Rational Server Selection for Mobile Agents

Carsten Pils and Stefan Diepolder

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

Abstract. As mobile agents have the ability to operate autonomously and in a disconnected way they are considered to suit mobile computing environments. Mobile users can dispatch agents into the fixed network where the agents operate in the users behalf. Thus, in contrast to client/server interactions agents do not suffer from poor performing wireless access networks. In this paper the performance of mobile agents and client/server interactions are analysed with respect to heterogeneous networks and server resources. It is argued that without a certain knowledge of the available resources agents can hardly decide whether they should migrate or just apply client/server calls to access a remote service. To this end, it is proposed that agents should access server selection systems in order to plan their migration strategy. However, while server selection systems process agent requests the agents are waiting idle. Thus, access to server selection systems comes at a cost and therefore agents must be careful about it. To solve this decision problem an algorithm is proposed which estimates the benefits of accessing server selection systems. Finally, the decision algorithm is evaluated with the help of a simulation model.

1 Introduction

Deployment of wireless access technologies like UMTS and WLAN along with the development of powerful hardware devices are leading to complex mobile applications which are characterised by compelling quality of service requirements. However, network resources are still scarce and therefore applications must efficiently exploit them. In recent years, mobile agents have been considered an attractive approach to cope with these limits and motivated many research studies investigating the benefits of mobile agent deployment in telecommunication systems. In contrast to traditional solutions, a mobile agent can leave the mobile device, migrate to the network location where a service is located, and perform a custom task. While the agent operates in the fixed network it filters information and therefore only the final results are transmitted back to the mobile device. Moreover, the mobile agent can continue its operations even though the mobile device is disconnected. Once the device re-connects, the agent returns its results to the user. Since the agent avoids network transmission of intermediate data and continues even in presence of network disconnections it is expected to complete the overall task much faster than a traditional client/server solution.

J.-B. Stefani, I. Demeure, and D. Hagimont (Eds.): DAIS 2003, LNCS 2893, pp. 61–72, 2003.
© IFIP International Federation for Information Processing 2003

Albeit it is apparent that applications operating at the border between fixed and mobile networks have to cope with heterogeneous resources and asymmetric network connections these conditions have not thoroughly investigated in terms of the mobile agent paradigm. However, application developer must understand the performance characteristics of the mobile agent paradigm to develop applications efficiently exploiting resources.

In this paper mobile agent performance in heterogeneous networks is analysed. It is argued that agents need some knowledge about the available resources to plan their migration strategy. To this end, it is proposed that agents should use server selection systems to plan their routes. Section 3 discusses whether the server selection approaches known from literature meet the agent requirements. It is concluded that only slight modifications to the known approaches are required. Since applying server selection comes at a cost a decision algorithm is proposed in section 4 which is analysed with the help of a simulation model described in section 5. Finally, section 6 concludes the paper.

2 Mobile Agent versus Client Server

In contrast to client/server approaches a mobile agent filters information at a remote server and returns the results rather than downloading all data and filtering it locally. Thus, the benefit of mobile agent deployment depends on the quotient of the valuable data size and the total data size to be filtered. Straßer and Schwehm who compared the performance of client/server and mobile agent approaches with the help of analytical models in [1] called this quotient *selectivity*. In their study they analysed the trade-off between the overhead of agent migration and the reduction of the reply size. Ismail and Hagimont investigated a similar scenario in [2] with the help of benchmarks and a network of globally distributed workstations. Pulafito et al analysed the scenario in [3] with stochastic Petri nets and showed that the network bandwidth has a major impact on the agent, i.e. if the network bandwidth is high the computation becomes the bottleneck of the agent interaction. Other studies [4] [5] confirm Puliafito's observation. Yet, none of them analysed the impact of network and server dynamics comprehensively.

