Cost Model and Adaptive Scheme for Publish/Subscribe Systems on Mobile Environments

Sangyoon Oh[1,2], Sangmi Lee Pallickara[2], Sunghoon Ko[1],
Jai-Hoon Kim[1,3], Geoffrey Fox[1,2]

[1] Community Grids Lab., Indiana University, Bloomington, IN. U.S.A.
{ohsangy,suko,jaikim,gcf}@indiana.edu
[2] Department of Computer Science, Indiana University, Bloomington, IN. U.S.A.
leesangm@cs.indiana.edu
[3] Graduate School of Information and Comminucations, Ajou University, Suwon, S. Korea
jaikim@ajou.ac.kr

Abstract. One of main advantages of publish/subscribe systems is decoupling of publishers and subscribers in time, space, and synchronization. Thus publish/subscribe model is appropriate in many push based data dissemination applications such as data dissemination services, information sharing, service discovery, etc. However, to our best knowledge, research of performance modeling and adaptive schemes for publish/subscribe has not been announced yet. This paper presents cost model for publish/subscribe systems, analyze its performance, and compare to other interaction-based models such as client-server model and polling models. Based on the cost analysis, we propose adaptive model which can dynamically select an appropriate model for each client independently.

1 Introduction

Publish/subscribe system have been widely used in many applications [8, 9, 10, 11]. Publish/subscribe system consists of publisher (ES: Event Source), server (EBS: Event Brokering System), and subscriber (ED: Event Displayer). After publisher publishes data (events) asynchronously to a server, the server disseminates the data (events) to subscribers which registered its interest on the server. Thus publish/subscribe model is appropriate in many applications such as as data dissemination services [12], information sharing [13], service discovery [14], etc. As these kinds of services are popular in mobile and ubiquitous environments, publish/subscriber model will be more widely used. Figure 1 depicts system configurations of publish/subscribe systems for mobile environments.

Many researches have been performed so far to propose architecture, add useful functions, and improve performance of publish/subscribe systems, including Siena [15], Gryphon [16], JEDI [17], Rebeca [18], Elvin [19]. However, to our best knowledge, research of performance modeling and adaptive schemes for publish/subscribe has not been announced yet. In this paper, we present cost model for pub-

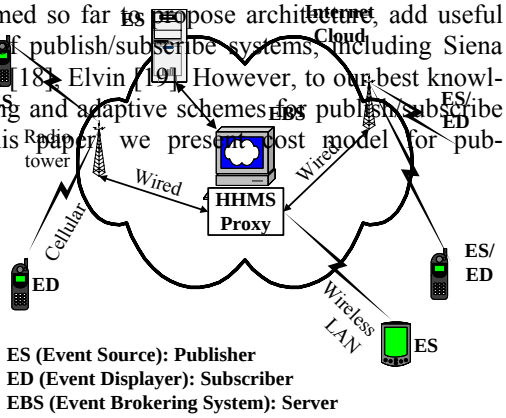

Figure 1. Pub/Sub System Configurations

lish/subscribe systems, analyze its performance, and compare to other interaction based models such as client-server model and polling models. We can estimate performance and effectively adopt publish/subscribe systems by using our proposed cost model and analysis of publish/subscribe systems. Based on the cost analysis, we propose adaptive model which can dynamically select an apropriate model (ex. publish/subscribe, request/reply, polling models) for each client independently. We believe the adaptive scheme we introduce here is very useful for the mobile and ubiquitous services where characteristics of device and networks are diverse and dynamically change. In mobile and ubiquitous services, many types of mobile devices are used and its performance, system resources, executing application, and user's use pattern are all different. Thus, independent model selection for each mobile device, service, and user is very useful for mobile and ubiquitous environments.

As a summary, cost analysis model and adaptive scheme can be used as follows:

- *static* model selection: We can choose one of appropriate models for all devices depending on system and application parameters.
- *hybrid* model selection: Each device can adopt an appropriate model independently, which is very useful for mobile and ubiquitous environments where mobile devices and users' preference are diverse and numerous.
- *dynamic* model selection: Model can be changed during a service according to change of status of system and network, which is common in mobile and ubiquitous services.

