know the server resource capacities nor the waiting time δ. Yet, it is assumed that they have a basic knowledge of the system heterogeneity. Thus, let R be the resource capacity distribution describing the probability that a randomly selected server has a resource capacity $\boldsymbol{c} \in \Re^m_{\geq 0}$, with $\bar{x} \leq \boldsymbol{c}$ (where $\leq$ is a componentwise comparison and $\bar{x} \in \Re^m_{\geq 0}$). With the help of R, $d_{\varnothing}$ can be estimated by:

$$d_{\varnothing}(\boldsymbol{r}) = \min\left\{d \in \Re_{\geq 0} \middle| \boldsymbol{r} \nabla(E[R] \cdot d) = 0\right\}$$

where $\boldsymbol{r}$ is the resource requirement and ∇ is the resource consumption operator defined as: the mapping of a process's resource requirement to the requirements remaining after consuming a specified capacity is defined as $\nabla : \Re^m_{\geq 0} \times \Re^m_{\geq 0} \mapsto \Re^m_{\geq 0}$. That is, if capacity $\boldsymbol{c}$ is required to satisfy a resource requirement $\boldsymbol{r}$ equation $\boldsymbol{r} \overline{\nabla} \boldsymbol{c} = \boldsymbol{0}$ holds. According to the definition of R, its average value $E[R]$ is the resource capacity an agent expects when it selects a server randomly or the number of alternative servers is 1. To estimate d_{min}, the random distribution R_n giving the maximum resource capacity $\boldsymbol{c}$, with $\bar{x} \leq \boldsymbol{c}$ out of n randomly selected servers is required (again, $\leq$ is the componentwise comparison). Obviously, R_n is an order statistic distribution [5] and thus it is given by: $R_n(x) = R_1(x)^n = R(x)^n$ and $\frac{dR_n(x)}{dx} = n \cdot R(x)^{n-1} \cdot \frac{dR(x)}{dx}$.

$E[R_n]$ is the average maximum server capacity when a server is selected out of n. Consequently, d_{min} can be estimated by function $d_{min}(\boldsymbol{r}, n)$ as follows:

$$d_{min}(\boldsymbol{r}, n) = \min\left\{d \in \Re_{\geq 0} \middle| \boldsymbol{r} \nabla(E[R_n] \cdot d) = 0\right\}$$

Finally, given the random distribution $Err(x)$ (prediction error distribution) that a prediction has a deviation of ξ with $\xi \leq \bar{\xi}$ for an interval of length h the estimated utility function is:

$$E[U(\boldsymbol{r}, n)] = d_{\varnothing}(\boldsymbol{r}) - d_{min}(\boldsymbol{r}, n) - \delta - E[Err(d_{\varnothing}(\boldsymbol{r}) - d_{min}(\boldsymbol{r}, n))] \quad (2)$$

3 Performance Analysis

Given that rational server selection is widely deployed it does not only improve the performance of individual agents. Just as much it influences the performance of the overall system: As server selection systems struggle to assign agents to the

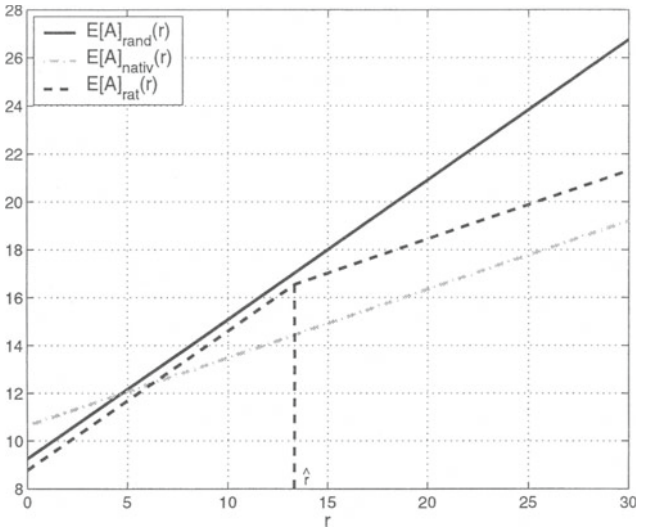

Fig. 1. Customer response time versus customer resource requirement

Table 1. Queuing model: Parameters

Para.	Description	Para.	Description
m	Number of servers (2)	μ_i	deterministic service rate of server i ($\mu_1 = 1, \mu_2 = 6$)
λ	customer arrival rate (exponentially distributed, $\lambda = 1$)	$\frac{1}{\alpha}$	Average service requirement (exponentially distributed, $\alpha = 0.55$)
$\hat{r}$	break-even resource requirement	δ	latency of server selection system interaction ($\delta = 10$)

best performing destination system, they effectively balance the server load and thus increase the throughput rate of the whole system. In general, load-balancing is just oriented at a single resource. Therefore, only one resource is considered in the analysis which is shared by the agents as is the case for network or processor resources. Thus, to compare random, native (server selection without application of the decision algorithm), and rational server selection (i.e. application of the decision algorithm) a simple $M/M/m$ queuing model is used: The service rates μ_i are deterministic, yet the customer's resource requirements are exponentially distributed with rate α. Table 1 summarises all model parameters and their settings in this analysis. To ease the analysis, it is assumed that customers, i.e. agents respectively, compete for the same kind of resource and that the server selection time is constant. Moreover, all servers are able to satisfy the customer's service requirements. Thus, on each request, the server selection system evaluates all servers. Finally, errors of the selection system have not been considered. Next

a performance model for rational server selection is developed. This model will finally be compared with random and native server selection.

