Analysis for Optimal Degree of Replication

Jai-Hoon Kim
Graduate School of Information and Communications
Ajou University, S. Korea
jaikim@ajou.ac.kr

Geoffrey Fox
Pervasive Technology Institute
Indiana University, USA
gcf@indiana.edu

1. Introduction

Replication technologies are widely used in distributed systems to improve performance in terms of availability and scalability. In distributed systems, there are many cases requiring replication. We can be categorized the reasons of replication as follows:

- Fault-tolerance (availability): When a system component fails, another replicated component takes over the task. System or component is replicated for enhancing availability (ex., dual system, RAID, etc.)
- Load distribution (scalability): By replicating computing or data resources, their tasks can be distributed to avoid bottleneck which causes performance degradations. Mirrored Web site, replicated database, and replicated disk storage are examples of task or data replications to improve scalability.
- Cache (performance): Cache is a kind of replication in which data is replicated on a closer place to access fast (e.g., CPU cache, disk cache, and Web proxy)

Many system components (ex., disk storage, file systems, databases, software tasks, and system itself) can be replicated to obtain the advantages. In fault-prone environments, replication strategies of system components (hardware or software) are popularly adopted to improve system availability [1,5,6,7]. By replicating components, system can continue its jobs even though some component fails (ex., duplex system or TMR). Also, replication has the benefit of scalability by distributing the task load among components (ex., mirrored Web site or clusters) [2,3,4]. However, it requires cost for physical redundancy and maintaining consistency among redundancy.

Many replication methods [1,2,3,4,5,7] were proposed to increase availability and scalability. Some researches [2,3,4] were performed to obtain proper degree or location of replication to reduce total access cost by considering trade-off between cost for resource access and cost for component replication (e.g., cost of component or maintaining consistency among components). However, few researches were investigated to obtain proper degree of replication to minimize total costs in error-prone environments. We need to consider many aspects to decide replications and degree of it; cost of physical component, re-do (or recovery) overhead on a failure, failure rate, and access (read and write) rate and cost.

We analyze cost of replication in various aspects, which help us to choose proper degree of replication in error-prone systems. Fig.1 shows the outline of our scheme for choosing degree of replication. Our proposed cost model and algorithm to choose proper degree of replication is simple. However, it would help us to decide whether or not to replicate and, if replicate, degree of replication in our system design.

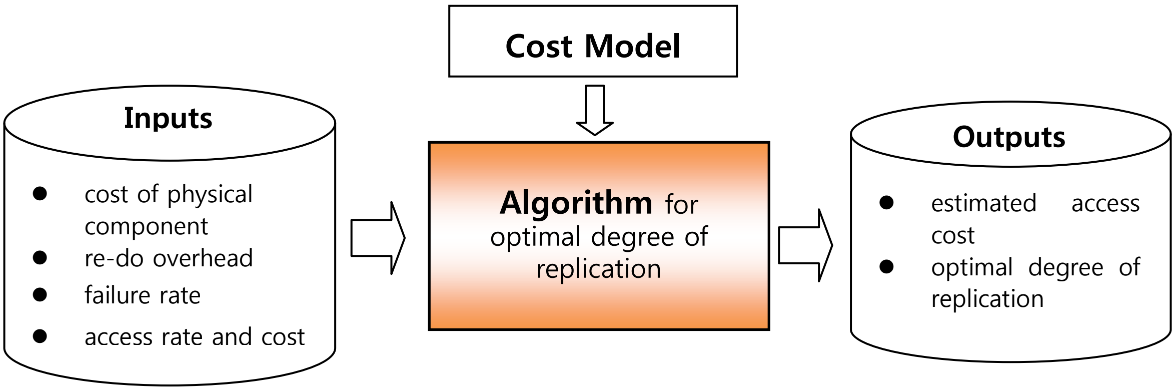
Cost Model
Inputs
cost of physical component
re-do overhead
failure rate
access rate and cost
Algorithm for optimal degree of replication
Outputs
estimated access cost
optimal degree of replication

Fig.1 Algorithm for choosing degree of replication

2. Cost Analysis for Degree of Replication

The total cost of replication is:

$$c_s(x) = c_p(x) + c_a(x) + c_r(x),$$

where $c_s(x)$ is total cost function, x is degree of replication, $c_p(x)$ is cost of physical replication, $c_a(x)$ is cost of resource access, and $c_r(x)$ is cost of re-do overhead due to all x replications' failures.