Our analysis shows that publish/subscribe model are appropriate in many cases and adaptive scheme are essential especially in mobile and ubiquitous environments where mobile devices and users' preference are diverse and numerous and status of system and network are dynamic. We can easily give many examples that publish/subscriber model and adaptive scheme have advantages as follows:

- Broadcast notification services in many areas such as real-time sports news, stock market, etc. (publish/subscribe model)
- Many applications such as location based services are available using many types of devices and communication protocols (adaptive scheme: hybrid model selection).
- Users can alternatively choose on/off-line or power on/off to save communication cost or batter power, or during their movement (publish/subscribe model, adaptive scheme: dynamic model selection).
- Users can alternatively use wired or wireless connection (Ethernet or CDMA) during services (adaptive scheme: dynamic model selection)
- Programmer can choose model according to data access patterns and system parameters for designing application (adaptive scheme: static model selection)
- System manager can choose model according to service characteristics (adaptive scheme: static model selection)
- Users can choose model according to their preference (adaptive scheme: hybrid model selection)
- System can automatically choose model for each user according to his/her preference or use pattern (adaptive scheme: hybrid model selection)

We also experimentally measured and compared performance of publish/subscribe model to client/server model on our test bed including mobile device and

NaradaBrokering [4] (our publish/subscribe based message brokering system) to verify correctness of our performance model on the real systems. Our cost analysis model is simple but accordant with experimental results

2 Cost Model

In this section, we present the system models and examine the analytic cost model of three different models; publish/subscribe, request/reply, and periodic polling models.

2.1 System Models

To evaluate the cost model for different systems, we assume following basic system parameters to analyze cost.

- α : publish rate in publish/subscribe model
- β : event access rate in publish/subscribe model or data request rate in client-server model
- c_{ps} : publish/subscribe cost per event, c_{pub} (cost for publish event) + c_{sub}(cost for subscribe event)
- c_{rr}: cost per request and reply in request/reply (client-server) model.
- $c_{poll}(\alpha,T)$: cost of periodic publish or polling, where T is length of period.
- $c_d(\alpha,T)$: cost of delaying publish in polling model
- s(n) : effect of sharing among n subscribers, e.g., server can deliver events with low cost when it broadcasts event to many subscribers.
- t_{ps} : time delay for publish/subscribe, t_{pub})(ime delay for publish) + t_{sub}(time delay for subscribe)
- t_{rr} : time delay for request and reply
- $t_{poll}(\alpha, T)$: time delay for periodic publish.

2.2 Cost Analysis

In this analysis, we analyze cost of three different models without any failure of communication link or node. We consider (1) conceptual total cost (e.g., the number of message, amount of message, or time delay) per unit time for each model, (2) cost for each access by client (or subscriber), (3) time delay for access after subscriber's (or client's) intention, and (4) time delay between event occurrence and notification to subscriber (or recognition by client). Cost can be the number of message, amount of message, or time delay. Table 1 shows the summary of the cost for each model analyzed in this paper.

Table 1. The cost of the selected model

Model	Publish/Subscribe	Request/Reply	Polling
conceptual total cost per time unit	α (c_{pub} + n s(n)c_{sub})	β n c_{rr}.	($c_{poll}(\alpha,T)$ + $c_{delay}(\alpha,T)$) / T
cost for each access	$\frac{\alpha}{\beta}(\frac{c_{pub}}{n} + c_{sub})$	c_{rr}	$c_{poll}(\alpha,T)$ + $c_{delay}(\alpha,T)$
time delay between intention and access	0	t_{rr}	T/2

time delay between event occurrence and notifica-tion/recognition (or access)	$t_{ps} = t_{pub} + t_{sub}$ $(t_{ps} = t_{pub} + t_{sub+} \frac{1}{\beta})$	$\frac{1}{2\beta}$	T/2

