软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2014,25(11):2702−2714 [doi: 10.13328/j.cnki.jos.004548]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种基于隐马尔可夫模型的虚拟机失效恢复方法*

张建华[1,2,3], 张文博[1], 徐继伟[1,2,3], 魏 峻[1,2], 钟 华[1], 黄 涛[1,2]

[1](中国科学院 软件研究所 软件工程技术中心,北京 100190)

[2](计算机科学国家重点实验室(中国科学院 软件研究所),北京 100190)

[3](中国科学院大学,北京 100049)

通讯作者: 张建华, E-mail: jhzhang@otcaix.iscas.ac.cn

摘 要: 随着虚拟化技术的发展与普及,越来越多的企业将关键业务系统部署到了虚拟化平台上.虚拟化技术降低了企业的硬件和管理成本,但同时也给系统的可靠性带来了严峻挑战.传统的方法通过运行时系统状态备份的方法来提高系统的失效恢复能力,但该方法会引入了巨大的系统开销.提出了一种基于隐马尔可夫模型的系统失效恢复性能优化方法.通过对系统运行时状态的预测分析,计算系统未来运行状态的概率趋势,并在运行过程中动态调整系统失效恢复功能与正常业务功能之间的资源分配,从而降低了系统的运行时性能开销,提高了业务系统服务能力.实验分析显示,该方法可以在保障系统可靠性的同时有效地降低系统的性能开销,在系统运行状态稳定的情况下,最高可以降低 2/3 的系统响应时间.

关键词: 虚拟机;失效恢复;隐马尔可夫模型;预测

中图法分类号: TP316

中文引用格式: 张建华,张文博,徐继伟,魏峻,钟华,黄涛.一种基于隐马尔可夫模型的虚拟机失效恢复方法.软件学报,2014,25(11):2702−2714. http://www.jos.org.cn/1000-9825/4548.htm

英文引用格式: Zhang JH, Zhang WB, Xu JW, Wei J, Zhong H, Huang T. Approach of virtual machine failure recovery based on hidden Markov model. Ruan Jian Xue Bao/Journal of Software, 2014,25(11):2702−2714 (in Chinese). http://www.jos.org.cn/1000-9825/4548.htm

Approach of Virtual Machine Failure Recovery Based on Hidden Markov Model

ZHANG Jian-Hua[1,2,3], ZHANG Wen-Bo[1], XU Ji-Wei[1,2,3], WEI Jun[1,2], ZHONG Hua[1], HUANG Tao[1,2]

[1](Technology Center of Software Engineering, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

[2](State Key Laboratory of Computer Science (Institute of Software, The Chinese Academy of Sciences), Beijing 100190, China)

[3](University of Chinese Academy of Sciences, Beijing 100049, China)

Corresponding author: ZHANG Jian-Hua, E-mail: jhzhang@otcaix.iscas.ac.cn

Abstract: With the development and popularization of virtualization technology, more and more enterprises will deploy their business-critical systems on virtualization platform. While reducing the company's hardware and management costs, virtualization also brings severe challenges for system reliability. While the runtime system state replication backup method can improve the failure recovery capabilities of system, it also introduces huge overhead. This paper presents a performance optimization method based on hidden Markov model for system failure recovery. It analyzes runtime states of the system, and calculates the probability of system running tendency. Business system optimization is achieved by dynamically adjusting resources allocation between the failure recovery function and normal business function to reduce the runtime overhead. Experimental results show that the presented approach can guarantee reliability of the system while effectively reducing performance overhead by up to 2/3.

* 基金项目: 国家自然科学基金(61173003); 国家高技术研究发展计划(863)(2012AA011204); 国家科技支撑计划(2012BAH14B02)

收稿时间: 2013-08-20; 定稿时间: 2013-12-05

Key words: virtual machine; failure recovery; hidden Markov model; prediction

随着现代企业IT资源的不断增加,大量异构的物理服务器并存,多种业务系统分布于不同的系统平台之上,使得整个IT系统的管理非常复杂,并且资源利用率低下.为了降低企业的IT运营成本,提高管理效率,以虚拟化技术为基础的云计算[1]平台近年来得到了越来越广泛的应用.通过虚拟化[2,3]技术,企业可以把原来运行于物理计算机的关键业务系统迁移到虚拟机平台上(如图1所示),使得原来多个利用率低下的业务系统可以集中部署到一台物理计算机上,从而可以大大提高企业整体资源的利用率,降低管理成本.

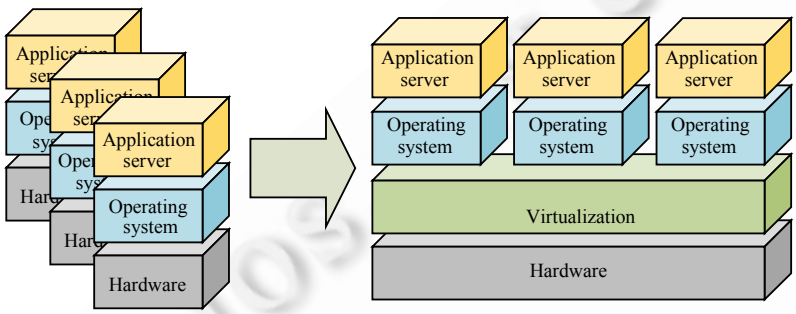


Fig.1 Centralized deployment of business systems on the virtualization platform

图1 虚拟化平台下的业务系统集中部署

虚拟化在给企业带来好处的同时也带来了巨大的挑战.在传统架构的平台上,1台物理机仅运行1个用户的业务系统,因此,物理机的宕机仅会对其本身所运行的业务系统造成影响.物理机作为一种天然的隔离和保护屏障,有效地保障了不同业务系统之间不会相互影响.但是在采用了虚拟化技术的云平台上,一个物理机上可能运行着多个虚拟机,不同用户的业务系统则运行于虚拟机上,在这种新的架构平台上,因某一个业务系统导致的物理机宕机将会导致更加严重的后果.近几年,许多知名的公有云服务商发生了多次服务失效事件,且导致了严重的损失.如,Amazon的宕机[4]事件导致了依赖其服务的多家公司停止服务,Google账户的24小时中断服务[5]导致大量用户不能访问自己的云端数据,FlexiSale服务商的18小时宕机事件[6]也导致了大量用户访问不到自己的数据内容.在虚拟化平台环境下,这种业务系统集中部署和管理的方法,使得虚拟化环境下服务器的失效恢复问题显得尤为重要.大量的研究工作针对这个问题进行了深入的研究,其中一种方法是基于日志重放的系统备份方法,通过在备份节点重放主节点的每一个操作来实现对主节点系统的备份;另一种方法是基于系统状态复制的备份方法,通过每隔一段时间把当前系统的状态备份到远程节点来实现系统的可靠性保障.但是,上述工作在保证系统失效恢复的同时却引入了很大的系统开销:在普通处理任务情况下,系统大约有50%的性能开销;在特殊情况下,如网络密集型应用,系统开销则高达75%[7].这严重影响了业务系统的可用性,导致大量计算资源浪费在预防很少发生的系统失效上.