2.1 A Simple Performance Model

Apparently, mobile agent and client/server have different resource requirements and therefore dynamics of server and network resource are expected to have a major impact on the performance comparison of the two paradigms. To characterise the impact of resources the performance of a mobile agent MA and its corresponding client/server solution CS is studied with the help of a simple analytic model. The task of both MA and CS is starting at a host A to access a service on a host B and to return the results to A. Furthermore, it is assumed that MA and CS are based on equivalent deterministic algorithms and use the same interfaces to access the service and to interact with the user.

Table 1. Capacities and resource requirements.

Parameter	Description	Parameter	Description
$\boldsymbol{r}_A$	MA's resource requirements at host A	$\boldsymbol{r'}_A$	CS's resource requirements at host A
$\boldsymbol{r}_{A,B}$	MA's resource requirements on the link between host A and B	$\boldsymbol{r'}_{A,B}$	CS's resource requirements on the link between host A and B
$\boldsymbol{r}_{B,A}$	MA's resource requirements on the link between host B and A	$\boldsymbol{r'}_{B,A}$	CS's resource requirements on the link between host B and A
$\boldsymbol{r}_B$	MA's resource requirements at host B	$\boldsymbol{r'}_B$	CS's resource requirements at host B
$\boldsymbol{c}_A$	Host A's resource capacity	$\boldsymbol{c}_B$	Host B's resource capacity
$\boldsymbol{c}_{A,B}$	Resource capacity of the server uplink link, i.e. A to B	$\boldsymbol{c}_{B,A}$	Resource capacity of the server downlink, i.e. B to A

Definition 1 (Process delay operator)
Given m the number of resources, a resource requirement $\boldsymbol{r} \in \Re^m_{\geq 0}$ of a process α, and a resource capacity $\boldsymbol{c} \in \Re^m_{\geq 0}$ the process delay operator $\otimes_\alpha : \Re^m_{\geq 0} \times \Re^m_{\geq 0} \mapsto \Re_{\geq 0}$ of a process α maps resource requirement $\boldsymbol{r}$ and the $\boldsymbol{c}$ resource capacity to the process execution time. If a resource capacity $\boldsymbol{c}$ does not meet an application's requirement $\boldsymbol{r}$ the process execution time is $\boldsymbol{r} \otimes_\alpha \boldsymbol{c} = \infty$.

Based on the process delay operator, process execution time T_{ma} of MA and interaction time T_{cs} of CS are given by (see table 1):

$$T_{ma} = \boldsymbol{r}_A \otimes_{ma} \boldsymbol{c}_A + \boldsymbol{r}_{A,B} \otimes_{ma} \boldsymbol{c}_{A,B} + \boldsymbol{r}_B \otimes_{ma} \boldsymbol{c}_B + \boldsymbol{r}_{B,A} \otimes_{ma} \boldsymbol{c}_{B,A}$$
$$T_{cs} = \boldsymbol{r'}_A \otimes_{cs} \boldsymbol{c}_A + \boldsymbol{r'}_{A,B} \otimes_{cs} \boldsymbol{c}_{A,B} + \boldsymbol{r'}_B \otimes_{cs} \boldsymbol{c}_B + \boldsymbol{r'}_{B,A} \otimes_{cs} \boldsymbol{c}_{B,A}$$

Network Resources. While MA must upload its code and data segment CS just sends short request messages. On the downlink however, MA transfers only filtered data while CS filters all data at host A. Thus, if MA's code and data size and CS's request size have minimum representation, CS operation requires only a single interaction, and the relevant link resources are bandwidth and delay the following inequality is satisfied:

$$\boldsymbol{r}_{A,B} \otimes_{ma} \boldsymbol{c}_{A,B} \geq \boldsymbol{r'}_{A,B} \otimes_{cs} \boldsymbol{c}_{A,B}$$