Cost of publish/subscribe model

Since we assume that c_{pub} is cost for that ES(Even Source) publish events to EBS(Event Brokering System), and c_{sub} is cost for that ED(Event Displayer) subscribes events from EBS(Event Brokering System), cost of publish/subscribe model for each event publish and subscribe is $c_{pub} + n\ s(n)c_{sub}$. Please remember that n is the average number of subscriber and s(n) is sharing effect among n nodes. When publish rate is α, cost per time unit is $\alpha\ (c_{pub} + n\ s(n)c_{sub})$. Now, we consider cost in the view point of subscriber (per each event access of subscriber). We analyze three performance metrics, (1) conceptual cost for each access, (2) time delay for subscriber to access event after its intention, (3) and time delay until notification to subscriber after event occurring. The average number of event occurred before each access is cost for each access: $\sum_{i=0}^{\infty} \frac{\beta}{\alpha+\beta}\left(\frac{\alpha}{\alpha+\beta}\right)^{i} = \frac{\alpha}{\beta}$, where c_{pub} is shared among n subscriber and c_{sub} is required for each subscriber. Thus, average cost for each access is $\frac{\alpha}{\beta}(\frac{c_{pub}}{n} + c_{sub})$. There is no time delay for access after subscriber's intention since event has already been received. Time delay between event occurrence and notification to subscriber is $t_{ps} = t_{pub} + t_{sub}$.

Cost of request/reply model

Cost for each request and reply is assumed to c_{rr}. Thus total cost is nc_{rr}, where n is the number of client. When request rate is β, cost per time unit is: $\beta\ n\ c_{rr}$. Time delay for access after client's intention is t_{rr} as we assume. Time delay between event occurrence and recognition of subscriber is depends on request rate (similar to polling rate): $\frac{1}{2\beta}$

Periodic (polling) model

Periodic model is appropriate for applications in which delayed message is acceptable. Cost of periodic model (periodic publish or polling) per period is $c_{poll}(\alpha, T) + c_{delay}(\alpha, T)$. Thus, cost per time unit is $(c_{poll}(\alpha, T) + c_{delay}(\alpha, T)) / T$, where $c_{poll}(\alpha, T)$ can be between c_{rr} and $\alpha Tc_{rr.}$ If we assume periodic publish, cost per time unit is $(c_{pub}(\alpha, T) + n\ s(n)\ c_{sub}(\alpha, T) + c_{delay}(\alpha, T))/T$, where $c_{pub}(\alpha,T)$ is between c_{pub} and αTc_{pub}, $c_{pub}(\alpha,T)$ and $c_{sub}(\alpha,T)$ is be between c_{sub} and αTc_{sub},, and $c_{delay}(\alpha,T)$ is proportional to between c_{delay} and αTc_{delay}. Average time delay for access after client's intention is T/2. Time delay between event occurrence and recognition of subscriber is T/2.

3. Adaptive Scheme

In this section, we describe adaptive scheme that can choose an appropriate model among publish/subscribe and request/reply models. Each client node can select its own model independently (hybrid model) and change its model during its service (dynamic model). Our adaptive scheme based cost analysis presented in section 2.

In this paper, we consider cost for each client's access as a cost metric. During a period of time, the average number of events occurred per client's access is measured for each client. At the end of the period, the average cost for each client's access is computed using the analysis in section 2, which is $\frac{\alpha}{\beta}(\frac{c_{pub}}{n} + c_{sub})$, where $\frac{\alpha}{\beta}$ is average number of event occurred per client's access and n is the number of subscriber. In our adaptive scheme, average number of event and the number of subscriber are obtained experimentally during the execution of application. At the end of the period, the model that is expected to require less cost than the other model during the following period is selected independently for each client. We can summarize our adaptive model as follows:

(1) During the period of time, average number of event occurred per client's access is measured for each client.

(2) If $\frac{\alpha}{\beta}(\frac{c_{pub}}{n} + c_{sub}) > c_{rr}$, choose request/reply model for the next period.