针对现有工作的不足,本文提出了一种基于隐马尔可夫模型(hidden Markov model,简称HMM)的虚拟机失效恢复方法.由于业务系统的运行往往具有一定的规律性和周期性,因此可以通过建立系统运行时状态的隐马尔可夫模型,并对系统的未来运行状态进行预测分析:当系统的预测运行状态处于正常时,系统资源被更多地分配给业务系统执行业务计算,从而保障客户的服务质量;当系统的预测状态偏离正常状态而有可能发生错误时,则调整系统的资源分配,把更多的资源用于系统运行时状态的复制和备份上,从而保证系统能够快速地从可能发生的失效中恢复.

本文的主要贡献包括如下几个方面:1) 提出并建立了一种虚拟化环境下的基于隐马尔可夫模型的系统状态变化的预测模型;2) 大幅度降低了失效恢复系统的运行时开销,提高了其商业化应用的可能性;3) 实现了该方法的原型系统,并验证了其有效性.

本文第 1 节给出本文方法的描述与系统的整体架构.第 2 节介绍如何构造虚拟化环境下系统状态的隐马尔可夫模型.第 3 节给出系统的自动化失效恢复方法.第 4 节给出实验设计和结果分析.第 5 节介绍并比较相关研究工作.第 6 节对本文工作加以总结.

1 方法描述与整体架构

虚拟化技术的应用,使企业得以把原来独占物理主机的业务系统集中部署到虚拟化环境下.这种部署方式在提高资源利用率和降低管理成本的同时,也严重降低了系统的可靠性.虽然一些工作[7,8]试图通过备份的方式来解决这一问题,但却引入了较大的系统开销.另外,由于系统的运行状态具有一定的规律性和周期性,因此可以通过对业务系统历史数据的分析来对系统的未来运行状态进行预测.本文提出的基于隐马尔可夫模型的虚拟机失效恢复方法即是基于以上观察,首先构造系统状态转移预测模型 HMM,并对其各项参数进行训练以得到正确反映系统行为特征的模型;然后,通过实时地对系统当前状态的采集,形成一个系统状态变化的时间序列,并通过 HMM 模型对此时间序列进行计算分析,从而达到预测系统未来运行状态的目标;在得到系统的运行状态趋势之后,则通过动态地调整业务系统与可靠性保障系统之间的资源开销,使得系统运行在趋于稳定的时候把更多的资源用于业务计算,提供系统的对外服务能力;而在系统可能即将发生问题时,则把更多的资源用于系统的可靠性保障,从而可以保证系统能够从可能发生的失效中快速恢复,减少系统的宕机时间.

整个系统架构如图 2 所示,分成如下几个部分:首先是数据采集模块,负责从虚拟化环境中采集系统状态数据,并对采集的数据进行初步的预处理,以符合后续的模型输入要求;系统预测模块则主要负责根据当前的系统状态时间序列,计算并输出系统未来状态的概率趋势,从而指导动态调整模块对系统的调整;系统的预测模型是通过历史数据的训练得到的,并且在系统的运行过程中,根据最新的数据动态地更新,从而使得模型更加符合系统当前的运行状态.

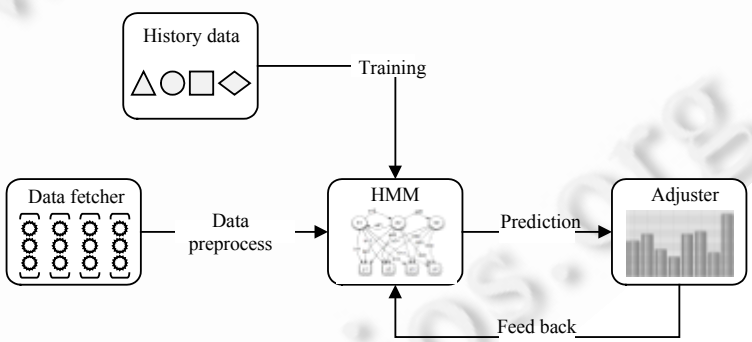


Fig.2 System architecture

图 2 系统体系架构

2 虚拟化环境下的系统状态建模

根据上一节对本文方法的描述与分析,我们首先需要建立虚拟化环境下的系统隐马尔可夫模型,并对模型参数进行训练和学习,以此来对系统运行时状态进行预测.本节将详细介绍虚拟化环境下系统状态模型的构造、预测结果的计算方式、模型参数的计算等.

2.1 状态预测模型的选择

为了实现对系统未来状态的预测,首先需要建立系统状态的预测模型.隐马尔可夫模型作为一种有效的模式识别和预测模型,其模型的基本元素和结构与我们所要建立的模型的系统特性基本相吻合.在虚拟化环境下,系统运行状态的健康状况在系统出现宕机等异常之前往往是不可观测的,只能通过其他一些系统特征参数(如 CPU、内存)来猜测和推断当前的系统健康状况.这正好与隐马尔可夫模型的隐藏状态相一致,一个隐马尔可夫

模型如图 3 所示,它可以用一个五元组来描述$\lambda=(S,O,A,B,\pi)$,其中,$S=\{S_1,S_2,\ldots,S_N\}$是隐马尔可夫模型的隐含状态,N 是指隐马尔可夫模型的隐含状态的总数.因此,我们可以把系统的健康状况建模成隐马尔可夫模型的隐藏状态.

