

Reliability analysis of web server cluster systems based on proportional hazards model

Hou Chunyan¹ Wang Jinsong¹ Chen Chen²

(¹School of Computer Science and Engineering, Tianjin University of Technology, Tianjin 300384, China)

(²College of Computer and Control Engineering, Nankai University, Tianjin 300071, China)

Abstract: An approach for web server cluster (WSC) reliability and degradation process analysis is proposed. The reliability process is modeled as a non-homogeneous Markov process (NHMH) composed of several non-homogeneous Poisson processes (NHPPs). The arrival rate of each NHPP corresponds to the system software failure rate which is expressed using Cox's proportional hazards model (PHM) in terms of the cumulative and instantaneous load of the software. The cumulative load refers to software cumulative execution time, and the instantaneous load denotes the rate that the users' requests arrive at a server. The result of reliability analysis is a time-varying reliability and degradation process over the WSC lifetime. Finally, the evaluation experiment shows the effectiveness of the proposed approach.

Key words: web server cluster; load-sharing; proportional hazards model; reliability; software aging

DOI: 10.3969/j.issn.1003-7985.2018.02.007

A web server cluster (WSC) is a kind of k -out-of- n load-sharing system (LSS), in which at least k -out-of- n components must work for the successful operation of the system. The load sharing mechanism introduces dependency between the time to failure among the components, making modeling and inference of such systems different from simpler redundant systems^[1].

In the past few decades, the computing capacity of web server clusters (WSCs) has increased dramatically. However, a linear increase of cluster size results in an exponential failure rate. System software and applications running on cluster systems are becoming more and more complex, which makes them prone to bugs and other software failures^[2]. After a WSC is put into operation, its aging and degradation can make the software failure rate even higher over time. It is preferable that one can manage the system degradation process to gracefully handle failures before potential outages occur. Degradation

measurements on WSCs provide information about their reliability.

Much research has been done on accelerated life testing (ALT) models for reliability and the degradation analysis of LSSs^[3-4]. ALT models play an important role in determining the relationships between load and component lives or failure rates. Many empirical studies of mechanical systems^[5] and computer systems^[6] have proved that the workload strongly affects the component failure rate. ALT is the technology that utilizes the failure time data of products under higher stresses to extrapolate the lifetime and reliability of the products under normal operating conditions. ALT models have significant effects on the estimation accuracy of the lifetime and reliability of products. The main problem with the existing ALT models is that they are only applicable for hardware LSSs. System software deployed in a WSC executes intermittently so that the regular chronological time scale is not applicable for modeling a WSC.

The PHM was first proposed by Cox^[7] and has been widely applied to relate the failure probability to both historical service lifetime and condition monitoring variables^[8]. For this time-dependent model, failure prediction is treated as estimating the remaining lifetime for a system with regard to a specific hazard level under the current conditions. Mohammad et al.^[9] provided a closed-form analytical solution for the reliability of PHM load-sharing k -out-of- n systems with identical hardware components, where all surviving components share the load equally. It also considers system failures caused by imperfect load distribution. This approach did not explicitly model how condition variables affect the component failure rate.

1 System Model

The definition of system models meets the following assumptions: 1) There are n i. i. d servers in a WSC, where software is deployed and runs, and the system functions successfully if and only if it can respond to user requests promptly; 2) User requests to a WSC meet a stationary stochastic process with a constant arrival rate, and are distributed to all active components equally; 3) No repair or maintenance is considered; 4) The components are either operational or failures. The components, once failed, are removed from the system immediately.

Received 2017-11-23, **Revised** 2018-02-03.

Biography: Hou Chunyan(1980—), female, doctor, lecturer, chunyan-hou@163.com.

Foundation items: The National Natural Science Foundation of China (No.61402333, 61402242), the National Science Foundation of Tianjin (No.15JCQNJC00400).

Citation: Hou Chunyan, Wang Jinsong, Chen Chen. Reliability analysis of web server cluster systems based on proportional hazards model[J]. Journal of Southeast University (English Edition), 2018, 34(2): 187 – 190. DOI: 10.3969/j.issn.1003-7985.2018.02.007.