If only a single resource is considered equation 2 can be simplified to:

$$E[U(r,n)] = \frac{r}{E[R]} - \frac{r}{E[R_n]} - \delta - E\Big[Err\big(\frac{r}{E[R]} - \frac{r}{E[R_n]}\big)\Big] \tag{3}$$

Consequently, for a fixed n there exists a break-even resource requirement $\hat{r}$ which meets $U(\hat{r},n) = 0$. That is, customers having resource requirement smaller than $\hat{r}$ select servers randomly; on the contrary, customers applying server selection have a requirement greater than $\hat{r}$. The break-even resource requirement is given as:

$$\hat{r} = \frac{\Big(\delta + E\big[Err\big(\frac{\hat{r}}{E[R]} - \frac{\hat{r}}{E[R_n]}\big)\big]\Big) \cdot E[R] \cdot E[R_n]}{E[R_n] - E[R]} \tag{4}$$

Thus, to model rational server selection two customer classes must be distinguished, namely customers which apply random selection and those which apply server selection. Looking at a server i the fraction of customers with resource requirement greater than $\hat{r}$ depends on its load share. To model the ratio between the two customer classes the divided exponential distribution is derived (see appendix). Its density function f is:

$$f(\hat{r},\omega,\alpha,x) = \frac{1}{\omega \cdot \big(1 - e^{-\alpha\cdot\hat{r}}\big) + (1-\omega)\cdot e^{-\alpha\cdot\hat{r}}} \cdot \begin{cases} \omega \cdot \alpha \cdot e^{-\alpha\cdot x} & x < \hat{r} \\ (1-\omega)\cdot\alpha\cdot e^{-\alpha\cdot x} & x \geq \hat{r} \end{cases} \tag{5}$$

where ω reflects the ratio between the customer classes. Thus, the mean $E[S_i](\hat{r})$ of a system's i service time is given by:

$$E[S_i](\hat{r}) = \int_0^\infty f(\hat{r},\omega_i,\alpha,x) \cdot \Big(\frac{x}{\mu_i}\Big) dx$$

With the help of density function f the ratio between two customer classes can be modelled. Yet, this approach requires a model transformation, i.e. an adaptation of the individual arrival rates. However, at first the weights ω_i must be derived. Preconditioned that none of the servers is overloaded when random selection is applied, customers with a requirement smaller than $\hat{r}$ are equally distributed among the servers. The others are fairly distributed among the servers where each server i receives a share of $\frac{\mu_i}{\sum_{j=1}^{m}\mu_j}$. Thus the relation ω_i at server i is:

$$\omega_i = \frac{1}{m \cdot \big(\frac{1}{m} + \frac{\mu_i}{\sum_{j=1}^{m}\mu_j}\big)}$$

The arrival rate of customers at server i, λ_i, is:

$$\lambda_i = \lambda \cdot \Big(\frac{1}{m} \cdot \big(1 - e^{-\alpha\cdot\hat{r}}\big) + \frac{\mu_i}{\sum_{j=1}^{m}\mu_j} e^{-\alpha\cdot\hat{r}}\Big)$$

where the first summand is the fraction of customers with resource requirements lower than $\hat{r}$ arriving at i and the second the fraction of those with requirements greater or equal than $\hat{r}$. By applying the Pollaczek-Kinchin formula [6], the average response time of a customer with resource requirement $r \in \Re_{\geq 0}$, $E[R]_{rat}(\hat{r}, r)$ is:

$$E[R]_{rat}(\hat{r}, r) = \sum_{i=1}^{m} \begin{cases} \frac{1}{m} \cdot \left(E[W_i]_{rat}(\hat{r}) + \frac{r}{\mu_i} \right) & r < \hat{r} \\ \frac{\mu_i}{\sum_{j=1}^{m} \mu_j} \cdot \left(E[W_i]_{rat}(\hat{r}) + \frac{r}{\mu_i} \right) + \delta & r \geq \hat{r} \end{cases} \tag{6}$$

where the average service time at server i is given by:

$$E[W_i]_{rat}(\hat{r}) = \frac{\lambda_i^2 \cdot E[S_i^2](\hat{r})}{2 \cdot (1 - \rho_{rat})}$$

$$\rho_{rat}(\hat{r}) = \sum_{i=1}^{m} \frac{\lambda}{m^2} \cdot E[S_i](\hat{r})$$