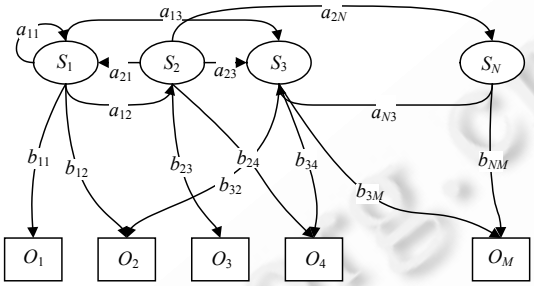


Fig.3 Hidden Markov model

图 3 隐马尔可夫模型

另外,为了对系统的隐藏状态进行合理的猜测和推断,只能通过由隐藏状态引发的可观测的系统特征参数的变化规律来分析.例如,通过一系列的 CPU、内存、网络等特征参数的连续变化规律来推断系统的运行状态.这与隐马尔可夫模型的结构也是相吻合的,这些观测状态则对应于隐马尔可夫模型的参数 $O=\{O_1,O_2,\ldots,O_M\}$,表示隐马尔可夫模型中观测状态的集合,其中,M 为所有出现的观测状态的总数.

隐马尔可夫模型中的另外两个参数是两个状态转移矩阵,描述了系统各个状态之间的转移概率趋势,其中,

- $A=\{a_{ij},i,j=1,2,\ldots,N\}$为隐马尔可夫模型的隐含状态转移矩阵,$a_{ij}=P(x_{t+1}=S_j|x_t=S_i)$表示 t 时刻 HMM 模型由状态 S_i 转移到 S_j 的概率.
- $B=\{b_{jk},j=1,2,\ldots,N,k=1,2,\ldots,M\}$为隐马尔可夫模型的输出状态矩阵,即,隐藏状态到输出状态的概率矩阵,$b_{jk}=P(x_{t+1}=V_k|x_t=S_j)$表示 t 时刻 HMM 模型在状态 S_j 时产生观测状态 V_k 的概率.

隐马尔可夫模型中的参数$\pi=\{\pi_1,\pi_2,\ldots,\pi_N\}^T$表示初始状态概率分布,即,系统初始状态时处于各个隐藏状态的概率,其中,$\pi_i=P(x=S_i)$表示初始时刻时系统处于某个状态的概率.

2.2 状态空间的构造

根据上述隐马尔可夫模型的描述,我们首先需要建模系统的隐藏状态和可观测状态.对于系统的隐藏状态,这里我们取两个维度的向量数组 $S=\{H,L\}$,其中,H 表示系统的健康状况,其取值范围为$\{h_1,h_2,h_3\}$,分别表示系统处于正常、异常和失效状态;L 表示系统的负载状况,其取值范围为$\{l_1,l_2,l_3\}$,分别表示系统负载较低、适中和较大的状态.因此,我们可以得到系统的 9 个隐藏状态以及它们之间的概率转移矩阵:

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix},$$

其中,N 为隐藏状态个数,$0\leqslant a_{ij}\leqslant 1$ 且有 $\sum_{i=1}^{N} a_{ki}=1,k\in[1,9]$. 后续我们将给出如何通过参数训练来具体计算这些状态间的转移概率值.

可观测状态是由隐藏状态产生的可以观测的系统的运行状态,其中包含多个维度的系统状态信息,这里我们采用 CPU 利用率、内存利用率、网络利用率、磁盘读写速度这 4 个特征向量构成的向量数组 $V=\{C',M',N',D'\}$来表示系统的一个可观测状态,其中每一个特征向量都可以有多个状态.为了降低状态空间的复杂度,减少计算的难度,我们对每个特征向量进行离散化处理,将其划分成 2~3 个状态.以网络连接数为例,将其划分成两个状态$\{t_1,t_2\}$,分别表示当前网络连接数明显低于历史平均值和当前网络连接数明显高于历史平均值.因此,我们可以

得到构成隐马尔可夫模型的输出状态的概率矩阵:

$$B=\begin{bmatrix} b_{11} & b_{12} & b_{13} & \cdots & b_{1M} \\ b_{21} & b_{22} & b_{23} & \cdots & b_{2M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{N1} & b_{N2} & b_{N3} & \cdots & b_{NM} \end{bmatrix},$$

其中,N,M 分别为隐藏状态和可观测状态的个数,$0\leqslant b_{ij}\leqslant 1$ 且有 $\sum_{i=1}^{M} b_{ki}=1, k\in[1,9]$. 后续我们将给出如何通过参数训练来具体计算输出状态概率矩阵的值.

由于隐马尔可夫模型处理的是一维空间的状态序列,因此我们需要把从系统状态中获取的多维状态空间转换到一维状态空间,即,给定一个观测状态 $\{C_1',M_1',N_1',D_1'\}$, 计算其在一维空间中的顺序号.这里,我们假设观测状态中某个状态向量 O 的状态个数用函数 $f(O)$表示,那么我们可以得到观测状态 $\{C_1',M_1',N_1',D_1'\}$ 在一维空间中的顺序号为

$$N(\{C_1',M_1',N_1',D_1'\})=C_1'f(M')f(N')f(D')+M_1'f(N')f(D')+N_1'f(D')+D_1',$$

其中, $N(\{C_1',M_1',N_1',D_1'\})$ 表示观测状态转换到一维空间中的顺序号.

2.3 状态预测算法

系统状态的预测就是要求给定观测序列 $Y=(y_1,y_2,\ldots,y_t)$,计算系统的隐藏状态在下一个时刻 X_{t+1} 进入异常状态的概率 $P(X_{t+1}=X_i)$,其中,X_i 为异常状态,如图 4 所示.我们分两步来计算这个概率的值:

- 首先,我们计算 t 时刻系统分别处于不同状态的概率向量数组 π^t;
- 然后,在此基础上做一次状态转换的概率转移,从而得到 t+1 时刻系统处于不同状态的概率向量数组 X^{t+1}.

最后,对其中的异常状态求和,即为系统在 t+1 时刻出现异常的概率.

