Proceedings of 2010 Conference on Dependable Computing (CDC'2010)
November 20-22, 2010,Yichang,China

SA Based Software Deployment Reliability Estimation Considering Component and Host Node Dependences

Xihong Su, Zhibo Wu, Hongwei Liu, Xiaozong Yang, Decheng Zuo

School of Computer Science and Technology
Harbin Institute of Technology
Harbin, 150001,China
suxh@ftcl.hit.edu.cn

***Abstract* - System deployment architecture can have a significant impact on its reliability. Although many approaches for architecture-based reliability estimation exist, little work has been dong in incorporating the influence of system deployment architecture and hardware resources. This paper proposes a novel approach to estimate system reliability at architecture level, incorporating the influence of deployment architecture and hardware resources. Additionally, we give the definitions of component dependence and host node dependence. On the basis of matrix of dependence degree among components and matrix of dependence degree among host nodes, we optimize greedy algorithm to deploy software components onto host nodes. Finally, we calculate system reliability of the deployment architecture.**

***Keywords* - Software architecture, Software deployment, component, Reliability**

I. INTRODUCTION

Over the past a few decades, we have witnessed an unrelenting pattern of growth in the size and complexity of software systems, which will likely continue well into the foreseeable future. This pattern is further evident in an emerging class of embedded and distributed software systems that are growing in popularity due to increase in the speed and capacity of hardware, and decrease in its cost, the emergence of wireless ad hoc networks, proliferation of sensors and handheld computing devices, and so on. A number of researches have shown that a promising approach to resolve the challenges is to employ the principles of software architectures [1].

Several recent approaches have begun to quantify software reliability at the level of architectural models, or at least in terms of high-level system structure [2-8]. All of these efforts focus on system-level reliability prediction. However, these reliability estimation approaches are typically limited to certain classes or exclusively concentrates on software reliability, neglecting the influence of hardware resources and system deployment.

Highly distributed and mobile systems are challenged by the problem of disconnected operations. Disconnected operations force subsystems executing on each host node to temporarily operate independently from other host nodes. In this paper, we consider that the main source of disconnected operation is host node failures and component failures.

This paper is organized as follows. Section 2 explains the related concepts. In Section 3, we provides a brief description of problem definition, and presents optimized greedy algorithms to deploy components onto host nodes, and gives a way of calculating system reliability. In section 4, we run the optimized greedy algorithm on the basis of randomly generating inputs, and analyze the simulated results. Finally, the last section concludes the paper.

II. TERMINOLOGY AND SCOPE

A. *Software Architecture*

Software architecture is an abstraction of the run-time elements of a software system during some phase of its operations. A system may be composed of many levels of abstraction and many phases of operations, each with its own software architecture [9].

B. *Software Deployment*

Software deployment is referred to as a collection of activities, which are to make software available for use until uninstalling it from devices. These activities include delivery, installation,configuration, activation, updating, reconfiguration and un-installation of the software [10].

C. *Component Dependence*

A graph $G=(C,E)$ is undirected graph, C represents a set of software components. $C=\{C_1,C_2,\ldots,C_n\}$, $|C|=n$. E represents a set of interactions among software components.

Definition 1 Component dependence: given an undirected graph $G=(C,E)$, if $\exists C_i,C_j \in C$, $(C_i,C_j)\in E$, then component C_i depends on C_j. The relationship of component C_i and C_j is component dependence.

Given an undirected graph $G=(C,E)$, $\forall C_i,C_j,C_k \in C$, in order to describe whether component C_j is more important than component C_k for component C_i, we introduce the concept of component dependence degree.

Definition 2 Component dependence degree: given an undirected graph $G=(C,E)$, $\forall C_i,C_j \in C$, $cdd_{i,j}$ describes the degree of component C_i depends on component C_j. $cdd_{i,j}$ is a real number [0,1].

It makes sense that component dependence degree is determined by some factors, such as individual component reliability, host node reliability, architectural style, frequency of interaction among components and so on. Component dependence degree may be the linear relation of these factors. At last, we need to normalize component dependence degree to be a real number [0, 1].