$E[S_i^2](\hat{r})$ is the second moment of the service time at server i. Apparently, at the break-even point $\hat{r}$ there is no difference between selecting customers randomly or server selection. Therefore, it can easily calculated by solving the equation:

$$\sum_{i=1}^{m} \frac{1}{m} \cdot \left(E[W_i]_{rat}(\hat{r}) + \frac{\hat{r}}{\mu_i} \right) = \sum_{i=1}^{m} \frac{\mu_i}{\sum_{j=1}^{m} \mu_j} \cdot \left(E[W_i]_{rat}(\hat{r}) + \frac{\hat{r}}{\mu_i} \right) + \delta$$

The random and native server selection performance models are special cases of the rational server selection model. That is, random selection corresponds to a rational server selection setting where the break-even point is infinite. Likewise, native server selection corresponds to rational server selection with a break-even point of zero. Thus, based on equation 6 derivation of the average response times of random server selection $E[R]_{rand}(r)$ and native server selection $E[R]_{native}(r)$ are straightforward and are given by:

$$E[R]_{rand} = \lim_{\hat{r} \to \infty} E[R]_{rat}(\hat{r}, r) \quad E[R]_{nativ} = E[R]_{rat}(0, r)$$

The performance evaluation has been restricted to a queuing system comprising only two servers. Though this scenario is quite simple, it is sufficient to illustrate the characteristics of rational server selection. Figure 1 shows the average customer response time versus customer resource requirements of the random, native and rational server selection approach. Apparently, the server selection approaches outperform random selection if a customer's resource requirement exceeds the breakeven point. However, those customers which use server selection even though their resource requirements are less than the breakeven point perform poor. Comparison of rational and native server selection shows: If rational selection is used customer's having less resource requirements perform well at the costs of those customers having significant requirements. But if native server selection is used, customer's having considerable requirements perform well at the costs of those having small.

4 Conclusions and Future Work

With the help of a simple performance model the performance characteristics of rational server selection have been discussed. According to this analysis, a relaxed load-balancing results in improved agent performance. Future work will focus on implementation of the decision algorithm in a server selection system and its evaluation in a real world scenario.

References

1. Gray, R.S., Kotz, D., Nog, S., Rus, D., Cybenko, G.: Mobile agents for mobile computing. Technical Report PCS-TR96-285, Dartmouth College, Computer Science, Hanover, NH (1996)
2. Theilmann, W., Rothermel, K.: Efficient dissemination of mobile agents. In: Proceedings. 19th IEEE International Conference on Distributed Computing Systems. Workshops on Electronic Commerce and Web-based Applications, Austin, TX, USA, IEEE Comput. Soc (1999) 9–14
3. Brewington, B., Gray, R., Moizumi, K., Kotz, D., Cybenko, G., Rus, D.: Mobile agents in distributed information retrieval. In Klusch, M., ed.: Intelligent Information Agents. Springer-Verlag, Germany (1999) 355–395
4. Pils, C., Diepolder, S.: Rational server selection for mobile agents. In Stefani, J.B., Demeure, I., Hagimont, D., eds.: 4th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS03). Volume 2893 of Lecture Notes in Computer Science (LNCS)., Paris, France, Springer-Verlag, Germany (2003) 61–72
5. Ogawa, J.: Distribution and moments of order statistics. In Sarhan, A.E., Greenberg, B.G., eds.: Contributions to order statistics. Wiley publications in statistics. John Wiley and Sons, Inc. (1962) 11–19
6. Kleinrock, L.: Queueing systems volume 1: Theory. John Wiley and Sons (1975)

Appendix

The divided exponential distribution models two customer streams arriving at a server A which are distinguished by different resource requirements. Basically, the resource requirements are exponentially distributed. However, dividing the customers in a number of streams according to weights $P(A|r < \hat{r})$ and $P(A|r \geq \hat{r})$ their resource requirements result in the divided distribution. Preconditioned, that the fraction of customers with resource requirement $r < \hat{r}$ and $r \geq \hat{r}$ are known, i.e. $P(A|r < \hat{r})$ and $P(A|r \geq \hat{r})$ respectively, the probability that a customer arriving at A has a resource requirement s is:

$$P(r|A) = \frac{P(r,A)}{P(A)} = \begin{cases} \frac{P(r,A,r<\hat{r})}{P(I)} & r < \hat{r} \\ \frac{P(r,A,r\geq\hat{r})}{P(I)} & r \geq \hat{r} \end{cases} \tag{7}$$

Since the streams descend from an exponential distribution with rate α, $P(A) = P(A|r < \hat{r}) \cdot \int_0^{\hat{r}} \alpha \cdot e^{-\alpha \cdot x} dx + P(A|r \geq \hat{r}) \cdot \int_{\hat{r}}^{\infty} \alpha \cdot e^{-\alpha \cdot x} dx$. Moreover, $P(r, A, r < \hat{r}) = P(A|r < \hat{r}) \cdot \alpha \cdot e^{-\alpha \cdot x}$, $P(r, A, r \geq \hat{r}) = P(A|r \geq \hat{r}) \cdot \alpha \cdot e^{-\alpha \cdot x}$. Finally, giving that $\omega = P(A|r < \hat{r})$ the density function of $P(r|A)$ is f (see equation 5).