Now, we analyze the three costs; $c_p(x)$, $c_a(x)$, and $c_r(x)$.

(1) We assume that cost of physical replication is proportional to the number of replicated component:

$c_p(x) = xc_{phy}$, where c_{phy} is cost of physical component per time unit.

(2) We assume that cost of access consists of read cost and write cost (consistency overhead among replications). In general, read cost adversely proportional to the number of replication while consistency overhead is proportional to:

$$c_a(x) = \lambda_{read}\frac{c_{read}}{x} + \lambda_{write}xc_{consist},$$

where λ_{read}, λ_{write}, $\frac{c_{read}}{x}$, and $xc_{consist}$ denote read rate, write rate, read overhead, and consistency overhead, respectively.

(3) Re-do overhead is required on all x replicated components' failures:

$$c_r(x) = (1 - \varepsilon^{-\lambda T})^x c_{re-do},$$

where λ, T, and c_{re-do} are failure rate, transaction time, and re-do overhead, respectively. Thus, $c_s(x) = c_p(x) + c_a(x) + c_r(x)$

$$= xc_{phy} + \lambda_{read}\frac{c_{read}}{x} + \lambda_{write}xc_{consist} + (1 - \varepsilon^{-\lambda T})^x c_{re-do}$$

We analyze replication cost as the function of replication degree. Our analysis is simple but it is useful to decide proper replication degree in the various applications in which condition is different to each other.

In general, failure rate is small. Thus, we can approximate $c_s(x)$ as follows:

$$c_s(x) = xc_{phy} + \lambda_{read}\frac{c_{read}}{x} + \lambda_{write}xc_{consist} + \left(1 - \varepsilon^{-\lambda T}\right)^x c_{re-do}$$

$$\cong xc_{phy} + \lambda_{read}\frac{c_{read}}{x} + \lambda_{write}xc_{consist} + (\lambda T)^x c_{re-do} \qquad (1)$$

If we assume that access cost (c_{access}, read and write cost) is constant (c_a) or ignored ($c_a = 0$), then;

$$c_s(x) = xc_{phy} + c_a + (\lambda T)^x c_{re-do} \qquad (2)$$

➔ $\frac{\partial c_s}{\partial x} = c_{phy} + (\lambda T)^x \ln(\lambda T) c_{re-do}$

$$\text{As } \frac{\partial c_s}{\partial x} = 0 \text{ when } x = ln_{\lambda T}\left(-\frac{c_{phy}}{\ln(\lambda T)c_{re-do}}\right),$$

$$c_s(x) \text{ is minimum when } x = ln_{\lambda T}\left(-\frac{c_{phy}}{\ln(\lambda T)c_{re-do}}\right)$$

Now, we obtain the optimal solution of x (degree of replication). $c_s(x)$, total cost function of x (degree of replication), has the minimum value,

$$ln_{\lambda T}\left(-\frac{c_{phy}}{\ln(\lambda T)c_{re-do}}\right)c_{phy} + c_a - \frac{c_{phy}}{\ln(\lambda T)c_{re-do}}c_{re-do},$$

when x = $ln_{\lambda T}\left(-\frac{c_{phy}}{\ln(\lambda T)c_{re-do}}\right)$.