Matrix *CDD* describes the component dependence degree between two arbitrary components. $C=(C_1,C_2,\ldots,C_n)$ is a finite set of software components

$$CDD=\begin{array}{c|ccc} & C_1 & \cdots & C_n \\ C_1 & cdd_{1,1} & \cdots & cdd_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ C_n & cdd_{n,1} & \cdots & cdd_{n,n} \end{array}.$$

D. Host Node Dependence

A graph $G=(C,E)$ is undirected graph, C represents a set of software components. $C=\{C_1,C_2,\ldots,C_n\}$, $|C|=n$. E represents a set of interactions among software components. Component subsets $CS_1,CS_2,\ldots,CS_k$ and host nodes $H_1,H_2,\ldots,H_k$ satisfy two following conditions:

1) $\forall i,j, CS_i\bigcap CS_j=\phi$, $\bigcup_{i=1}^{K} CS_i=C$;

2) $\forall i,j$, CS_i can be if and only if deployed on host node H_j.

We use host node dependence degree $hnd_{i,j}$ to describe the dependence relationship among component sets deployed on host node H_i and H_j. $hnd_{i,j}$ should be a real number [0,1]. $hnd_{i,j}$ can be obtained using (1)

$$hnd_{i,j}=\begin{cases} 0 & if\ i=j \\ \dfrac{FH_{i,j}}{\sum_{j=1}^{n} FH_{i,j}} & if\ \ i\neq j \end{cases}, \qquad (1)$$

$FH_{i,j}$: frequency of interaction among components deployed onto host node H_i and components deployed onto host node H_j.

Matrix *HND* describes the host node dependence degree among host nodes. Where $H=(H_1,H_2,\ldots,H_n)$ is finite set of host nodes.

$$HND=\begin{pmatrix} hnd_{1,1} & \cdots & hnd_{1,n} \\ \vdots & \ddots & \vdots \\ hnd_{n,1} & \cdots & hnd_{n,n} \end{pmatrix}.$$

III. Problem and Approach

A. Problem Definition

The distribution of software components onto host nodes greatly influences the system reliability in the face of host node failures and component failures. During system run-time, the parameters that influence system deployment architecture may change. Therefore, the current existing software deployment architecture may be ill-suited for the given state of system environment, and the system need to be redeployed to improve system reliability.

There are many factors influencing system redeployment, (1) available memory on each host; (2) possible restrictions on component locations; (3) frequency of interaction among components; (4) architectural style and so on. We translate the multi-dimensional influence on system deployment architecture into component dependence degree matrix and host node dependence matrix. On the basis of the two matrixes, we optimize greedy algorithm to deploy software components onto host nodes.

B. Our Approach

In our approach, we assume that the frequencies of interaction among software components are stable over a given period of time T. Therefore, the dependence degree among host nodes is also stable over a given period of time T. A greedy algorithm always makes the choice that looks best at the moment. That is, it makes a locally optimal choice in the hope that this choice will lead to a globally optimal solution [11]. On the basis of greedy algorithm, we design three sub-algorithms: splitting software components into component subsets, calculating dependence degree among component subsets and deploying component subsets onto host nodes.

1) The first sub-algorithm: splitting software components into component subsets

We optimize greedy algorithm to produce M component subsets. The algorithm is described as follows:

```
// $SS$ is the set of having been selected components;
// $RS$ is the set of remained components.
//M is the number of host nodes; N is the number of components.

INPUTS: Component dependence degree matrix $CDD$, $SS=\phi$,
$RS=\{C_1,C_2,\ldots,C_N\}$
OUTPUTS: M component subsets

//find four approximate max values of dependence degree among
 components
for (i=1;i<=M;i++)
  {
  find the max value of dependence degree among components from
matrix $CDD$, for example, $cdd_{k,l}$
   delete row $k$, $l$ and column $k$, $l$ from matrix $CDD$;
   $SS[i]=\{C_k,C_l\}$;
   $label(C_k)=label(C_l)=i$;
   $RS=RS-\{C_k,C_l\}$;
  }

// Remained components can be allocated to the above M component
subsets: // $SS[1],SS[2],\ldots,SS[M]$.
While ( $RS!=\phi$ )
  { $SS=\bigcup\sum_{i=1}^{M} SS[i]$;
  for each $C_k\in SS$ and each $C_l\in RS$
  {
     find the max value of degree of component dependence $cdd_{k,l}$;
     $label(C_l)=label(C_k)$;
     $SS(label(C_k))=SS(label(C_k))\bigcup C_l$;
     $RS=RS-C_l$;
     }
}
Output the M component subsets: $SS[1]$ ,…, $SS[M]$
```

2) The second sub-algorithm: calculating dependence degree among component subsets