Furthermore, given that $\boldsymbol{c} = \boldsymbol{c}_{A,B} = \boldsymbol{c}_{B,A}$ it is concluded that (selectivity is greater 0):

$$\boldsymbol{r}_{A,B} \otimes_{ma} \boldsymbol{c} + \boldsymbol{r}_{B,A} \otimes_{ma} \boldsymbol{c} \leq \boldsymbol{r'}_{A,B} \otimes_{cs} \boldsymbol{c} + \boldsymbol{r'}_{B,A} \otimes_{cs} \boldsymbol{c}$$
$$\Rightarrow \boldsymbol{r'}_{B,A} \otimes_{cs} \boldsymbol{c} \geq \boldsymbol{r}_{B,A} \otimes_{ma} \boldsymbol{c}$$

Consequently, the uplink is the critical resource of MA while the downlink is critical to CS. In presence of asymmetric connections with considerably poor uplink

performance and a well performing downlink, CS outperforms MA even though selectivity is high. On the other hand a poor downlink and well performing uplink support the MA paradigm. If CS requires more than one request/replay pair to reproduce MA functionality, MA benefits from its batch job alike properties (i.e. an agent requires only a single migration and transfer of results). CS, however, suffers from network latency (this effect has been analysed by Puliafito et al who assumed that the number of CS interactions is geometrical distributed).

Processing Resources. Apart from network resources, the benefit of mobile agent deployment depends on the performance of the client and the server machine. Apparently, agent migration, data filtering, and transferring results consume more processing resources than a single agent migration and thus following inequality is satisfied ($\boldsymbol{c} = \boldsymbol{c}_A = \boldsymbol{c}_B$):

$$\boldsymbol{r}_B \otimes_{ma} \boldsymbol{c} \geq \boldsymbol{r}_A \otimes_{ma} \boldsymbol{c}, \forall \boldsymbol{c} \in \Re^m_{\geq 0}$$

Agent management, data retrieval and filtering, and the transfer of results consume more processing resources than data retrieving and forwarding, thus it is concluded that:

$$\boldsymbol{r}_B \otimes_{ma} \boldsymbol{c_B} \geq \boldsymbol{r'}_B \otimes_{cs} \boldsymbol{c_B}, \forall \boldsymbol{c_B} \in \Re^m_{\geq 0}$$

$$\boldsymbol{r}_B \otimes_{ma} \boldsymbol{c}_B \geq \boldsymbol{r'}_A \otimes_{cs} \boldsymbol{c}_A, \quad \forall \boldsymbol{c_B} = \boldsymbol{c_A} \in \Re^m_{\geq 0}$$

where $\boldsymbol{c} = \boldsymbol{c}_A = \boldsymbol{c}_B$. Consequently, the load of host B has a major impact on the agent's performance. The benchmark study of Gray et al in [4] shows that even though selectivity is high the client server approach outperforms the mobile agent approach if the server is overloaded and the network bandwidth is high. Johansen showed in [6] that sharing the computational resources between client and server can improve the overall agent performance.

2.2 Agent Migration Strategies

As these rather simple observations show, resource capacities have a strong influence on the optimal (in the sense of smallest response time) agent migration strategy. Particularly, if an agent has the choice between alternative hosts, knowledge of network and host resource becomes crucial for its migration strategy. But, as long as neither network nor hosts provide any quality of service guarantees the resource capacities are hardly known in advance. A solution to this problem are server selection systems. Server selection systems use active resource probing or resource monitoring to route clients to well performing mirror sites or give feedback about server performance. From a client's perspective the benefit of server selection can be expressed by the difference between the average service time $d_\varnothing$ and the service time of the selected server d_{min}.

3 Server Selection for Mobile Agents

Application of server selection to mobile agents is not new. Gray et al proposed in [7] the integration of a server selection system in an agent system. Theilmann

and Rothermel studied in [8] so-called dynamic distance maps for mobile agents and claim that agent communication costs are reduced by up to 90%. In [9] Brewington et al propose an agent migration planning support applying a so-called network sensing tool. The way systems measure the network performance has a major impact on their precision. In literature two basic measurement approaches are distinguished: active and passive measurements.