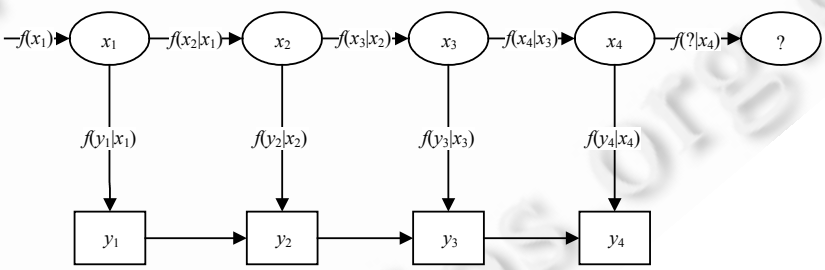


Fig.4 Probabilistic prediction algorithm process

图 4 概率预测算法过程

设 t 时刻系统隐藏状态向量表示为 $\pi^t=[\pi_1^t,\pi_2^t,\pi_i^t,\ldots,\pi_N^t]^T$, 其中,$N$ 表示系统隐藏状态的个数, π_i^t $(1\leqslant i\leqslant N)$ 表示 t 时刻系统隐藏状态处于状态 S_i 的概率,则 π^t 可以通过下面的公式来计算:

$$\pi_i^t=\max_{s_1,s_2,\ldots,s_{t-1}} P(x_1,x_2,\ldots,x_{t-1},x_t=i,y_1,y_2,\ldots,y_t\mid\lambda) \tag{1}$$

即,找到一个隐藏状态序列 $x_1,x_2,\ldots,x_{t-1}$,使得 $x_1,x_2,\ldots,x_{t-1},x_t=i$ 产生输出序列 $y_1,y_2,\ldots,y_t$ 的概率最大.其中,λ表示隐马尔可夫模型的参数 A,B 和 π.我们可以依次计算 π^t 的各个向量.因此,系统在 t+1 时刻处于各个状态的概率矩阵可表示为 $A_{t+1}'=\pi^t A$.对矩阵 A_{t+1}' 在列向量上求和,即可得到 t+1 时刻系统处于不同状态的概率向量数组,即

$$X_{t+1}=\left[\sum_{i=1}^{N}A_{t+1}'(i,1),\sum_{i=1}^{N}A_{t+1}'(i,2),\ldots,\sum_{i=1}^{N}A_{t+1}'(i,N)\right] \tag{2}$$

下面我们给出具体的算法伪代码.

算法 1. 系统状态预测算法.

输入:隐马尔可夫模型的参数 A,B,π,S,O.其中,

- A 为 $N\times N$ 的转移矩阵,令 A_{ij} 表示从状态 s_i 转移到 s_j 的概率;
- B 为 $N\times M$ 的输出矩阵,令 B_{ij} 表示从从状态 s_i 转移到 o_j 的概率;
- 初始状态概率分布 π,其中,π_i 表示 x_1 为 s_i 的概率;
- 观测状态序列 $Y=\{y_1,y_2,\ldots,y_T\}$ 表示在时刻 t 观测到的状态 o_i 的序列.

输出:系统在 T+1 时刻处于各个隐藏状态 S_i 的概率,即求 $P(X_{t+1}=s_i)$.

1. *FORCAST*(A,B,π,S,O,Y)
2. foreach (s_i in S)
3. $F[i,1]=\pi_i * B_{iy_1}$
4. end foreach
5. for (i=2,3,...,T)
6. foreach (s_j in S)
7. $F[j,i]=\max_z(F[z,i-1]*A_{zj}*B_{jy_i})$
8. end foreach
9. end for
10. for (i=1,2,…,N)
11. for (j=1,2,…,N)
12. $C_{ij}=F_{iT}\sum_{k=1}^{N}A_{kj}$
13. end for
14. end for
15. for (i=1,2,…,N)
16. $P_i=\sum_{k=1}^{N}C_{ki}$
17. end for
18. return P

通过上述计算过程,可以得到 t+1 时刻系统处于不同状态的概率.据此,我们可以根据对系统状态的预测结果对系统进行动态的调整.比如,下一个状态以较高的概率进入异常状态,我们则增加系统的备份频率,从而保证系统失效时可以较快地恢复正常,并尽可能地减少数据的丢失.

上述算法的复杂度为 $O(N^2T)$,其中,N 为系统隐藏状态的个数,T 是观测序列的长度.由于系统隐藏状态的个数和观测序列的长度是固定的,因此上述算法的执行并不会导致过多的系统性能损失.后续我们将通过实验数据给出验证.

2.4 模型参数训练

由于上述算法的计算过程需要事先知道隐马尔可夫模型的 3 个参数 A,B,π,因此本节将给出如何通过已有的历史数据训练得到所需要的隐马尔可夫模型的参数.这个问题可以归结为:在给定隐藏状态的集合 S 和观测状态的集合 O,通过一个已知的历史观测序列($y_1,y_2,\ldots,y_N$)来计算隐马尔可夫模型的 3 个参数 A,B,π.这是一个迭代计算的过程,包括下面几个步骤:首先,随机初始化隐马尔可夫模型的这 3 个参数;然后,我们定义向前变量 $\alpha_t(i)$ 表示在时刻 t 隐藏状态处于 S_i 且满足观测序列($o_1,o_2,\ldots,o_t$)的概率:

$$\alpha_t(i)=P(o_1,o_2,...,o_t,q_t=s_i\mid\lambda)=\sum_{k=1}^{N}\alpha_{t-1}(k)A_{ki}B_{j(o_t)}.$$

定义向后变量 $\beta_t(i)$ 表示在时刻 t 隐藏状态处于 S_i 且满足观测序列($o_{t+1},o_{t+2},\ldots o_T$)的概率:

$$\beta_t(i)=P(o_{t+1},o_{t+2},...,o_T \mid q_t=s_i,\lambda)=\sum_{k=1}^{N}A_{ik}B_{j(o_t)}\beta_{t+1}(k).$$

因此,我们可以得到 t 时刻处于隐藏状态 i、t+1 时刻处于隐藏状态 j 的概率为

$$\xi_t(i,j)=P(q_t=s_i,q_{t+1}=s_j \mid O,\lambda)=\frac{\alpha_t(i)A_{ij}B_{j(o_{t+1})}\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)A_{ij}B_{j(o_{t+1})}\beta_{t+1}(j)}.$$

在 t 时刻隐藏状态处于 i 的概率为

$$\gamma_t(i)=P(q_t=s_i \mid O,\lambda)=\sum_{j=1}^{N}\xi_t(i,j)=\frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)}.$$

由此,我们可以得到初始时刻隐藏状态 i 的期望值 $\overline{\pi}_i=\gamma_1(i)$, 隐藏状态转移矩阵的期望值 $\overline{A}_{ij}=\frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}$, 转移状态矩阵的期望值 $\overline{B}_{jk}=\frac{\sum_{t=1,o_t=k}^{T}\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(j)}$.