(3) else, choose publish/subscribe model.

(4) Repeat step1 and step3

Measuring the number of events per client's access is important our design issues. We can measure the number as follows for each model chosen by adaptive scheme:

- Request/reply model: Whenever a server receives client's request, counter of associated client is increased. Then, average number of client's request per event is computed for each client at the end of period and an appropriate model for the client is selected. Server informs client of the selected model.
- Publish/subscribe model: A publisher includes event Id. and the number of subscribers on sending event message. When a subscriber accesses event, it compares current event Id. to the event Id. previously accessed. A subscriber computes an average number of events per access at the end of period and chooses an appropriate model for the subscriber. The subscriber informs publisher of the selected model.

Fig.2 shows that publish/subscribe model is appropriate when the number of client is large and/or the number of event per client's access is small. When we assume that cost of each access (c_{rr}) is equal to 2 in the request/reply model, our adaptive scheme will select publish/subscribe model when its cost per client's access is less than or equal to 2.

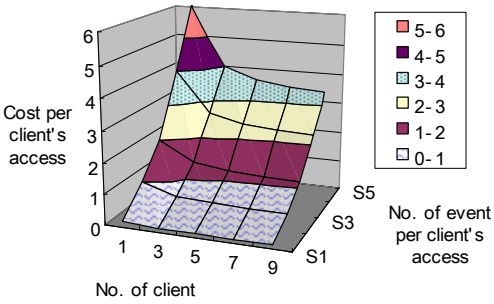

Fig. 2. Cost per client's access of publish/subscribe model (c_{pub}=1 and c_{sub}=1)

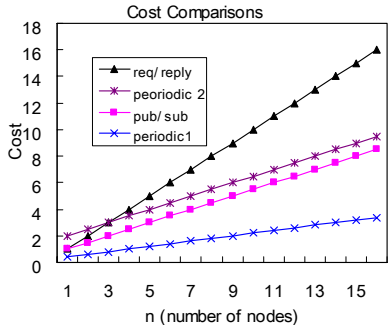

Fig. 3. Communication cost per transaction by varying number of clients (α = 0.5, s(n)=1, c_{ps} = 2, and c_{rr} = 2; $c_{pub}(\alpha,T)= c_{pub}$, $c_{sub}(\alpha,T)= c_{sub}$, and $c_{delay}(\alpha,T)= 0$ for periodic1; $c_{pub}(\alpha,T)= \alpha T c_{pub}$, $c_{sub}(\alpha,T)= \alpha T c_{sub}$, $c_{delay}(\alpha,T)= 2\alpha T c_{dealy}$ for periodic2)

4. Performance Comparisons

In this section, we compare cost between publish/subscribe and request/reply models. Also, we measure performance on our test-bed as shown in Fig.1 to verify correctness of our analysis models.

Table2. Parameters

Param.	Values
α, β	0.5
c_{ps}	2
c_{pub}, c_{sub}	11
$c_{rr:}$	2
$c_{poll}(\alpha, T)$	1 or αT
$c_{delay}(\alpha, T)$	0, T, or αT
s(n)	1/n - 1
t_{ps}	1
t_{proc}	1 or 5
t_{rr}	1
$t_{poll}(\alpha, T)$	1, T, or αT

4.1 Parametric Analysis

In this section, we describe performance comparisons by parametric analysis. We set system parameters as shown in Table 2. Fig.3 shows performance comparisons between publish/subscribe, request/reply, and polling systems. In this experiment, cost is communication cost for each transaction. Since publish/subscriber system disseminates data via server instead of individually for each client, it requires less cost than request/reply system. As the number of client node increases, the cost gap between two systems increases. Periodic polling system saves cost by transferring data once per period when delay cost is negligible. However, cost increases as delay cost increase. Polling system is viable approach for applications where data delay is allowed and delay cost is negligible.