Algorithm to obtain the optimal degree of replication (x_{opt}) is as follows:

Algorithm I (when c_a is variable)

```
$for\ (x = 1;\ x \le x_{max};\ ++x)$
    compute $c_s(x) = xc_{phy} + \lambda_{read}\frac{c_{read}}{x} + \lambda_{write}xc_{consist} + (\lambda T)^x c_{re-do}$
compute natual number $x_{opt}, such\ that\ c_s(x_{opt}) = \min_{x>0} c_s(x)$
```

Algorithm II (when c_a is constant)

```
compute real number $x = ln_{\lambda T}\left(-\frac{c_{phy}}{\ln(\lambda T)c_{re-do}}\right)$
if ( $x \le 1$)  $x_{opt} = 1$
else if ($x$  is natural number)  $x_{opt} = x$
else compute  $x_{opt},\ such\ that\ c_s(x_{opt}) = \min\{c_s(\text{floor(x)}), c_s(\text{ceiling(x)})\}$
```

3. Performance Measures

We measure the performance of degree of replication by varying system parameters (cost of physical component, failure rate, and re-do overhead).

Fig.2 shows the performance for different cost of physical component. We assume the system parameters as follows: $\lambda_{read} = 0.3, \lambda_{write} = 0.2, c_{read} = 1, c_{consist} = 2, \lambda = 0.01, T = 20,$ and $c_{re-do} =$ 50. In this analysis, costs are minimal on x (degree of replication) about 2.5, 1.75, and 1.25 when c_{phy} (cost of physical component) are 1, 5, and 10, respectively. However, as degree of replication is integer, costs are minimal on x being 3, 2, and 1 when c_{phy} are 1, 5, and 10, respectively. We find that optimal degree of replication is different for each cost of $c_{phy.}$ Our analysis is simple. But, we can choose a proper degree of replication for each cost of physical component by using the results of our cost analysis. In general, high cost of physical component limits the replication.

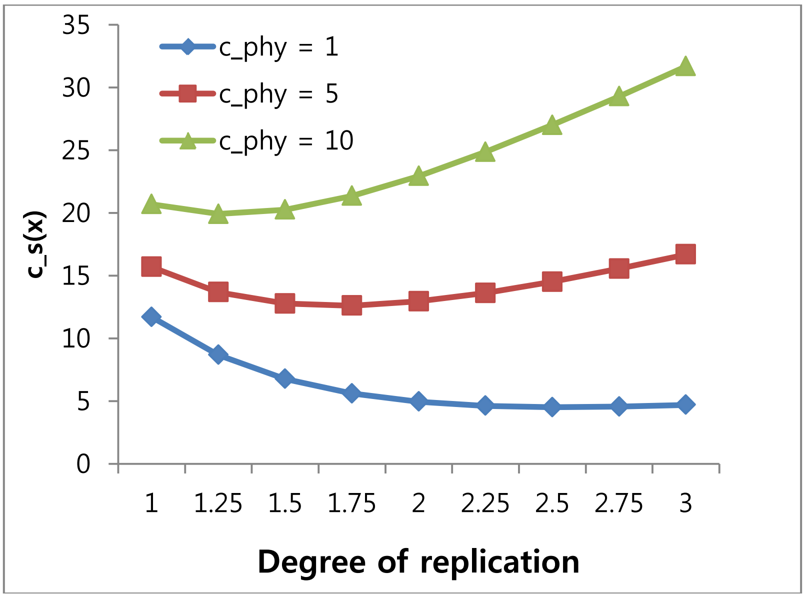

Fig. 2 Performance comparison by varying cost of physical component

Similarly, we measure the performance for different re-do overheads. When a system fails, we need to re-do. Re-do overhead is different for each task. If PC fails and is rebooted while we type a report, the re-do overhead is re-typing the report. On the other hand, if a computer embedded on a space shuttle fails, which results in out of control and may lose the space shuttle, the re-do overhead (re-launching the space shuttle) is very huge. Fig.3 shows the performance for different cost of re-do overheads. In this experiment, we assume the system parameters as follows: $c_{phy} = 5$, $\lambda_{read} = 0.3, \lambda_{write} = 0.2, c_{read} = 1$, $c_{consist} = 2, \lambda = 0.01$, and $T = 20$. In this analysis, we can find that as re-do overhead increases, optimal degree of replication increases. By using our cost analysis, we can choose proper degree of replication according to re-do overhead.