After selecting M component subsets, we need to calculate the dependence degree among component subsets. On the basis of the frequency matrix of interaction among components, the dependence degree among component subsets can be calculated as follows:

```
INPUTS: the frequency matrix of interaction among component subsets
OUTPUTS: the matrix of dependence degree among component subsets, CDS

for (i=1; i<=M;i++)
  { $CDS[i][i]=0$ ;
      for (j=1;j<=M;j++)
        {
          for each $C_k \in SS[i]$ and $C_l \in SS[j]$
              $a[j]=a[j]+f(C_k,C_l)$ ;
        }
      $sum=\sum_{i=1}^{M} a[i]$ ;
      $CDS[i][j]=a[j]/sum$ ;
  }
```

3) The third sub-algorithm: deploying component subsets onto host nodes

On the basis of the matrix of dependence degree among component subsets and the matrix of dependence degree among host nodes, we deploy component sets onto host nodes. The steps are as follows:

Step 1: From matrix CDS and matrix HND , choose $CDS[i][j]=0\ (i \neq j)$ and $hnd_{m,n}=0\ (m \neq n)$. Components of component subset $SS[i]$ may be deployed onto host node H_m or H_n .

Step 2: Find the max value of dependence degree among component subsets from matrix CDS , for example, $CDS[k][l]$. Find the max degree value of host node dependence from matrix HND , for example, $hnd_{p,q}$. Components of component sets $SS[k]$ may be deployed onto host node H_p or H_q .

Step 3: Components of one component subset can only and only if be deployed onto one host node every system redeployment. If candidate host node set of component subset $SS[i]$ includes only one host node H_j , components of component subset $SS[i]$ can be deployed on host node H_j .

Step 4: If component subset $SS[i]$ has multiple candidate host node sets, components of component subset $SS[i]$ will be deployed on one host node that is the intersection of candidate host node sets.

Step 5: Loop step 1 to step 4 until all components have been deployed on host nodes.

C. System Reliability

Reliability is defined as the probability for components or systems to function successfully against the expected environment for a given time. System reliability estimations are often obtained on the basis of the reliability information of subsystems or components [12]. In this paper, we consider that the main source of the degradation of system reliability is component failure and host node failure. The system reliability can be calculated as follows:

$$R_{system}=1-p_{system}=1-\bigcup_{i=1}^{N} PH_i-\left(\prod_{i=1}^{N}(1-PH_i)\right)\cdot\left(\bigcup_{i=1}^{N} PCSH_i\right)$$

(2)

$$PCSH_i=1-\prod_{j=1}^{m}(1-PC_j)=1-\prod_{j=1}^{m}(1-PCH_j\cdot(1-PH_i)) \quad ,$$

(3)

R_{system}	system reliability
p_{system}	failure probability of system
N	number of host nodes
$PCSH_i$	failure probability of components deployed on host node H_i
PCH_j	failure probability of component C_j deployed on one host node, if the host node does not fail.
PC_j	failure probability of component C_j .
PH_i	failure probability of host node H_i .

IV. Experiment

The experiment run has four host nodes and fourteen components. We run the described algorithm on randomly generated inputs. These inputs include conditional failure probability of components, failure probability of host nodes, matrix of dependence degree of host nodes, matrix of dependence degree among components and the frequency matrix of component interaction.

A. Randomly Generated Inputs

1) Failure probabilities of four host nodes

Failure probabilities of four host nodes can be real numbers [0.0, 0.01]. $PH_1=0.0013$, $PH_2=0.0005$, $PH_3=0.0009$, $PH_4=0.0002$.

2) Frequency matrix of component interaction

Frequency matrix of component interaction is CF . Each entry in matrix CF is an integer number [0, 8]. It represents the frequency of interaction among components.

$$CF=\begin{bmatrix}
0&2&0&0&0&0&0&0&7&4&0&0&0&0\\
2&0&7&0&0&0&0&0&0&0&1&0&0&0\\
0&7&0&6&0&0&0&0&0&4&0&0&7&0\\
0&0&6&0&6&0&0&0&0&0&0&4&0&0\\
0&0&0&6&0&7&0&0&0&0&0&0&3&0\\
0&0&0&0&7&0&4&0&0&0&0&0&0&7\\
0&0&0&0&0&4&0&0&7&0&0&0&0&4\\
0&0&0&0&0&0&0&0&5&0&0&5&0&0\\
7&0&0&0&0&0&7&5&0&2&0&0&0&6\\
4&0&4&0&0&0&0&0&2&0&6&0&0&0\\
0&1&0&0&0&0&0&0&0&6&0&0&0&0\\
0&0&0&4&0&0&0&5&0&0&0&0&3&0\\
0&0&7&0&3&0&0&0&0&0&0&3&0&4\\
0&0&0&0&0&7&4&0&6&0&0&0&4&0
\end{bmatrix}.$$