3.1 Active Measurements

By injecting a stimulus into the network and measuring the response active measurement systems estimate the round trip time and the bandwidth. Pathchar [10] and bprobe [11] are prominent examples of active measurement tools. To measure the link speed these tools send successive ICMP ECHO packets and measure the inter-arrival time of the responses. Apparently, active measurement can be used to get information about every connection, however the measurement might take some seconds. Thus, only applications which exchange a large amount of data can benefit from these approaches. Another class of probe based approaches where clients do not suffer from long measurement times are Internet distance maps [12], [13], [14]. Instead of measuring the connections between each client/server pair Internet distance maps probe only the connection performance between dedicated landmarks.

3.2 Passive Measurements

In contrast to active measurement tools, passive tools do not probe the network, but monitor the traffic. Similar to Internet distance maps these tools can instantly provide measurement results, however, there is still a considerable chance that they fail to provide any information. If there has been no traffic between peers of interest passive approaches cannot provide precise information. A representative of such a passive approach is the Shared Passive Network Performance Discovery (SPAND) tool [15] proposed by Seshan et al. A different approach has been proposed by Andrews et al [16] who propose to cluster networks with respect to passive measurements.

3.3 IP-Anycast

None of the server selection approaches discussed so far considers host performance. In [17] Bhattacharjee et al propose an IP-Anycast system where anycast resolvers collect sample data from their care of replica servers and automatically route clients to a well performing one. IP-Anycast is a feature of the IPv6 protocol and is used to address a group of replica servers. A message send to an anycast address will be received by only one server of the group. It is up to the anycast resolver to decide which server is going to receive the message. In the light of agent migration strategies, IP-Anycast has still two disadvantages: Firstly, none of the known anycast approaches provides any performance information to the

agent. Secondly, an agent having the choice between different anycast groups can hardly decide. However, the concept of the distributed sample database applied by the anycast resolvers is able to provide the lacking information. Thus, the systems can easily be enhanced to provide the required performance feedback.

3.4 The Server Selection Problem

The benefit of server selection depends on the number of servers to select from and their heterogeneity. The larger the set and the more heterogeneous the servers are the greater the benefit. However, server selection does not come for free as it requires retrieval and processing of sample data. Accordingly, the cost function C_s of server selection is defined as the agent idle time during the selection process and therefore the utility function U is given by: $U = d_{\varnothing} - d_{min} - C_s$. A consequence of this simple observation is that applying server selection might cause performance loss. In the remainder, this problem is called the *server selection problem.*

4 Rational Server Selection

To analysis the *server selection problem* an omniscient agent is assumed which knows the resource capacity function $\boldsymbol{c}_i(t) : \Re_{\geq 0} \mapsto \Re^m_{\geq 0}$ of each server i. $\boldsymbol{c}_i$ returns the resource capacity vector of server i at time t.

Definition 2 (Resource consumption operator)
The resource consumption operator of a process α $\nabla_\alpha : \Re^m_{\geq 0} \times \Re^m_{\geq 0} \mapsto \Re_m$ *maps* α*'s resource requirements to the requirements remaining after consuming a specified resource capacity.*

Thus, $\boldsymbol{r} \nabla \boldsymbol{c} = \boldsymbol{0}$ indicates that the resource requirement $\boldsymbol{r}$ is satisfied if the corresponding process has at least consumed capacity $\boldsymbol{c}$. Given the agent's resource requirement vector $\boldsymbol{r} \in \Re^m_{\geq 0}$ the service time $d_i(\boldsymbol{r})$ of server i is:

$$d_i(\boldsymbol{r}) = \min \left\{ x \in \Re_{\geq 0} \middle| \boldsymbol{r} \nabla \int_{t_0}^{t_0+x} \boldsymbol{c}_i(t) dt = 0 \right\}$$

The server selection utility function $U(\boldsymbol{r})$ is:

$$U(\boldsymbol{r}) = \sum_{i=1}^{n} \frac{d_i(\boldsymbol{r})}{n} - \min \left\{ d_1(\boldsymbol{r}), \ldots, d_n(\boldsymbol{r}) \right\} - C_s$$

