Research on Software Runtime Credibility Based on Cluster Analysis

Weiqing Wan, Qingyan Zeng and Zhicheng Wen*
Jiangxi University of Engineering, Xinyu, Jiangxi Province, 338000, China
*Corresponding author e-mail: zcwen@mail.shu.edu.cn

Abstract. Monitoring and analyzing the reliability of software runtime behavior in an open, dynamic, and complex network environment is very important for modern distributed software. The characteristics of the extrinsic representation in the execution of distributed software are collected, and the data is executed based on the history of the software, and the wide square cube cluster is constructed, and the reliability is analyzed, and the fault tolerance processing and the fault diagnosis of the succeeding system are analyzed. Performance optimization is facilitated. Simulation experiments show that the proposed method has a unique advantage in comparison with other methods in reliability analysis of software behavior.

1. Introduction

In recent years, software credibility has become a research hotspot, including software identity credibility and its behavior credibility. Software operations can be trusted in the main direction of software reliability, and the actions and results appear to be able to be able to control mainly the actions in accordance with people's needs It does not provide fraud. Therefore, the reliability of software is mainly studied for the reliability of software behavior[1,2].

We monitor software runtime behavior, collect external representation information, and provide basic data such as software fault diagnosis, prediction and reliability evaluation. Monitoring or data collection exists for a more mature monitoring mechanism, tools, monitoring system, etc., in order to collect information such as runtime behavior and state of software[3]. Each software behavior has an expression feature outside the CPU, such as CPU occupancy rate, memory space usage size, number of threads, and network throughput. It is necessary to study the relationship between the intention, the scene, the act and the effect of the action in the open and dynamic environment.

Based on the extrinsic representation of execution of distributed software, we systematically collect relevant data and construct a large cube cluster model. By using this model, we analyze the reliability of the execution of software actions by monitoring relevant data in the context environment, and analyze the structure of the software system. In this paper, the method of cluster analysis is adopted, and the statistical law of the externals expression related to the software operation is analyzed, and the intention of the software interactive entity element is further inferred. In future, it becomes the basis for the positive prediction of the software behaviour system.

2. Software credibility

In general, the operational reliability of software entities can be classified into reliability, security, availability, and ageing. Therefore, when the software entity is running, the CPU usage, memory coverage, open line count, network interface flow, file system size, proxy, I / O size, log file size The
network connection situation, the transaction situation, the expiration time, the failure time, the
difference of the operation, the opponent's key port, the opponent's hole attack, the response time, the
interaction time, etc. are long. If the software entity's behavior is unreliable, efficient, or fraudulent,
the software entity's CPU usage is high, the memory occupancy is high, the number of threads is large
It may attack or scan the server[4,5].

Therefore, the behavior of the software entity can construct a reliable three-layer structure. The first
layer is the root node "reliability", and the second layer is reliability, safety, availability and aging.
The third layer is further detailed and represented as CPU usage, occupation memory, open path and
network interface traffic under reliability. The size of the file system, the configuration of the proxy,
the I/O size, etc., xi shows the above features, as shown in Fig 1.

![Fig. 2 A reliable hierarchical model of software behavior](image)

As described above, the reliability of the software may consist of reliability, safety, availability, and
aging and each component may also be subdivided to represent feature xi. Each observation xi index
usually has continuous values, and discrete processing is necessary to facilitate cluster analysis.

**Lemma 1** Assumes continuous random variable X~N(µ, σ²), then Z=(X-µ)/σ~N (0, 1). Here, µ
represents the mathematical expectation of the random variable x, and σ² represents the dispersion.

Probabilistic knowledge can divide random variable x observations into five intersecting intervals
Si: (-∞, µ-3σ)∪( µ+3σ, +∞), (µ-3σ, µ-2.5σ)∪(µ+2.5σ, µ+3σ), (µ-2.5σ, µ-2σ)∪(µ+2σ, µ+2.5σ),
(µ-2σ, µ-σ)∪(µ+σ, µ+2σ) and [µ-σ, µ+σ].

Thus, the observed random variable X can be discretized into five interval SSi, and their
(corresponding probability is PSi.}

3. Constructing general cube

The generalized cube is composed of m-dimensional, and the reliability of the cluster analysis
software and the evaluation of the externality characteristic constitute different dimensions, and
individual discrete values constitute different abstract levels. The current software behavior is
reflected from different angles for data analysis of each dimension and hierarchy.

**Definition 1** Dataset specifies the set of dimensions A={A1, A2, ..., Am | m>0} and level set
H={H1, H2, ..., H_L | L>0} (Hi represents the level set of the dimension Ai, and H_j
represents the jth
level of the dimension A_i). The quantity in H is {H1, H2, ..., H_L}, combine one level. Such
 combinations are all. The {H_{j1}, H_{j2}, ..., H_{jm}}performs the packet aggregation operation in the data
set s, and the set of obtained cube cells constitutes the data cube (A, H).

**Definition 2** The cube cell may be expressed as a binary (A_H, M). where: 1) A_H={H_{j1}, H_{j2}, ..., H_{jm}}, H_{ji}∈H_i is the concrete level j1 selected by the dimension Ai; 2)m is the number of the cells to
be subjected to the polymerization operation after being discreted through the "3σ rule" for the ah
data set s and falling into the cell. Cubes are cubic: (A, H)={ (A_H, M)}.