3) Matrix of dependence degree of host nodes

Degree matrix of host node dependence is matrix HND . Each entry in matrix HND represents dependence degree how one host node depends on another host node.

$$HND = \begin{bmatrix} 0 & 0.2728 & 0.7272 & 0 \\ 0.2751 & 0 & 0 & 0.7249 \\ 0.6550 & 0 & 0 & 0.3450 \\ 0 & 0.1449 & 0.8551 & 0 \end{bmatrix}.$$

4) *Conditional failure probability of fourteen components*

Conditional failure probability of fourteen components can be real number [0.0, 0.01]. $PCH_1 = 0.0004$, $PCH_2 = 0.0013$, $PCH_3 = 0.0007$, $PCH_4 = 0.0003$, $PCH_5 = 0.0011$, $PCH_6 = 0.0002$, $PCH_7 = 0.0006$, $PCH_8 = 0.0012$, $PCH_9 = 0.0008$, $PCH_{10} = 0.0013$, $PCH_{11} = 0.0021$, $PCH_{12} = 0.0012$, $PCH_{13} = 0.0001$, $PCH_{14} = 0.0002$.

5) *Degree matrix of component dependence*

Degree matrix of component dependence is CDD .Each entry in matrix CDD is a real number [0,1]. It represents the degree how a component depends on another component.

$$CDD = \begin{bmatrix}
0 & 0.8133 & 0.3097 & 0.4439 & 0.2168 & 0.1363 & 0.4256 & 0.0550 & 0.9776 & 0.4796 & 0.1351 & 0.3167 & 0.1630 & 0.3580 \\
0.5568 & 0 & 0.8068 & 0.2999 & 0.3518 & 0.6170 & 0.1628 & 0.2998 & 0.3134 & 0.2411 & 0.4960 & 0.2334 & 0.3371 & 0.3382 \\
0.4856 & 0.9275 & 0 & 0.6376 & 0.0528 & 0.2690 & 0.6562 & 0.3931 & 0.0294 & 0.5459 & 0.2875 & 0.0084 & 0.1990 & 0.4877 \\
0.6522 & 0.1121 & 0.3743 & 0 & 0.8481 & 0.2207 & 0.1962 & 0.2154 & 0.3576 & 0.3911 & 0.0609 & 0.3969 & 0.2293 & 0.0910 \\
0.2319 & 0.6271 & 0.2916 & 0.1443 & 0 & 0.7129 & 0.4762 & 0.1824 & 0.0272 & 0.2113 & 0.2625 & 0.3499 & 0.8021 & 0.3738 \\
0.4787 & 0.3438 & 0.0974 & 0.3516 & 0.7937 & 0 & 0.6133 & 0.0768 & 0.3109 & 0.0929 & 0.1863 & 0.0850 & 0.3683 & 0.5149 \\
0.2265 & 0.2945 & 0.3974 & 0.6461 & 0.3066 & 0.9413 & 0 & 0.0074 & 0.9992 & 0.0217 & 0.6171 & 0.4688 & 0.3710 & 0.2216 \\
0.4927 & 0.3155 & 0.3333 & 0.5159 & 0.1102 & 0.3299 & 0.0311 & 0 & 0.7207 & 0.1595 & 0.1233 & 0.9697 & 0.5206 & 0.4250 \\
0.9442 & 0.0034 & 0.1930 & 0.6302 & 0.6544 & 0.0178 & 0.2886 & 0.7045 & 0 & 0.8445 & 0.0134 & 0.4148 & 0.6705 & 0.0682 \\
0.9096 & 0.6820 & 0.8386 & 0.3099 & 0.1311 & 0.6434 & 0.6711 & 0.5799 & 0.1326 & 0 & 0.3697 & 0.4820 & 0.4869 & 0.6641 \\
0.6222 & 0.8995 & 0.2584 & 0.2688 & 0.0683 & 0.2816 & 0.6505 & 0.3525 & 0.3100 & 0.1870 & 0 & 0.2376 & 0.5175 & 0.2077 \\
0.0133 & 0.3928 & 0.0429 & 0.5365 & 0.1252 & 0.5802 & 0.2280 & 0.7221 & 0.1348 & 0.6913 & 0.5893 & 0 & 0.6416 & 0.1611 \\
0.4675 & 0.0059 & 0.4393 & 0.1633 & 0.1662 & 0.4496 & 0.6585 & 0.6685 & 0.2233 & 0.4120 & 0.2938 & 0.2632 & 0 & 0.6382 \\
0.6437 & 0.4010 & 0.2744 & 0.2110 & 0.6114 & 0.3796 & 0.6799 & 0.1557 & 0.3965 & 0.4714 & 0.1567 & 0.4138 & 0.6609 & 0
\end{bmatrix}.$$