把上述技术得到的期望值代入$\alpha_t(i)$,$\beta_t(i)$进行迭代计算,直到前、后两次参数的期望值相差不超过 1%则结束迭代,即得到从历史数据中训练的隐马尔可夫模型参数.

3 系统的失效恢复

通过上述对系统状态的建模与分析,我们可以做出对于系统未来运行状态的判断,从而可以自适应地调整系统备份和系统服务计算之间的资源分配.本节将介绍系统失效恢复的过程以及如何进行动态的资源调整.

3.1 系统的失效恢复过程

在虚拟化环境下,一个典型的失效恢复架构如图 5 所示.网络中有多台物理主机,每台物理主机上运行着多个虚拟主机,这些虚拟主机通过网络共享磁盘的方式将所有存储存放在一个存储共享网络(SAN)上.对于需要失效恢复的虚拟机(以下称为主节点),在另外一台物理机上有它的一个运行状态备份(以下称为备份节点),备份节点按照一定的频率备份主节点的运行状态,当主节点失效时,备份节点自动启动以代替主节点来继续进行计算任务.因为采用共享磁盘的备份方式,因此主要的备份过程就是内存数据的重复复制过程.

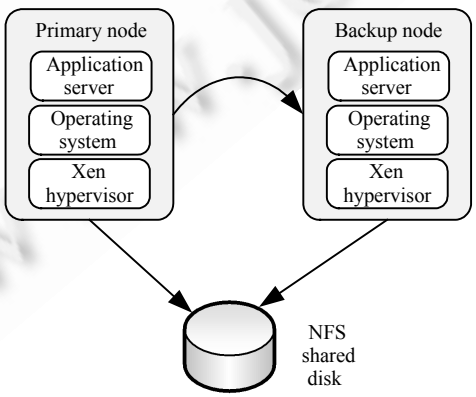


Fig.5 Failure recovery architecture

图 5 失效恢复架构

内存的备份过程是一个在主节点生成检查点、然后将检查点备份到备份节点的过程.这个过程类似于虚拟机的在线迁移[9−11],其区别在于:虚拟机的在线迁移在复制完检查点后会停止主节点,并切换到备份节点,从而完成迁移;而本文的备份过程是一个持续的备份过程,在完成一次检查点的备份后,主节点继续运行,备份节点则处于等待下一次的备份过程.如果主节点在此过程中出现异常停机,则备份节点开始替代主节点执行计算任务.

由于内存数据在两次备份间隔过程中只有部分发生了变化,因此备份过程仅需要复制这些变化的内存数据到备份节点.Xen 虚拟化环境提供了一种影子页表的机制,通过标识页表的保护位,使得虚拟机在进行内存读写的时候引发中断,从而使程序执行流程陷入 Xen 虚拟化管理层,由虚拟化管理层来标记内存的已更改信息.然后,在每次备份的过程中复制这些变化的内存数据到备份节点.

3.2 资源的动态调整

在虚拟机的失效恢复过程中,对主节点系统状态的备份过程是一个非常消耗系统资源的过程.大量的对内存状态更改的记录、备份和网络传输过程占用了正常业务系统所需的计算资源.因此,这里对资源的动态调整是通过调节失效恢复过程中的备份频率来完成的.

首先,通过上一节中给出的系统的隐马尔可夫模型,基于系统当前的运行状态,做出对系统未来运行状态的概率预测结果:如果预测结果显示系统进入异常状态的概率小于阈值,则降低系统状态的备份频率,从而降低系统状态备份过程中的资源占用,提高正常业务系统的服务能力;如果预测结果显示系统进入异常状态的概率高于阈值,即系统有很大的可能即将发生错误导致失效,则提高系统状态的备份频率,使得系统的运行状态能够尽可能快地备份到备份节点,从而在系统失效时能够较快地恢复系统状态,降低系统失效时的数据损失.

4 实 验

本文所述方法已在基于 Xen[12]的虚拟化环境下进行了原型实现.本节我们将通过实验来验证分析本文所提出的方法(下称 Hefery)的有效性以及其引入的系统运行时性能开销.

4.1 实验部署环境

实验的硬件部署环境如图 6 所示.

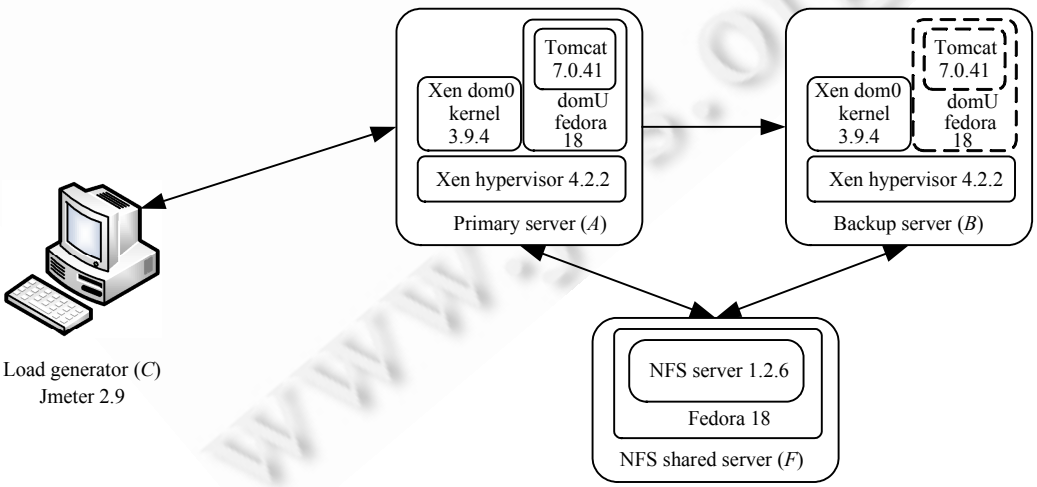


Fig.6 Experimental system

图 6 实验系统

在图 6 中,服务器 *A* 作为主节点运行 Xen 服务程序,其配置了 Intel Core i7 CPU 3.40 Ghz,4 G 内存以及千兆网卡,服务器 *B* 作为备份节点用于接收和保存服务器 *A* 发送过来的备份信息.服务器 *B* 同样配置了 Intel Core i7