For example, the CPU utilization rate, B is the memory size, and C is the network flow rate, and
the data set s is a sample containing the three observation indices, and after 5 discrepancies is
performed in accordance with the "3σ rule", one three-dimensional data cube is shown in Fig. 2. After
the set operation, each cube has a corresponding value ( (a_i, b_j, c_k), M_{ijk}), where Mijk represents the
number of cubic grids (a_i, b_j, c_k) after the data set s has been discreted.
4. Cluster Analyses

The reliability of software consists of reliability, safety, availability and ageing, and each node can build a generalized cube. On the set of externals characterization $x_1, x_2, \ldots, x_m$ is the original data $s$ of $n$ groups (n data in total, n is large enough) in the same time zone, and if each component $x_i$ follows the "3σ rule", it is possible to construct the discredited data of $n$ groups. When m dimensional cubes with five hierarchies reach a $5^m$ cube, the data $s$ of $n$ groups (only 0 to 4 of each component) is distributed to the cube of $5^m$, and the sum of the numbers of data in all cubes is $n$.

$\sum_{j_1=0}^{4} \sum_{j_2=0}^{4} \cdots \sum_{j_m=0}^{4} M_{j_1j_2\ldots jm} = n$, here, $M_{j_1j_2\ldots jm}$ take a positive integer.

The value of cubic case $M_{j_1j_2\ldots jm}$ represents the total number of original data $S$ clusters of this $n\times m$ group.

**Definition 3** Let $M_i(i=0,1,2,3,4)$ denote the lattice sum value of the cube $M_{j_1j_2\ldots jm}$ margins, in the case when the fixed m-1 subscript value is i, that is, the fixed m-1 undertaken acquisition value is i, the sum value for the other lower bound $j$ is obtained, and the lower $j$ can take the value from i to 4 and $j$ can be located at any of the m positions. Intuitively, $M_i$ represents the number of original data for taking i, etc. in the book wide cube:

$M_i = \sum_{j_m=0}^{4} \sum_{j_{m-1}=0}^{4} \cdots \sum_{j_1=0}^{4} M_{j_1j_2\ldots jm} - (m-1)M_{i\ldots i\ldots i}$, $j$ in $j_1, j_2, \ldots, or j_m$

In the expression, the fixed m-1 underneath is i and one lower j is for the sum from i to 4. If m underneath is completed sequentially, the obtained is $M_i$. Note $M_{ii,\ldots,i}$ can not obtain the sum by overlapping.

If the sample is large enough, the frequency is close to the probability, and the probability of dropping to i ($i=0, 1, 2, 3, 4$) is as follows.

$P_{ci} \approx f_{ci} = \frac{M_i}{n}$

In general, the probability $P_{ci}$ in grade i, etc., should be greatly different from the corresponding SSi probability $P_{Si}$ of "3σ rule" or the like. If the software is unreliable, the $P_{ci}$ ($i=0, 1, 2, 3, 4$) may be much larger than the probability $P_{Si}$ of the usual I interval SSi. It may be less than the probability $P_{Si}$ of the corresponding i interval. This allows you to analyze the runtime reliability of software through a sample cluster.
5. Simulation experiment

Before the cluster analysis, it is necessary to calculate the frequency approximation of the cube at n sample points by collecting the sampled samples discretely and then collecting them into a generalized cubic cube. The generalized cube is inconvenient to figure drawing, and shows three dimensional cube cluster diagram with three evaluation indices of CPU utilization rate, memory occupancy size, and subnet average data stream as an example, and dynamically collects and dismantles 2000 data from the third order It was collected in the original cube. Experiments showed that the number of sample points concentrated at level 1 (cube C(1,1,1) was overwhelmingly, followed by grade 2 (cubic C(2,2, 2)), and the other 3,4,5 stages gradually decreased. Of course, a small amount of samples are scattered in other cubic lattices.

6. Conclusions

Software systems are characterized by loose polymerization, open dynamic and complex behavior, and are faced with many new challenges in terms of availability, reliability and maintainability and monitoring and behavior analysis at runtime. The reliability of software is primarily responsible for its operation, that is, the operation and outcome of the software system is consistent with the expectations of the people, and is characterized by its external characteristics, and these features are reliable to some extent displayed. This paper proposes the reliability of the execution time by generalized cube cluster analysis software.

7. Acknowledgements

Weiqing Wan(1973.10-), male, Fengcheng county, Jiangxi Province, Han, postgraduate degree, associate professor, majoring in: software engineering, trusted software. Qingyan Zeng(1974.10-), male, Wan'an county, Jiangxi Province, Han, postgraduate degree, lecturer, majoring in: network Engineering. Corresponding author: Zhicheng Wen (1972.11-), male, Dong'an county, Hunan province, Han, Ph.D, professor, majoring in: network security, trusted software.

References

[1] X. Zhang, X. Wang,Y. Kang,Trustworthiness requirement-oriented software process modelling, Journal of Software: Evolution and Process, 30(12)(2018) 58-71.
[2] J. Tian, Y. Guo, Software behavior trust forecast model based on check point scene information, Journal on Communications, 39(9)(2018) 147-158.
[3] M. Sun, S. Zhang, P. Nie, X. Nie, Research on session segmentation of web search query logs based on naive Bayes, Journal of Nanjing University(Natural science), 54(6)(2018) 1132-1140.
[4] W. Zhang, Z. Wang, J. Yuan, H. Liu, A locally attribute weighted naive Bayes classifier, Journal of Beijing jiaotong University, 42(2)(2018) 14-21.
[5] L. Yang, C. Cao, J. Sun, L. Zhang, Study on an improved naive Bayes algorithm in spam filtering, Journal on Communications, 38(4)(2017) 140-148.