Software applications executing continuously for a long-running time show a phenomenon of software aging. This phenomenon is the result of the exhaustion of system resources, memory leaks and the accumulation of internal error conditions. The aging rate is dependent on software workload. We employ Cox's PHM to model the relationship between software failure rate and workload including cumulative and instantaneous loads. We introduce cumulative execution time $X(t)$ from start time to time t to describe cumulative load, which reflects software age. Since software performs intermittently, $X(t) < t$. Instantaneous load denotes the rate that users' requests arrive at a WSC. Therefore, the software failure rate is expressed as

$$h(t) = b \exp(\alpha X(t) + \beta Y(t))$$

where b is the constant baseline failure rate, the value of which depends on how well software is developed and tested; α and β are the regression coefficients estimated by observed data; and $Y(t)$ is the rate that users' requests arrive at a server. According to the software reliability theory, software reliability significantly depends on the operational profile^[10]. The WSC operational profile is defined as follows.

Definition 1 (profile) A profile models how a WSC is visited, and it is defined as the tuple $\langle \text{double } \mu, \text{int } \omega \rangle$, where μ is the rate that users' requests arrive; and ω is the average amount of workload included in a request.

When a WSC is put into operation at time zero, n servers are working, and they are equally sharing the total requests arriving at the system. We define system states as the number of failing servers, that is 0, 1, 2, ..., $(n - k)$, $(n - k + 1)$. A WSC fails when the number of failures exceeds $(n - k)$. From an overall point of view, the WSC failure process can be represented by the pure birth Markov process without consideration of repair or maintenance. Since system failure rates are not constant but vary with time, the failure process is a non-homogeneous Markov process (NHMP). On the other hand, surviving components process different amounts of workload at different states, which leads to non-continuous system failure rates with consecutive system states. Therefore, NHMP can be further divided into $(n - k + 1)$ non-continuous NHPPs corresponding to NHMP states. Only one failure occurs during each NHMP.

2 Reliability Analysis

A NHMP state stands for the number of failing components. At state s ($0 \leq s \leq n - k + 1$), $(n - s)$ working components equally share the total requests. The rate that users' requests arrive at a surviving server is $Y_s(t) = \mu / (n - s)$. According to Definition 1, the average response time to a request is ω / γ , where γ is the rate that a server processes user requests. The cumulative execution time of a surviving component at state s is

$$\tau_s(t) = \frac{\mu \omega}{(n - s) \gamma} \left(t - \sum_{i=0}^{s-1} \Delta t_i \right)$$

where Δt_i is the expected time at state i . The total cumulative execution time of a component is

$$X_s(t) = \frac{\mu \omega}{\gamma (n - s)} \left[t - \sum_{i=0}^{s-1} \frac{(s - i) \Delta t_i}{n - i} \right]$$

Given $X_s(t)$ and $Y_s(t)$, the component failure rate $h_s(t)$ can be obtained, and the NHPP arrival rate is $\lambda_s(t) = (n - s) h_s(t)$. Only one failure occurs during the NHPP at state s . Thus, we have $\int_{\sum_{i=0}^{s-1} \Delta t_i}^{\sum_{i=0}^s \Delta t_i} \lambda_s(t) dt = 1$, from which the expected time can be solved as

$$\Delta t_s = \frac{1}{\phi_s} \ln \left[\phi_s / \left((n - s) b \exp \left(\beta \mu / (n - s) - \phi_s \sum_{i=0}^{s-1} (s - i) \Delta t_i / (n - i) \right) \right) + \exp \left(\phi_s \sum_{j=0}^{s-1} \Delta t_j \right) \right] - \sum_{k=0}^{s-1} \Delta t_k$$

where $\phi_s = \alpha \mu \omega / ((n - s) \gamma)$.

When the number of surviving components is less than k , a WSC will be unable to promptly respond to users' requests and the system fails. The minimal value of k is $\lceil \mu \omega / \gamma \rceil$. The WSC reliability process is a NHMP composed of $(n - k + 1)$ working states and one failure state. The system failure rate can be expressed as $\lambda(t) = f(t) / R(t)$, where $R(t)$ is the system reliability; and $f(t)$ is the failure probability density function given by $f(t) = -dR(t)/dt$. Therefore, the relationship between the failure rate and reliability can be solved as $\lambda(t) dt = -d \ln R(t)$. According to the failure rates at NHMP working states analyzed above, their reliability process is

$$R(t) = \exp \left[- \frac{(n - s) b}{\phi_s} \cdot \exp \left(\phi_s \left(t - \sum_{i=0}^{s-1} (s - i) \frac{\Delta t_i}{n - i} \right) + \frac{\beta \mu}{n - s} \right) \right] \quad \sum_{i=0}^{s-1} \Delta t_i \leq t \leq \sum_{i=0}^s \Delta t_i$$

3 Illustrative Example

Fig. 1 illustrates a high-level view on the business reporting system (BRS)^[10], which generates management reports from business data collected in a database. The bottleneck in BRS reliability is up to a load-sharing WSC, named GWSC, composed of six servers. Assume that the initial failure rate of the core graphic engine is 1×10^{-5} failures per day. User requests arrive at an average rate of 100 request/s. The time to respond to a request is about 30 ms.