In general, $\boldsymbol{c}(t)$ is even unknown to the server selection system (let alone to the agent). All information a system has is the latest sample data $\boldsymbol{c}_i(t_0)$ which is used to approximate $\boldsymbol{c}(t)$ over an interval $[t_0, t_0 + h]$ by $\boldsymbol{c}(t_0) \cdot h$. Evidently, the absolute approximation error $E_A(h)$ increases with the length of the interval and the instability of the server. Measurements of Internet servers [18] give reason to the assumption that well performing servers are stable. Furthermore, it is

expected that well designed server selection algorithms balance the server load and therefore stabilise them. Thus, it is assumed that the gradient of $E_A(h)$ in h is rather small.

4.1 The Estimated Utility Function

The problem in developing a decision algorithm for the *server selection problem* is that the agent does not know the server resource functions $\boldsymbol{c}_i(t)$. However, it is assumed that the agents have a basic knowledge of the system heterogeneity. Let R be the resource capacity distribution describing the probability that a randomly selected server has a resource capacity $X \leq \bar{x}$ ($X, \bar{x} \in \Re^m_{\geq 0}$). With the help of R $d_\varnothing$ can be estimated by:

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

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 that n randomly selected servers have a resource capacity $X \leq \bar{x}$ is required. Obviously, R_n is an order statistic distribution [19] and thus it is given by:

$$\begin{aligned} R_n(x) &= R_1(x)^n = R(x)^n \\ \frac{dR_n(x)}{dx} &= n \cdot R(x)^{n-1} \cdot \frac{dR(x)}{dx} \\ E[R_n] &= \int_0^\infty x \cdot n \cdot R(x)^{n-1} \cdot \frac{dR(x)}{dx} \end{aligned}$$

$E[R_n]$ 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:

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

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

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

As for the resource capacities, the waiting time is not known in advance. Particularly, if the server selection system must retrieve sample data from a remote location, the waiting time is hardly known. To this end, the waiting time is assumed to be distributed according to a random distribution F, i.e. C_s is given by F.

4.2 Decision Algorithm

The more time an agent waits the lesser is the benefit of server selection. The agent's situation is similar to the one of a customer waiting for a call centre

service. While the customer is waiting for a call centre agent he is weighting the costs of the call (patience or connection costs) against the expected service utility. Since the costs are increasing monotonic in time the customer's question is when he should put down the receiver. The customer's decision problem has gained some attention in literature and is known as the *problem of rational abandonment from invisible queues*. Mandelbaum and Shimkin have analysed the problem in [20]. They proved that there exist an optimal abandonment time T which depends on the monotonic properties of the hazard rate function $H(t) = \frac{dF(t)/dt}{1-F(t)}$. According to the rational server selection problem Mandelbaum and Shimkin define the utility function U_t with respect to the system abandonment time T by:

$$U_t(\boldsymbol{r}, n, T) = E[G(\boldsymbol{r}, n) \cdot I(T \geq V) - \min\{V, T\}]$$

where V is the agent's waiting time, I maps to 1 if the specified condition is true, and $G(\boldsymbol{r}, n)$ is the server selection gain which is given by

$$G(\boldsymbol{r}, n) = d_{\varnothing}(\boldsymbol{r}) - d_{min}(\boldsymbol{r}, n) - E[Err(d_{\varnothing}(\boldsymbol{r}) - d_{min}(\boldsymbol{r}, n))]$$

Considering the waiting time distribution, it is derived:

$$U_t(\boldsymbol{r}, n, T) = \int_0^T \big(G(\boldsymbol{r}, n) - t\big)\, dF(t) - T \cdot \big(1 - F(T)\big)$$