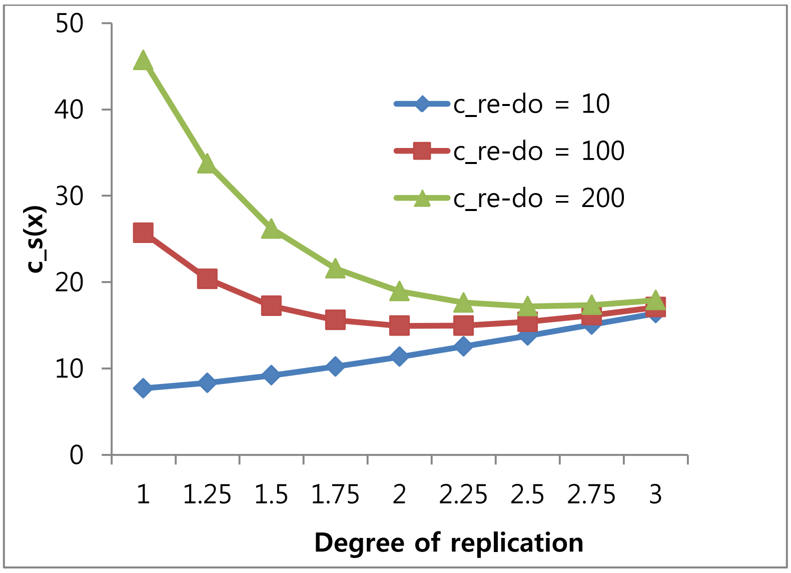

Fig.3 Performance comparison by varying cost of re-do

We also measure the performance for different failure rate. When a system fails more frequently, we need higher degree of replication to increase availability and reduce the re-do probability. Fig.4 shows the performance on different failure rates. We set up the system parameters as follows: $c_{phy} = 1$, $\lambda_{read} = 0.3, \lambda_{write} = 0.2, c_{read} = 1$, $c_{consist} = 2$, $T = 20$ and $c_{re-do} = 30$. By using our cost analysis, we can choose proper degree of replication according to the failure rate.

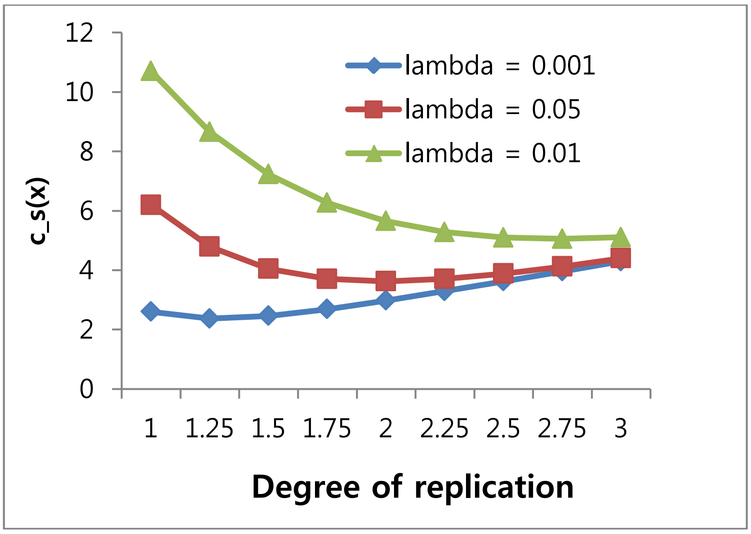

Fig.4 Performance comparison by varying failure rate

We measure the optimal degree of replication according to system variables. (We assume that access cost (c_a, read and write cost) is constant or ignored ($c_a = 0$) as we obtain Eq. (2).) Fig.5 shows the optimal degree by varying c_{phy} (1~4) and c_{re-do}(1~1000). We set up other system parameters as follows: $\lambda_{read} = 0.3, \lambda_{write} = 0.2, c_{read} = 1$, $c_{consist} = 2$, $\lambda = 0.001$, and $T = 20$. As values c_{phy} decreases and c_{re-do} increases, the optimal degree of replication increases. As Fig.5

shows a theoretical optimal value of degree of replication, we need to choose two nearest natural numbers and obtain the value which has smaller value of $\mathrm{c_s(x)}$ as described in Algorithm Ⅱ.