GWSC uses a k -out-of- n structure, where $n = 6$ and k can be solved as 3. Thus, the GWSC reliability process includes four working states from 0 to 3. First, we calcu-

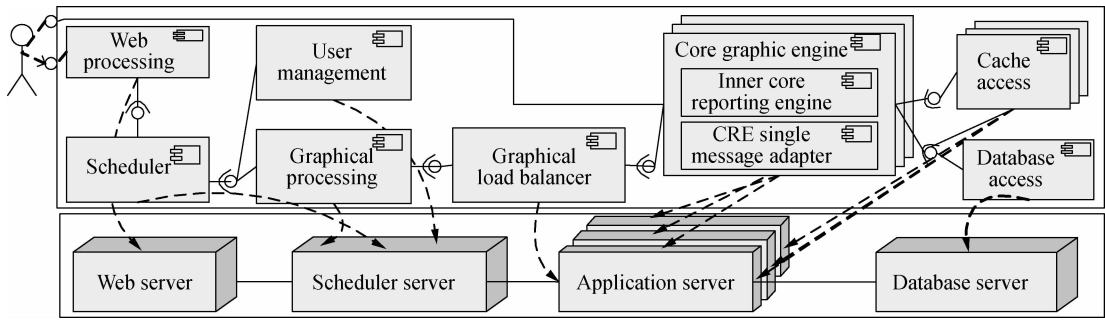


Fig. 1 An overview of the business reporting system

late the expected time at normal states, where two coefficients α and β are assumed to be 0.1 and 1×10^{-6} , respectively. The lifetimes of the four working states are 105.809 7, 12.159 7, 5.629 6 and 2.608 5 d, respectively, at the state from 0 to 3, from which the GWSC reliability and failure rate process can be obtained, as shown in Fig. 2. It can be seen that GWSC reliability gradually decreases but the failure rate increases correspondingly over time until GWSC fails. The complete lifetime is about 126 d, after which system maintainers may need to restart servers or upgrade system software in order to allow the system enter a state of normal operation again.

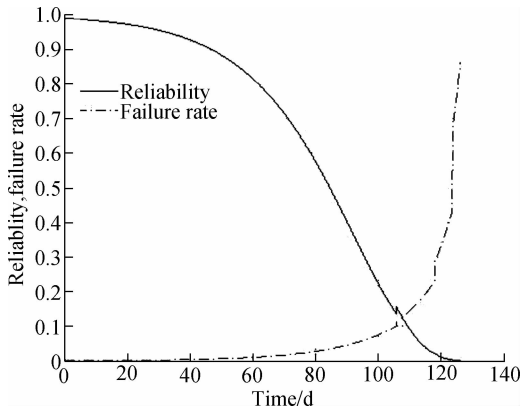


Fig. 2 GWSC reliability and failure rate process

The reliability processes are shown in Fig. 3 when GWSC is configured with 4, 6, 8, or 10 servers. The results illustrate that more servers improve system reliability and prolong its lifetime. The corresponding reliability processes are also shown in Fig. 4 when user requests arrive at various rates. It can be seen that the faster the user requests arrive, the lower the system reliability. With the WSC reliability process as a reference, software designers can adjust the system configuration to allow system reliability and lifetime to meet customer requirements.

4 Conclusion

In this paper, we propose an approach to model and analyze WSC reliability. The result is a time-dependent software reliability process. Using the model and method