CPU 3.40 Ghz,4 G 内存以及千兆网卡.服务器 *F* 作为 NFS 文件服务器,用于存储虚拟机的镜像文件,并同时共享给服务器 *A* 和服务器 *B*,其硬件配置为 Intel Core2 CPU 2.83 GHz,4 G 内存以及千兆网卡.另外有一台负载生成的客户端节点 *C*,用于产生对系统的压力负载.所有的服务器节点都通过一个千兆的交换机连接在同一个局域网中.

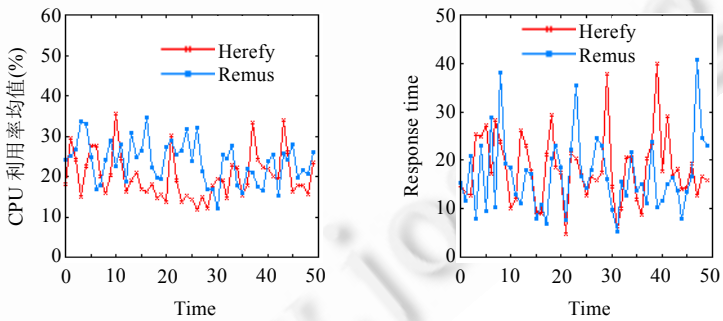
软件环境的部署如下:服务器 *A* 和服务器 *B* 部署了同样的 Xen 4.2.2 虚拟化运行环境,并采用了 Linux kernel 3.9.4 的 dom0 作为 Xen 的监控服务程序;服务器 *D* 则部署了 Fedora 18 Linux 操作系统,并安装了 NFS 网络文件服务系统用于提供共享文件服务;在主节点运行的 domU 则安装了 Fedora 18 Linux 操作系统,并在其上安装了 Tomcat 7.0.41 应用服务程序.客户端节点 *C* 部署了 Windows 操作系统,并运行 JMeter 压力生成程序来产生对服务器 *A* 的压力负载.

4.2 结果分析

4.2.1 系统开销

Herefy 方法在动态分析系统状态、调整系统的资源分配、提高系统的服务能力的同时,也会引入一定的开销.主要是因为 Herefy 方法需要实时地获取系统状态,并根据当前获取的状态信息来计算未来的系统状态的概率趋势.下面将通过实验量化分析 Herefy 方法的开销大小.

实验设置系统的备份间隔为 25 ms,并分两种情况进行分析:服务器 *A* 节点空载和服务器 *A* 节点满载.在系统空载时,即系统不运行任何对外服务程序,仅测试 Herefy 本身所引入的系统开销.测试结果如图 7(a)所示,服务器 *A* 节点在不使用 Herefy 的情况下,CPU 利用率均值为 20%,而在使用了 Herefy 的情况下,CPU 的利用率均值为 23%.Herefy 引入的 CPU 开销基本保持在 3%以内,因此,其引入的资源开销并不会对系统的运行造成较大的影响.另外,在满载的情况下,我们在虚拟机内运行一个 Tomcat 应用服务器,并通过 Jmeter 产生负载使得服务器 CPU 处于 100%的忙碌状态,然后测试 Herefy 对 Web 请求的响应时间的影响.实验结果如图 7(b)所示:在不使用 Herefy 的情况下,服务器的请求响应时间平均为 17.02 s;在使用了 Herefy 的情况下,服务器响应时间平均为 18.29 s.因此在系统满载的情况下,Herefy 方法对 Web 应用的请求响应时间的影响在 7%左右.



(a) No load (b) High load

Fig.7 System overhead

图 7 系统开销

4.2.2 隐马尔可模型预测效果

本实验的测试目的是验证系统的隐马尔可夫模型对系统运行状态趋势的预测能力.实验通过隐马尔可夫模型实时地对系统当前状态进行预测分析,输出系统运行状态的预测结果.图 8 给出了隐马尔可夫模型对系统状态的预测概率输出.从图中可以看出:在时间点 160 左右,当业务系统将要出现异常状态时,隐马尔可夫模型的预测概率输出明显变高,可以较好地给出业务系统未来的运行趋势.

另外,在时间点 20 左右,隐马尔可夫模型的输出概率也出现了小幅的升高.由于隐马尔可夫模型是通过分析计算系统的运行状态的时间序列来预测未来的状态的,所以,当业务系统运行的状态出现了类似于业务系统

异常状态下的行为特征序列时,模型的预测结果也会出现小幅的升高.对于这种问题,首先,因为业务系统在异常状态下是具有特定的状态序列的,正常状态下的业务系统的状态序列很难与异常状态下的业务系统状态序列吻合,因此,隐马尔可夫模型的预测输出结果的增幅会有较大的区别,我们可以通过设置一个阈值的方式来区分这两种不同的状态;其次,在正常的业务状态被判定为可能出现异常的情况下,会导致系统分配更多的资源来保证系统的可靠性,除了损失部分系统性能以外,不会产生不良影响.因此,这里我们倾向于把这个阈值设置为一个较低的值,从而可以捕获更多的业务系统异常状态而不会对系统产生不良影响.

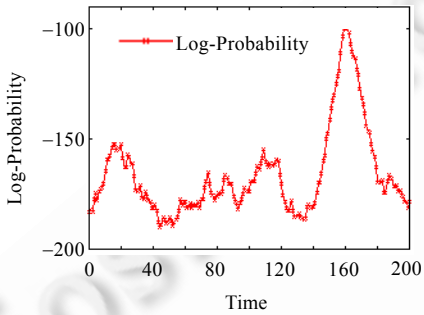


Fig.8 HMM output of forecast

图 8 隐马尔可模型预测输出

4.2.3 方法的有效性

本实验的目的是,通过使用和不使用 Hefery 方法来验证 Hefery 方法对系统运行时服务能力的改进的有效性.测试过程中,服务器 *A* 上运行 Xen 虚拟化环境,并启动了一个部署了 Tomcat 的虚拟机对外提供 Web 服务.在服务器 *A* 运行的过程中,其系统状态通过 Remus 方法备份到服务器 *B* 节点上.负载生成客户端产生足够的压力负载,使得部署 Tomcat 的虚拟机可以满负荷运行.