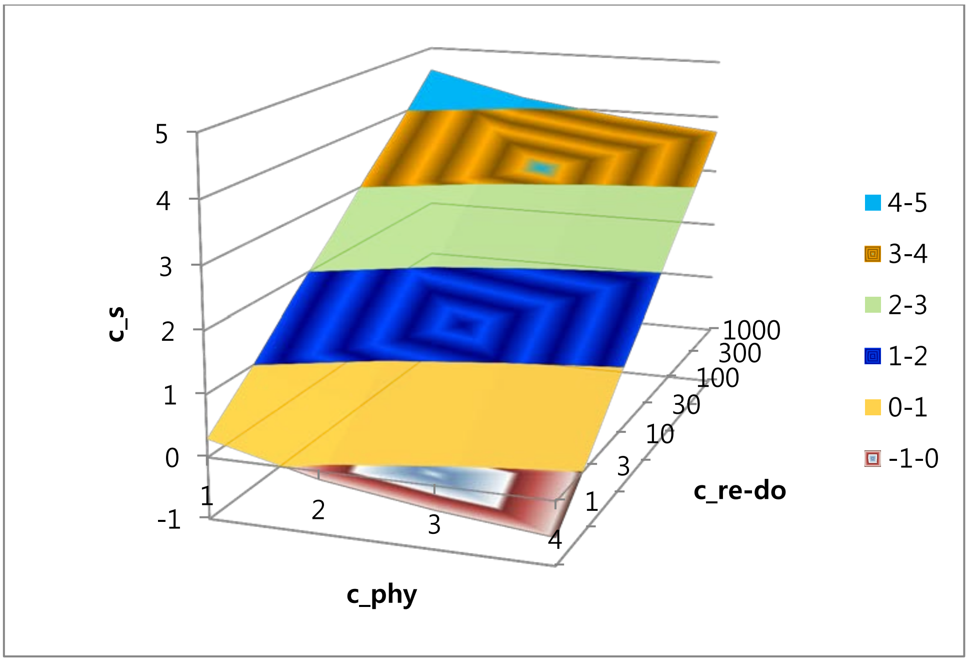

Fig.5 Optimal degree of replication by varying c_{phy} (1~4) and c_{re-do}(1~1000).

We also measure the optimal degree of replication according to other system variables. Fig.6 shows the optimal degree by varying $\mathrm{c_{phy}}$ (1~4) and λ (0.0001~0.002). We set up other system parameters as follows: $\lambda_{\mathrm{read}} = 0.3$, $\lambda_{\mathrm{write}} = 0.2$, $\mathrm{c_{read}} = 1$, $\mathrm{c_{consist}} = 2$, $\lambda = 0.001$, $\mathrm{T} = 200$, and $\mathrm{c_{re-do}} = 100$. As values $\mathrm{c_{phy}}$ decreases and λ increases, the optimal degree of replication increases. As described in Algorithm Ⅱ, we need to choose two nearest natural numbers and obtain the value which has smaller value of $\mathrm{c_s(x)}$.