4.2 Experimental Results

Our experiment attempts to get publish/subscribe cost per event for both a spectrum of message sizes and a number of mobile clients in a practical environment. The cost of request/reply event is also experimented for comparisons. The experiment environment consists of NaradaBrokering [4] system where a HHMS *(HandHeld Message Service)* [5] Proxy plugged in, mobile clients, and conventional PC

applications. Mobile applications are written in J2ME MIPD 2.0 [20] with the socket connection supporting. NaradaBrokering is used as a primary publish/subscribe system for a conventional wired distributed system. NaradaBrokering is being developing in Community Grid Laboratory at Indiana University. It is originally designed for a uniform software multicast to support a real-time collaboration linked by publish/subscribe.

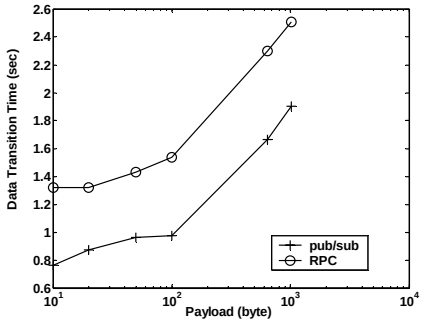

Fig. 4. Delay time by Payload
(Treo 600 Smartphone)

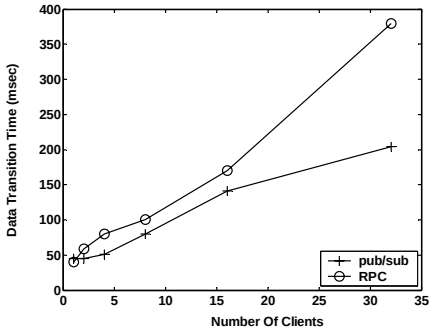

Fig. 5. Delay Time by Number of Clients
(J2ME Wireless Toolkit Emulator)

We made two different measures. One for measuring a practical 'cost per message' of publish/subscribe system and RPC system. The other is measuring 'cost of given number of clients' in wired environment with phone emulator that comes with J2ME Wireless Toolkit. This is a limited configuration, but it is still enough to exemplify the analysis we've made in Section 2. For the message cost experiments, we measured a round trip time (RTT) with a spectrum of message size. A client application (Event Displayer; ED) on Treo 600 mobile device which is connected to Internet through Sprint PCS Vision service just echoes message from the message publisher (Event Source; ES) which runs on wired Linux machine. Thou, we use a mobile device for the experiment. The RPC comparison experiment is set up with direct socket connections between clients and RPC server. The result is shown in Fig.4. The next experiment is done to get a message publishing cost of given number of clients. Client applications run on phone emulators on one desktop and two Laptops. Laptops equipped with Pentium4 processor and minimum 384 MB memory. For the connection, one has wired connection and the other has 802.11b wireless connection. A publisher application publishes a message and when it gets all ACKs, it gets the time stamp. Fig.5 shows the result, which is about accordance with analysis shown in Fig.3. We define the data transition time of publish/subscribe and RPC as RTT / 2 and RTT respectively from the semantics of each messaging scheme.

5. Conclusion

We present cost analysis model for publish/subscribe systems. Based on the cost analysis, we propose adaptive scheme which can dynamically select an appropriate model for each client independently. We can estimate performance and effectively adopt publish/subscribe systems by using our proposed cost model of publish/subscribe

systems and adaptive model. Experimental results (delay time by number of clients) from our test bed are quite similar to our cost analysis models, which verify that our cost analysis model is useful to select proper model and to design adaptive schemes.