B. *Deployment Algorithm and System Reliability*

On the basis of matrix of dependence degree among components, we use the first sub-algorithm to divide fourteen components into four component subsets. These four component subsets are $\{C_5, C_6, C_7, C_9, C_{13}, C_{14}\}$, $\{C_8, C_{12}\}$, $\{C_2, C_3, C_4, C_{11}\}$ and $\{C_1, C_{10}\}$.

We use the second sub-algorithm to calculate the dependence degree among component subsets. Then, we use the third sub-algorithm to deploy software components onto host nodes. $\{C_1, C_{10}\}$ can be deployed on host node H_1 , $\{C_2, C_3, C_4, C_{11}\}$ can be deployed on host node H_2 , $\{C_5, C_6, C_7, C_9, C_{13}, C_{14}\}$ can be deployed on host node H_3 , $\{C_8, C_{12}\}$ can be deployed on host node H_4 . Finally, system reliability can be calculated using (2): $R_{system} = 0.975439$.

V. CONCLUSIONS

It is very important to estimate system reliability at architecture-level. In this paper, we define the concepts of component dependence and host node dependence. On the basis of the matrix of dependence degree among components and the matrix of dependence degree among host nodes, we optimize greedy algorithm to deploy software components on host nodes. Then we give a way of calculating system reliability of SA based software deployment. On the basis of randomly generated inputs, we run optimized greedy algorithm and calculate system reliability.

ACKNOWLEDGMENT

The work described in this paper is supported by High Technology Research and Development Program of China (Project No. 2008AA01A201), National High Technology Research, Development Plan of China (Project No. 2006AA01A103), High Technology Research and Development Program of China (Project No. 2009AA01A404), National High Technology Research and Development Plan of China (Project No.2008AA01A204).

REFERENCES

[1] C.Seo, S.Malek, G.Edwards, D. Popescu, N, Medvidovic, B. Petrus, S. Ravula, "Exploring the Role of Software Architecture in Dynamic and Fault Tolerant Pervasive Systems," Proc. IEEE Symp. Software Engineering for Pervasive Computing (SEPCASE'07), May 2007, pp.9-15.

[2] V. Cortellesa, V.Grassi, "A Modeling Approach to Analyze the Impact of Error Propagation on Reliability of Component-Based Systems", In Proc.CBSE-10. Jul 2007, pp.140-156.

[3] S. Gokhale , K. Trivedi, "Reliability Prediction and Sensitivity Analysis Based on Software Architecture", Proc. ISSRE'02. 2002, pp.64-75.

[4] K.G. Popstojanova, et al, "Architectural level Risk Analysis Using UML, IEEE Transactions on Software Engineering", vol. 29, no.10, pp. 946-960, October 2003.

[5] K.G. Popstojanova, et al,et al, "Architecture-Based Approaches to Software Reliability Prediction", Journal of Computer & Mathematics with Applications, vol.46,no.7, pp.1023-1036, October 2003.

[6] R. Reusser, et al, "Reliability Prediction for Component-based Software Architectures", Journal of Systems and Software, vol.66, no.3, pp.241-252, 2003.

[7] W. Wang, Y. Wu, M. Chen, "An Architecture-based Software Reliability Model", In Proc. of Pacific Rim International Symposium on Dependable Computing, pp.143-150,1999.

[8] S. M. Yacoub, et al, "Scenario-Based Reliability Analysis of Component-Based Software", In 10th International Symposium on Software Reliability Engineering, Boca Raton, Nov.1999.

[9] R. Thomas, "Architectural Styles and the Design of Network-based Software Architectures", Ph.D.Dissertation, University of California, Irvine, 2000.

[10]C. Vo, T. Torabi, "A Framework of Over the Air Provider-initiated Software Deployment on Mobile Devices", Proc. IEEE Symposium on Software Engineering(ASWEC'08), March 2008, pp.633-638.

[11]C. Thomas, L.Charles, R.Ronald, S.Clifford, "Introduction to Algorithms", 2nd ed., MIT Press, 2001, pp.370-378.

[12]T. Jin, "Hierarchical Variance Decomposition of System Reliability Estimates with Duplicated Components", IEEE Transaction on Reliability, Dec.2008, pp.564-573.