Journal of System Simulation

Volume 29 | Issue 8 Article 8

6-1-2020

Performance Modeling of Cryptographic Service System Virtualization Based on ISSM

Songhui Guo
1. PLA Information Engineering University, Zhengzhou 450001, China;;2. State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China;

Qingbao Li
1. PLA Information Engineering University, Zhengzhou 450001, China;;2. State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China;

Sun Lei
1. PLA Information Engineering University, Zhengzhou 450001, China;;

Xuerong Gong
1. PLA Information Engineering University, Zhengzhou 450001, China;;

See next page for additional authors

Performance Modeling of Cryptographic Service System Virtualization Based on ISSM

Abstract

Abstract: The complicated architecture of cryptographic service system virtualization raised the difficulty of performance modeling. *A performance modeling approach based on ISSMs was proposed. The approach divided the execution process into two stages, host preprocessing and arithmetic-module calculating, and built two sub-models based on queuing theory. On this basis, the effectiveness of this approach was verified.* The results show that this method can analyze the impacts on system performance caused by task arrival rates, host and cryptographic card configurations quantitatively, and also be helpful for providing reasonable solutions to deploy virtualized cryptographic service system on cloud computing platforms.

Keywords

cryptographic service system, virtualization, performance modeling, response time, cloud computing, blocking probability

Authors

Songhui Guo, Qingbao Li, Sun Lei, Xuerong Gong, and Tianchi Yang

Recommended Citation

Guo Songhui, Li Qingbao, Sun Lei, Gong Xuerong, Yang Tianchi. Performance Modeling of Cryptographic Service System Virtualization Based on ISSM[J]. Journal of System Simulation, 2017, 29(8): 1692-1701.

第 29 卷第 8 期 系统仿真学报© Vol. 29 No. 8
2017 年 8 月 Journal of System Simulation Aug., 2017

基于 ISSM 的密码服务系统虚拟化性能建模

郭松辉[1,2]，李清宝[1,2]，孙磊[1]，龚雪容[1]，杨天池[1]
（1. 解放军信息工程大学，郑州 450001；2. 数学工程与先进计算国家重点实验室，郑州 450001）

摘要：针对密码服务系统虚拟化结构复杂导致性能建模难度大的问题，提出了*一种基于交互随机子模型(Interactive Stochastic Sub-Models, ISSMs)的性能建模方法，将任务执行过程划分为主机预处理和运算单元执行两个阶段，并基于排队论分别建立了两个子模型。在此基础上对密码服务系统虚拟化性能模型的有效性进行了验证。*结果表明，所建模型能够定量分析任务的到达速率、主机配置与密码卡配置对系统性能的影响，该模型同时也有助于指导云计算环境下密码服务系统虚拟化部署方案的设计。

关键词：密码服务系统；虚拟化；性能建模；响应时间；云计算；拒绝概率
中图分类号：TP391.9 文献标识码：A 文章编号：1004-731X (2017) 08-1692-10
DOI: 10.16182/j.issn1004731x.joss.201708008

Performance Modeling of Cryptographic Service System Virtualization Based on ISSM

Guo Songhui[1,2], *Li Qingbao*[1,2], *Sun Lei*[1], *Gong Xuerong*[1], *Yang Tianchi*[1]
(1. PLA Information Engineering University, Zhengzhou 450001, China;
2. State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China)

Abstract: The complicated architecture of cryptographic service system virtualization raised the difficulty of performance modeling. *A performance modeling approach based on ISSMs was proposed. The approach divided the execution process into two stages, host preprocessing and arithmetic-module calculating, and built two sub-models based on queuing theory. On this basis, the effectiveness of this approach was verified.* The results show that this method can analyze the impacts on system performance caused by task arrival rates, host and cryptographic card configurations quantitatively, and also be helpful for providing reasonable solutions to deploy virtualized cryptographic service system on cloud computing platforms.