References

1. L. Fiege, F. Gartner, O. Kasten, and A. Zeidler, "Supporting Mobility in Content-Based Publish/Subscribe Middleware," Middleware 2003, LNCS 2672, pp. 103 – 122, 2003.
2. M. Caporuscio, A. Carzaniga, and A. Wolf, "Design and Evaluation of a Support Service for Mobile, Wireless Publish/Subscribe Applications," IEEE Transactions on Software Engineering, vol. 29, no. 12, pp. 1059 – 1071, Dec. 2003.
3. U. Farooq, E. Parsons, and S. Majumdar, "Performance of Publish/Subscrive Middleware in Mobile wireless Networks," Proc. of WOSP'04, pp. 278-289, Jan. 2004.
4. Shrideep Pallickara and Geoffrey Fox, "NaradaBrokering: A Middleware Framework and Architecture for Enabling Durable Peer-to-Peer Grids," Proceedings of ACM/IFIP/USENIX International Middleware Conference Middleware, pp 41-61, 2003.
5. Sangyoon Oh, Geoffrey C. Fox, Sunghoon Ko GMSME: An Architecture for Heterogeneous Collaboration with Mobile Devices The Fifth IEEE International Conference on Mobile and Wireless Communications Networks (MWCN 2003) Singapore in Sep. / Oct., 2003.
6. Sangmi Lee, Sunghoon Ko, Geoffrey Fox, Kangseok Kim, Sangyoon Oh A Web Service Approach to Universal Accessibility in Collaboration Services in Proceedings of the 1st International Conference on Web Services ICWS Las Vegas June 2003.
7. P. Eugster, P. Felber, R. Guerraoui, and A. Kermarrec, "The Many Faces of Publish/Subscribe," ACM Computing Surveys, vol. 35, no. 2, Jun. 2003, pp. 114-131.
8. A. Rowstron, A.-M. Kermarrec, M. Castro, and P. Druschel, "SCRIBE: The design of a large-scale event notification infrastructure," in Networked Group Communication, 2001, pp. 30--43. http://citeseer.ist.psu.edu/rowstron01scribe.html
9. Sonic Software Corporation, "Sonic Customer Success: Vodafone", 2004, http://www.sonicsoftware.com/customers/docs/vodafone.pdf
10. Softwired Inc. "Capturing the excitement and intense bidding action of the real Auction house: eBay," http://www.softwired-inc.com/products/success.html
11. Ahmet Uyar, Shrideep Pallickara and Geoffrey Fox, "Audio Video Conferencing in Distributed Brokering Systems," in Proc. International Conf. on Communications in Computing, June 2003
12. G. Muhl, A. Ulbrich, K. Herrmann, and T. Weis, **"**Disseminating Information to Mobile Clients Using Publish-Subscribe," Proc. of the IEEE Internet Computing, Vol.8, No. 3, May/June 2004
13. Sagar Chaki, Pascal Fenkam, Harald Gall, Somesh Jha, Engin Kirda, and Helmut Veith, "Integrating Publish/Subscribe into a Mobile Teamwork Support Platform," Proc. of the 15th International Conference on Software Engineering and Knowledge Engineering 2003
14. Zhexuan Song, Yannis Labrou and Ryusuke Masuoka, "Dynamic Service Discovery and Management in Task Computing," Proceedings of the 1st International Conference on Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous 2004), August 22-25, 2004.
15. A. Carzaniga, D.Rosenblum, and A. Wolf, "Design and evaluation of a wide-area event notification service," in ACM Transactions on Computer Systems, 2001
16. M. Aguilera, R. Strom, D. Sturman, M.Astley, and T. Chandra, "Matching events in a content-based subscription system," Proc. of ACM Symp. on Principles of Distributed Computing, 1999
17. G. Cugola, E. Di Nitto, A. Fuggeta, " The JEDI Event-based infrastructure and its Application to the Development of the OPSS WFMS," IEEE Trans. of Software Engineering, 2001
18. L. Fiegen, G. Muhl, and F. Gartner, "A Modular Approach to Building Event-Based Systems," ACM Symposium on Applied Computing, 2002
19. B. Segall, D. Arnold, J. Boot, M. Henderson and T. Phelps, "Content Based Routing with Elvin4," In Proceedings of AUUG2K, 2000
20. Sun Microsystems, Inc. "Java2 Platform, Micro Edition(J2ME)," http://java.sun.com/j2me/