实验结果如图 9 所示.在不使用 Hefery 方法的情况下,由于 Remus 备份机制导致系统大量的资源消耗,使得系统的平均响应时间大约在 15s 左右;而在使用了 Hefery 方法的情况下,系统在健康状态正常时,动态地降低了 Remus 的备份频率及其资源消耗,使得系统的响应时间降低了 2/3,保持在 5s 以内,从而提高了系统的服务能力.在 Herefy 检测到系统的异常状态时,为了达到较高的系统可靠性和快速恢复能力,其动态调整 Remus 的备份频率及其资源消耗,使得更多的系统资源用于系统的保护,此时,系统的响应时间与不使用 Herefy 方法的情况下基本持平.

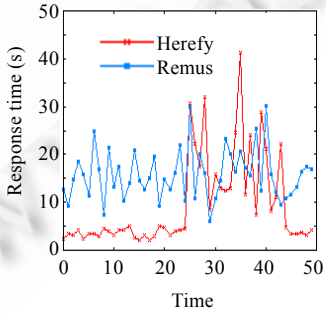


Fig.9 System response time comparison

图 9 系统响应时间对比

4.3 讨 论

上述实验验证了本文所提出的方法能够有效地对业务系统的异常状态进行预测,并根据预测结果对系统

的资源进行动态的调整和分配,使得业务系统的服务能力得到优化和提升.同时,实验结果表明:本文提出的方法所引入的开销较小,可以较好地保障业务系统的对外服务能力.

本文工作仍存在不足之处,虽然通过对系统运行状态的预测和动态调整系统的资源分配提高了系统的服务能力,但在系统发生异常并进行恢复的过程中会产生数据丢失的问题,即,在系统的两次备份间隔中还没有来得及完成备份的数据会丢失,从而会导致系统的异常恢复对用户是可察觉的.对于未来的下一步工作,我们希望通过缓存这部分数据来做到异常恢复过程对用户的透明性.主要包括两个方面:首先,在系统备份同步的间隔中,前端路由节点记录当前系统的所有外部用户的请求信息;然后在系统失效恢复后,通过在备份节点重放的方法来尽可能地恢复系统备份间隔中丢失的数据.

5 相关工作

近年来,云计算得到越来越广泛的应用,如 Amazon 的弹性云计算平台 EC2[13]、Google 的 App Engine[14]等.与此同时,云计算平台频繁的宕机事件[4−6]也使得虚拟化环境下的快速失效恢复问题成为工业界和学术界关注的焦点.

为了解决虚拟化环境下的服务器失效恢复问题,已经有大量的相关工作进行了深入的研究.它们大致可以分为两类:

- 一类是通过执行过程重放的方法来解决.如 Scales 等人[8]通过日志重放的方法实现了虚拟机的远程快速恢复,其在主节点虚拟机运行的同时,记录虚拟机的执行日志信息,并通过网络发送到备份节点,备份节点接收虚拟机的执行信息后,在本地通过重放还原主节点虚拟机的系统状态,从而使得主节点失效时可以快速切换到备份节点.
- 另一类是通过状态复制的方法解决.如 Cully 等人[7]实现了一种基于系统状态复制的运行时系统失效恢复方法,它通过实时地把系统 CPU 和内存状态备份到另一台服务器上来实现服务器的失效恢复能力.由于其每隔 25 ms 做一次系统状态的备份,有大量数据需要复制到备份节点上,因此性能开销较大,最高开销可达 75%.

由于上述方法的性能开销都比较大,因此另外一些工作对虚拟机的失效恢复的性能开销问题进行了优化研究.比如:

- Zhu 等人[15]通过减少检查点复制时的读失效和提高写失效的预测来提高系统状态备份的效率;
- Gerofi 等人[16]通过比较需要传输的内存的相似性,提取出其中的更新内容进行压缩后传输,从而达到提高备份传输效率的目的;
- Lu 等人[17]则提出了一种投机的内存传输策略,即,在基于时代的复制策略中提前传输部分内存到备份节点,从而提高传输效率.

这些方法都是针对虚拟机备份过程中的传输数据的优化,优化效果有限,虚拟机的失效恢复开销仍然较大.

隐马尔可夫模型作为一种有效的预测模型,在股票价格预测、DNA 序列的预测和机械寿命预测等方面有较多的研究工作.如,文献[18,19]通过对股票市场的时间序列时间进行分析,建立股票价格变化趋势的隐马尔可夫模型,从而对当前观测的股票价格变化信息进行计算分析,以预测未来股票的价格走势;文献[20,21]通过建立 DNA 序列的隐马尔可夫模型来对 DNA 序列中的蛋白质编码区进行预测分析.除了上述领域以外,隐马尔可夫模型也被用于其他领域,如,Yi 等人[22]使用隐半马尔可夫模型对网络用户浏览行为进行异常检测,其通过对网络页面的超链接特征和用户访问的特征建立模型来对用户浏览行为进行检测分析.本文则根据虚拟化环境的具体情况建立了基于隐马尔可夫模型的预测框架,通过对系统状态的预测分析,动态地调整虚拟机运行时的资源分配,从而降低系统因失效恢复功能而引入的较高的系统开销.

6 结束语

虚拟机的失效恢复机制是保障虚拟机可靠运行的重要方法,但是由于虚拟化环境的特殊性,对整个虚拟机

状态的复制备份过程是一个相当耗费资源的过程.传统的研究思路主要集中在对备份过程中产生的数据进行优化的方面,通过压缩或者去除冗余数据的方式来降低系统的开销;本文则提出了一种基于隐马尔可夫模型,对系统健康状态进行预测,动态地调整系统备份频率的方法,有效地降低了系统正常状态下的性能开销问题,使得更多的系统资源可以用于有效计算,从而提高用户的响应时间.