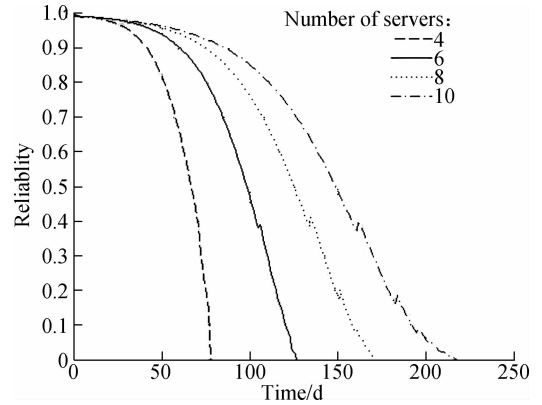


Fig. 3 GWSC reliability process with different number of servers

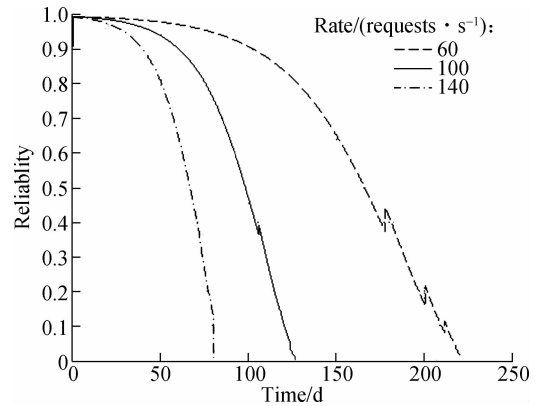


Fig. 4 GWSC reliability process under different operational profiles

proposed in this paper, it is simple to analyze the reliability degradation of multi-state software LSSs, which is caused by the failures of load-sharing components. The reliability analysis approach is very meaningful for supporting WSC management and design decisions.

References

- [1] Ye Z, Revie M, Walls L. A load sharing system reliability model with managed component degradation[J]. *IEEE Transactions on Reliability*, 2014, **63**(3): 721 – 730.
- [2] Vaidyanathan K, Harper R E, Hunter S W, et al. Analysis and implementation of software rejuvenation in cluster systems[C]//*ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*. Massachu-

- setts, USA, 2001: 62 – 71.
- [3] Park C. Parameter estimation for the reliability of load-sharing systems[J]. *IIE Transactions*, 2010, **42**(10): 753 – 765. DOI: 10.1080/07408171003670991.
- [4] Park C. Parameter estimation from load-sharing system data using the expectation-maximization algorithm[J]. *IIE Transactions*, 2013, **45**(2): 147 – 163. DOI: 10.1080/0740817x.2012.669878.
- [5] Liu H M. Reliability of a load-sharing k -out-of- n : G system: Non-iid components with arbitrary distributions[J]. *IEEE Transactions on Reliability*, 1998, **47**(3): 279 – 284. DOI: 10.1109/24.740502.
- [6] Huang L, Xu Q. Lifetime reliability for load-sharing redundant systems with arbitrary failure distributions[J]. *IEEE Transactions on Reliability*, 2010, **59**(2): 319 – 330. DOI: 10.1109/tr.2010.2048679.
- [7] Cox D R. Regression models and life tables (with discussion)[J]. *Journal of the Royal Statistical Society, Series B*, 1972, **34**(2): 187 – 220.
- [8] Zhang Q, Hua C, Xu G H. A mixture Weibull proportional hazard model for mechanical system failure prediction utilising lifetime and monitoring data[J]. *Mechanical Systems and Signal Processing*, 2014, **43**(1/2): 103 – 112. DOI: 10.1016/j.ymssp.2013.10.013.
- [9] Mohammad R, Kalam A, Amari S V. Reliability of load-sharing systems subject to proportional hazards model [C]//*Reliability and Maintainability Symposium(RAMS)*. Orlando, FL, USA, 2013: 1 – 5.
- [10] Hou C Y, Chen C, Wang J S, et al. A scenario-based reliability analysis approach for component-based software [J]. *IEICE Transactions on Information and Systems*, 2015, **E98-D**(3): 617 – 626. DOI: 10.1587/transinf.2014edp7241.

基于比例风险模型的 Web 服务器集群系统可靠性分析

侯春燕¹ 王劲松¹ 陈 晨²

(¹ 天津理工大学计算机科学与工程学院, 天津 300384)

(² 南开大学计算机与控制工程学院, 天津 300071)

摘要:提出一个用于 Web 服务器集群系统(WSC)可靠性和降级过程分析的方法. 可靠性过程建模为一个非齐次马尔可夫过程(NHMH), 该过程由若干个非齐次泊松过程(NHPPs)组成. 每个 NHPP 到达速率对应于系统软件失效率. 用 Cox 比例风险模型(PHM)建模软件失效率, 模型中同时考虑软件累积和瞬时工作负载. 软件累积工作负载表示软件累积执行时间, 而瞬时工作负载表示用户请求到达速率. 可靠性分析结果是一个在 WSC 生命周期内随时间变化的可靠性和降级过程描述. 最后, 评估实验证明了方法的有效性.

关键词:web 服务器集群; 负载共享; 比例风险模型; 可靠性; 软件老化

中图分类号:TP399