Preconditioned that $F(T)$ is continuously differentiable for $t > 0$ and that $F(t)dt$ has a right limit at 0 they differentiate F with respect to $T > 0$ and get:

$$\begin{aligned}\frac{dU_t(\boldsymbol{r}, n, T)}{dT} =& \big(G(\boldsymbol{r}, n) - T\big)\frac{dF(T)}{dT} - \frac{\big(1 - dF(T)\big)}{dT} + T \cdot \frac{dF(T)}{dT} \\ =& \big(G(\boldsymbol{r}, n)\big) \cdot \big(1 - F(T)\big) \cdot \left(H(T) - \frac{1}{G(\boldsymbol{r}, n)}\right)\end{aligned}$$

The first order condition for a local optimum at $T > 0$, namely, $\frac{dU_t(\boldsymbol{r},n,T)}{dT} = 0$, can be stated as: $H(T) = \frac{1}{G(\boldsymbol{r},n)}$ Consequently, the local optimum of U_t is the optimal waiting time in terms of the rational selection problem.

4.3 Consequences to the Migration Strategy

To apply the proposed decision algorithm, agents require the average values of the resource distribution and its corresponding n order statistic distributions. Since server selection systems monitor the resources anyway, the systems can calculate the average values and provide them to the agents. According to the decision algorithm, agents can only benefit from server selection if they have the choice between different destination hosts. But even though there is only a single destination host, an agent can still benefit from server selection: Firstly, agents can use the average value of the resource distribution to decide whether agent migration or client/server interaction is statistically optimal for operating an information retrieval task. Secondly, the more agents use server selection the more balanced the resources are.

5 Evaluation of the Rational Server Selection Approach

In this section the system dynamics and their implications to the resource distribution R are studied in case that the fraction of agents which use server selection increases. To this end, let $R^x,\ x \in [0,1]$ a series of random distributions and x be the fraction of agents applying server selection.

The study is subdivided into two parts: Firstly, some simple general observations are made. In the second part a simulation study is presented.

5.1 General Observations

Since server selection balances the load of the resources it is assumed that the standard deviations $\sigma(R^x)$ satisfy the inequality below given that the agent arrival rate is constant:

$$\sigma(R^x) \leq \sigma(R^y), x \geq y$$

Preconditioned that the investigated system resources can be modelled by $G/G/1$ queuing models and that the server resources are shared, the average values $E[R^x]$ satisfy the inequality:

$$E[R^x] \leq E[R^y], x \geq y$$

This stems from the fact, that the agent average waiting time decreases with the standard deviation of the resources. Consequently, the average number of agents in the system decreases as well. According to the lower and upper bounds of n order statistics found by Morigurti [21] and Huang [22] $E[R_n^x]$ satisfies the inequalities:

$$-\sigma(R^x) \cdot \sqrt{\int_0^1 [\underline{\varphi}_{1,n}(t)]^2 dt - 1} \leq E[R_n^x] - E[R^x] \leq \sigma(R^x) \cdot \sqrt{\int_0^1 [\underline{\varphi}_{n,n}(t)]^2 dt - 1}$$

where $\underline{\varphi}_{1,n}$ and $\underline{\varphi}_{n,n}$ are the greatest convex minorants of the first and second order statistic of the standard uniform distribution. Thus, as expected $E[R_n^x]$ decreases in x as well.

5.2 Simulation Study

The decision algorithm has been evaluated with the help of a simple simulation model. The model comprises a set of 16 servers which have service capacities of $2000 ops/sec$, $4000 ops/sec$, $8000 ops/sec$, and $16000 ops/sec$. Each of them serves incoming customers in a round robin fashion. Arrival rate and customer's resource requirements are exponential distributed with mean $1 \frac{arrival}{ms}$ and $10 ops$ respectively. The waiting time distribution F has been considered to be deterministic with value $10ms$, i.e. $C_s = 10ms$. Based on this model the following three scenarios have been evaluated:

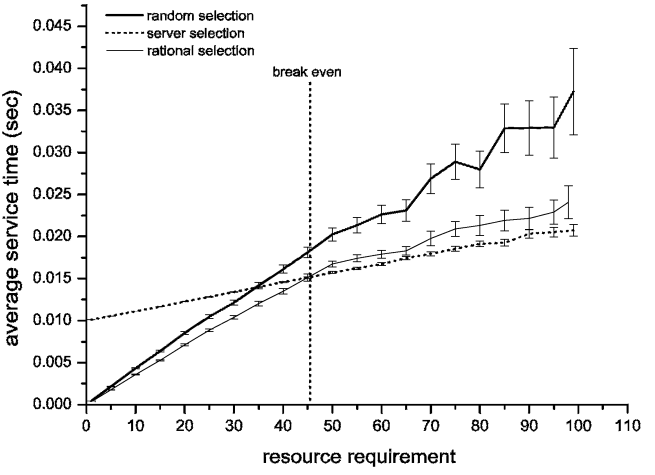

Fig. 1. Service latencies versus resource requirement.

1. **Random selection:** The customers select the destination server randomly.
2. **Native Server selection:** The customers use server selection to select one server out of a random set of three alternatives.
3. **Rational selection:** Like **native server selection** customers select the destination out of a random set. However, in contrast to **native server selection** the customers use the decision algorithm proposed in the previous section. Thus, only those customers having a significant resource requirement apply server selection.

The average service latencies with respect to the service requirements are shown in figure 1. 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.

6 Conclusions

Evidently, mobile agent and client/server based applications have different resource requirements. With the help of a simple analytic model the impact of resources on the performance of the two paradigms has been discussed. As a consequence of this study it has been argued that agents require some knowledge about the system resources when planning a migration strategy. However, as long as neither network nor server resources provide any quality of service

guarantees information about resource capacity are hardly obtained. In recent years, server selection systems have been proposed to estimate resource capacities of best-effort systems. Selection approaches which are known from literature have been investigated with respect to the agent migration planning problem. It has been concluded that only slight modifications to the known approaches are required to meet agent requirements. But unfortunately, server selection is not for free and thus agents must be careful about accessing these systems. To this end, a decision algorithm has been proposed which helps agents to decide whether there is any advantage in applying server selection. Apparently, this decision algorithm applies to any network application and is not restricted to agent technology. Finally, the decision algorithm has been studied with the help of a simple simulation model. In the simulation model the customers are competing for just a single resource type. The simulation shows that the decision algorithm outperform random selection. Since native selection (i.e. the agents do not use the decision algorithm) perfectly balances the load, it outperforms rational server selection if the agent resource requirements are significant.

References

1. M. Straßer and M. Schwehm, "A performance model for mobile agent systems", in *Proc. of the International Conference on Parallel and Distributed Processing Techniques and Applications*, Las Vegas, July 1997, pp. 1132–1140.
2. L. Ismail and D. Hagimont, "A performance evaluation of the mobile agent paradigm", in *ACM SIGPLAN Notices*, October 1999, vol. 34, pp. 306–313.
3. A. Puliafito, S. Riccobene, and M. Scarpa, "An analytical comparison of the client-server, remote evaluation, and mobile agents paradigms", in *First International Symposium on Agent Systems and Applications/Third International Symposium on Mobile Agents*, Palm Springs, USA, October 1999, pp. 278–92, IEEE Comput. Soc.
4. Robert S. Gray, David Kotz, Ronald A. Peterson, Peter Gerken, Martin Hofmann, Daria Chacon, Greg Hill, and Niranjan Suri, "Mobile-Agent versus Client/Server Performance: Scalability in an Information-Retrieval Task", in *Lecture Notes in Computer Science (LNCS). Mobile Agents. 5th International Conference*, G.P. Picco, Ed., Hanover, NH, December 2001, vol. 2240, pp. 229–243, Springer-Verlag, Berlin.
5. D. Kotz, G. Cybenko, R.S. Gray, Jiang Guofei, R.A. Peterson, M. Hofmann, D.A. Chacon, and K.R. Whitebread, "Performance analysis of mobile agents for filtering data streams on wireless networks", *Mobile Networks and Applications*, vol. 7, no. 2, pp. 163–174, 2002.
6. D. Johansen, "Mobile agent applicability", in *Proceedings of the 2nd Workshop on Mobile Agents (MA'98)*, K Rothermel and F. Hohl, Eds. 1998, vol. 1477 of *Lecture Notes in Computer Science (LNCS)*, pp. 80–98, Springer-Verlag, Germany.
7. Robert S. Gray, David Kotz, Saurab Nog, Daniela Rus, and George Cybenko, "Mobile agents for mobile computing", Tech. Rep. PCS-TR96-285, Dartmouth College, Computer Science, Hanover, NH, May 1996.
8. W. Theilmann and K. Rothermel, "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, May 1999, pp. 9–14, IEEE Comput. Soc.