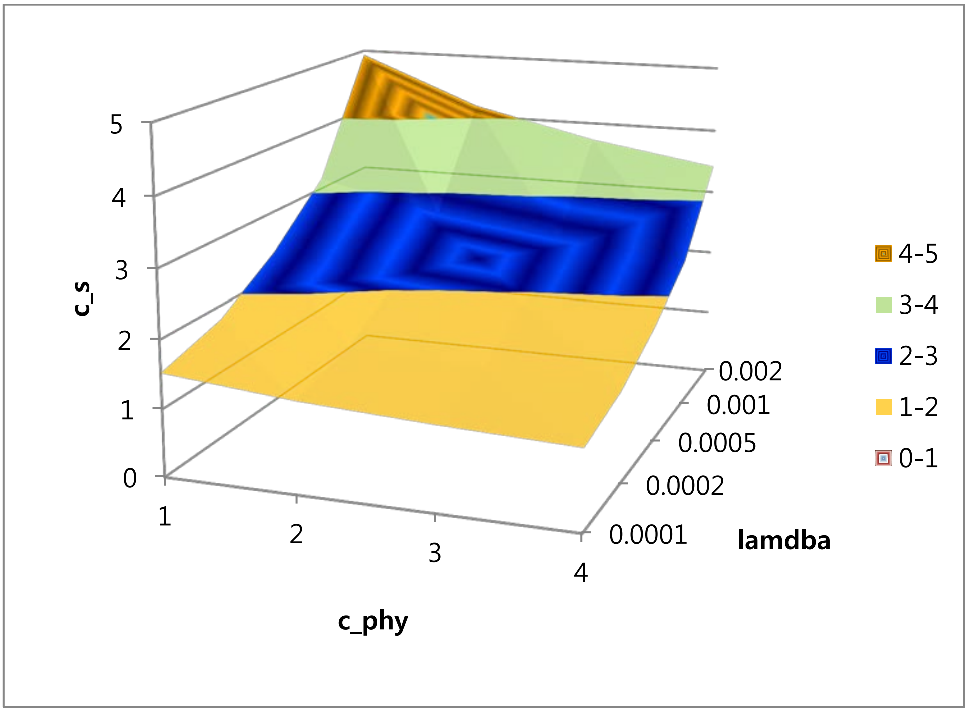

Fig.6 Optimal degree of replication by varying c_{phy} (1~4) and λ (0.0001~0.002).

4. Conclusion

Replication is widely used technology to increase availability and scalability. However, it requires cost for physical redundancy and maintaining consistency among replications. Many researches investigated replication methods to improve availability and scalability while reduce replication overhead. However, researches to obtain the proper degree of replication are seldom performed. We analyzed the cost of replication to choose the proper degree of replication according to various system parameters: cost of physical component, re-do overhead, and failure rate. Our analysis is simple. But, we can select the optimal degree of replication on a system design considering the various aspects. In general, low cost of physical component, high re-do overhead, and high failure rate encourage high degree of replication.

References

[1] Sung-Hwa Lim, Byoung-Hoon Lee, and Jai-Hoon Kim, "Diversity and fault avoidance for dependable replication systems," Information Processing Letters, vol. 108, pp. 33-37, 2008.

[2] Sung-Hwa Lim and Jai-Hoon Kim, "Optimal Server Replication Schemes to Reduce Location Management Cost in Cellular Network," *IEICE Trans. on Communications*, vol. E89-B, no. 10, pp. 2841-2849, 2006.

[3] Xiaohua Jia, Deying Li, Hongwei Du, and J. Cao, "On optimal replication of data object at hierarchical and transparent Web proxies," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, Issue 8, pp. 673-685, 2005.

[4] Avraham Leff, Joel Wolf, and Philip Yu, "Replication algorithms in a remote caching architecture," *IEEE Transactions on Parallel and Distributed Systems*, vol. 4, no. 11, pp. 1185-1204, 1993.

[5] Chetan Shankar, Anand Ranganathan and Roy Campbell, "Towards Fault Tolerant Pervasive Computing," *IEEE Technology and Society,* 24 (2005), 38-44.

[6] L. Strigini, "Fault Tolerance Against Design Faults," Hassan Diab and Albert Zomaya, (Eds.), *Dependable Computing Systems: Paradigms, Performance Issues, and Applications*, J. Wiley & Sons, 2005, pp. 213-241.

[7] Pallickara, S., H. Bulut, and G. C. Fox, "Fault-Tolerant Reliable Delivery of Messages in Distributed Publish/Subscribe Systems," *Fourth International Conference on Autonomic Computing (ICAC'07),* Jun 2007.