References:

[1] Chen K, Zheng WM. Cloud computing: System instances and current research. Ruan Jian Xue Bao/Journal of Software, 2009,20(5): 1337−1348 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3493.htm [doi: 10.3724/SP.J.1001.2009.03493]

[2] Rosenblum M, Garfinkel T. Virtual machine monitors: Current technology and future trends. Computer, 2005,38(5):39−47. [doi: 10.1109/MC.2005.176]

[3] Kivity A, Kamay Y, Laor D, Lublin U, Liguori A. KVM: The Linux virtual machine monitor. Proc. of the Linux Symp., 2007,1(1): 225−230.

[4] Amazon ec2 outage downs reddit, quora. 2011. http://money.cnn.com/2011/04/21/technology/amazon_server_outage/

[5] Extended gmail outage hits apps admins. 2008. http://www.computerworld.com/s/article/9117322/Extended_Gmail_outage_hits_Apps_admins

[6] Flexiscale suffers 18-hour outage. 2008. http://www.thewhir.com/web-hosting-news/flexiscale-suffers-18-hour-outage

[7] Cully B, Lefebvre G, Meyer D, Feeley M, Hutchinson N, Warfield A. Remus: High availability via asynchronous virtual machine replication. In: Proc. of the 5th USENIX Symp. on Networked Systems Design and Implementation. USENIX Association, 2008. 161−174.

[8] Scales DJ, Nelson M, Venkitachalam G. The design of a practical system for fault-tolerant virtual machines. SIGOPS Operating System Review, 2010,44(4):30−39. [doi: 10.1145/1899928.1899932]

[9] Clark C, Fraser K, Hand S, Hansen JG, Jul E, Limpach C, Pratt I, Warfield A. Live migration of virtual machines. In: Proc. of the 2nd Conf. on Symp. on Networked Systems Design & Implementation. USENIX Association, 2005. 273−286.

[10] Liu HK, Jin H, Liao XF, Hu L, Yu C. Live migration of virtual machine based on full system trace and replay. In: Proc. of the 18th ACM Int'l Symp. on High Performance Distributed Computing. New York: ACM Press, 2009. 101−110. [doi: 10.1145/1551609.1551630]

[11] Jin H, Deng L, Wu S, Shi XH, Pan XD. Live virtual machine migration with adaptive, memory compression. In: Proc. of the Cluster Computing and Workshops. 2009. 1−10. [doi: 10.1109/CLUSTR.2009.5289170]

[12] Barham P, Dragovic B, Fraser K, Hand S, Harris T, Ho A, Neugebauer R, Pratt I, Warfield A. Xen and the art of virtualization. In: Proc. of the 19th ACM Symp. on Operating Systems Principles. New York: ACM Press, 2003. 164−177.

[13] Amazon elastic compute cloud (amazon ec2). 2012. http://aws.amazon.com/ec2/

[14] Google app engine. 2012. http://developers.google.com/appengine

[15] Zhu J, Jiang ZF, Xiao Z, Li XM. Optimizing the performance of virtual machine synchronization for fault tolerance. IEEE Trans. on Computers, 2011,60(12):1718−1729. [doi: 10.1109/TC.2010.224]

[16] Gerofi B, Vass Z, Ishikawa Y. Utilizing memory content similarity for improving the performance of replicated virtual machines. In: Proc. of the 2011 4th IEEE Int'l Conf. on Utility and Cloud Computing. Washington: IEEE Computer Society, 2012. 73−80. [doi: 10.1109/UCC.2011.20]

[17] Lu MH, Chiueh T. Speculative memory state transfer for active-active fault tolerance. In: Proc. of the 2012 12th IEEE/ACM Int'l Symp. on Cluster, Cloud and Grid Computing. Washington: IEEE Computer Society, 2012. 268−275. [doi: 10.1109/CCGrid.2012.37]

[18] Hassan MR, Nath B. Stock market forecasting using hidden Markov model: A new approach. In: Proc. of the Intelligent Systems Design and Applications. 2005. 192−196. [doi: 10.1109/ISDA.2005.85]

[19] Zhang YJ. Prediction of financial time series with hidden Markov models [Ph.D. Thesis]. Simon Fraser University, 2004.

[20] Yin MM, Wang JTL. Application of hidden Markov models to gene prediction in DNA. In: Proc. of the Information Intelligence and Systems. 1999. 40−47.

[21] Luo ZJ, Song LH. DNA information mining based on hidden Markov models. In: Proc. of the 6th Int'l Conf. on Natural Computation. 2010. 238−241. [doi: 10.1109/ICNC.2010.5582898]

[22] Yi X, Zheng YS. Anomaly detection based on Web users' browsing behaviors. Ruan Jian Xue Bao/Journal of Software, 2007,18(4): 967−977 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/18/967.htm [doi: 10.1360/jos180967]

附中文参考文献:

[1] 陈康,郑纬民.云计算:系统实例与研究现状.软件学报,2009,20(5):1337−1348. http://www.jos.org.cn/1000-9825/3493.htm [doi: 10.3724/SP.J.1001.2009.03493]

[22] 谢逸,余顺争.基于 Web 用户浏览行为的统计异常检测.软件学报,2007,18(4):967−977. http://www.jos.org.cn/1000-9825/18/967.htm [doi: 10.1360/jos180967]

张建华(1983－),男,山东青岛人,博士生,主要研究领域为网络分布式计算,软件工程.
E-mail: jhzhang@otcaix.iscas.ac.cn

魏峻(1970－),男,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为网络分布式计算,软件工程.
E-mail: wj@otcaix.iscas.ac.cn

张文博(1976－),男,博士,副研究员,CCF 会员,主要研究领域为网络分布式计算,软件工程.
E-mail: zhangwenbo@otcaix.iscas.ac.cn

钟华(1971－),男,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为网络分布式计算,软件工程.
E-mail: zhongh@otcaix.iscas.ac.cn

徐继伟(1985－),男,博士生,主要研究领域为网络分布式计算,软件工程.
E-mail: xujiwei10@otcaix.iscas.ac.cn

黄涛(1965－),男,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为网络分布式计算,软件工程.
E-mail: tao@otcaix.iscas.ac.cn