9. B. Brewington, R. Gray, K. Moizumi, D. Kotz, G. Cybenko, and D. Rus, "Mobile agents in distributed information retrieval", in *Intelligent Information Agents*, M. Klusch, Ed., chapter 15, pp. 355–395. Springer-Verlag, Germany, 1999.
10. V. Jacobson, "Pathchar", `ftp://ftp.ee.lbl.gov`, 1997.
11. Robert L. Carter and Mark E. Crovella, "Measuring bottleneck link speed in packet-switched networks", *Perform. Eval. (Netherlands)*, vol. 27-28, pp. 297–318, October 1996.
12. W. Theilmann and K. Rothermel, "Dynamic distance maps of the internet", in *Proceedings IEEE INFOCOM 2000*, Tel Aviv, Israel, March 2000, vol. 1, pp. 275–84, IEEE Comp. Soc.
13. K. M. Hanna, N. Natarajan, and B.N. Levine, "Evaluation of a novel two-step server selection metric", in *Proceedings Ninth International Conference on Network Protocols. ICNP 2001*, Riverside, CA, USA, November 2001, pp. 290–300, IEEE Comput. Soc.
14. S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologically-aware overlay construction and server selection", in *Proceedings IEEE INFOCOM 2002 Conference on Computer Communications*, New York, NY, USA, June 2002, vol. 3, pp. 1190–1199, IEEE Comp. Soc.
15. S. Seshan, M. Stemm, and R.H. Katz, "Spand: Shared passive network performance discovers", in *USENIX Symposion on Internet Technologies and Systems*. 1997, pp. 135–146, USENIX Association.
16. M. Andrews, B. Shepherd, A. Srinivasan, and F. Winkler, P.and Zane, "Clustering and server selection using passive monitoring", in *Proceedings IEEE INFOCOM 2002 Conference on Computer Communications*, New York, NY, USA, June 2002, vol. 3, pp. 1717–1725, IEEE Comp. Soc.
17. Samrat Bhattacharjee, Mostafa H. Ammar, Ellen W. Zegura, Viren Shah, and Zongming Fei, "Application-layer anycast", in *IEEE INFOCOM '97*, Kobe, Japan, April 1997, vol. 3, pp. 1388–96, IEEE Comput. Soc. Press.
18. A. Myers, P. Dinda, and H. Zhang, "Performance characteristics of mirror servers on the internet", in *Proceedings of IEEE INFOCOM'99*, New York, NY, USA, March 1999, vol. 1, pp. 304–312, IEEE Comp. Soc.
19. Junjiro Ogawa, "Distribution and moments of order statistics", in *Contributions to order statistics*, Ahmed E. Sarhan and Bernhard G. Greenberg, Eds., Wiley publications in statistics, pp. 11–19. John Wiley and Sons, Inc., 1962.
20. Avishai Mandelbaum and Nahum Shimkin, "A model for rational abandonments from invisible queues", *Queueing Systems*, vol. 36, no. 1-3, pp. 141–173, 2000.
21. S. Morigurti, "A modification of schwarz's inequality with application to distributions", *Annual Math. Statist.*, vol. 24, pp. 107–113, 1953.
22. J. S. Huang, "Sharp bounds for the expected value of order statistics", *Statistics and Probability Letters*, vol. 33, pp. 105–107, 1997.