Keywords: cryptographic service system; virtualization; performance modeling; response time; cloud computing; blocking probability

引言

云计算因其资源共享、弹性配置、按需服务和位置无关等优点，发展成为一种重要的新兴计算模式。云计算通过整合分布式资源，构建能够满足多种服务要求的计算环境，满足用户定制化要求，降低用户计算和存储的使用与维护成本[1]。

随着云计算在军事、电子商务和电子政务等领域的应用和推广，面临为敏感信息提供机密性、完整性和真实性等保护的迫切需求。通过部署密码服务器，由其中的密码服务系统提供信息保护，是目前广泛应用的密码服务保障模式[2]。由于用户需求的多样性，需要部署多种类型密码

收稿日期：2015-11-02 修回日期：2016-01-31；
基金项目：国家自然科学基金(61072047)，国家 863 计划 (2012AA012704)；
作者简介：郭松辉(1979-)，男，四川仁寿，博士，研究方向为云计算、虚拟化、性能建模；李清宝(1967-)，男，四川乐山，博士，教授，博导，研究方向为云计算、系统结构。

服务器，提供不同的密码服务功能，如：信息的对称加解密保护、用户身份认证与授权等。通过对密码服务系统的虚拟化，将多台密码服务器由多个物理主机聚合到单台物理主机，构建为更易管理和维护的虚拟化密码服务系统，能够有效提高密码运算资源的利用效率[3]，增强系统灵活性。

云计算在为用户带来便利的同时，其自身结构的复杂性，也为提供密码服务功能带来了新的挑战：为用户提供既能保障服务质量(Quality of Service，QoS)而又最小化资源成本的密码服务，具有较大的难度[4]。对于用户而言，服务的可用性与响应时间是 QoS 的两项关键指标，通过对可用性和响应时间进行性能建模[5]，能够进行量化和特征分析[6]。

排队论是一种性能指标形式化建模和量化分析的有效方法[7]，通过对研究对象到达及服务特征的统计研究，得出这些数量指标(排队时间、拒绝概率等)的统计规律，然后根据这些规律来改进服务系统结构或部署，使得服务系统既能满足用户需求，又能使系统成本或部分指标最优。大量学者基于排队论对云计算性能建模方法进行了研究。Yigitbasi N 等人[8]研究了建立全局度量框架，采用实验方法对响应时间的多种指标进行度量，这种方法的主要不足是当系统较为复杂时难以获取系统的完整特征，实验分析的准确度不高。Salah K 等人[9]对弹性云中的虚拟机数量和各个虚拟机的性能进行建模分析，得到系统的平均延迟，在各虚拟机处理的任务差异较大时，会导致模型失真。Mytilinis I 等人[10]针对云计算的 I/O 密集型应用，提出了一种端到端的性能建模方法，能够对 I/O 和任务负载同时进行预测，但不能解释应用内部的执行行为，不能准确剖析系统内部的性能表现。文献[11]研究表明，当系统较复杂时，建立全局模型会引入大量参数，增加对模型进行分析的难度，因此，将全局模型拆分为多个子模型，采用回归分析等手段，能够达到简化分析难度提高分析准确率的目的，但文献中只给出了服务请求为特定的单个或多个按指数分布时间间隔到达的服务请求，并未对划分方法进行详细的分析讨论。文献[12]针对复杂服务请求，通过建立混合性能模型的性能测试框架，依据不同的性能指标，采取不同的性能建模方法进行测试，其度量准确性、预测有效性均较为优异，但对于密码服务等度量指标相对单一的系统，采用混合建模成本较高。

本文在综合分析已有研究成果的基础上，提出了一种基于ISSM的性能建模方法，该方法通过分析密码服务系统的任务执行过程，将任务执行过程划分为主机预处理和运算单元执行两个阶段，从而为复杂的云计算服务建立分立子模型，通过对子模型进行交互建模分析来获得整个系统的性能解决方案。

1 研究对象描述与分析

1.1 密码服务系统虚拟化

密码服务系统虚拟化的结构如图 1 所示，主要包括虚拟机监视器(Virtual Machine Monitor，VMM)、密码域、设备域和底层的密码卡、CPU、内存等硬件。其中，虚拟机监视器是实现对底层硬件资源抽象的软件层，虚拟机监视器对虚拟机(Virtual Machines，VMs)进行管理，实现虚拟机对物理资源的共享[13]；部署密码服务功能的虚拟机称为密码域，负责管理和调度密码卡的虚拟机称为设备域。在密码服务系统的虚拟化中，各密码域通过虚拟机监视器的管理和调度，实现对主机硬件资源的复用，如 CPU、内存和密码卡等。

1.2 模型分析

如图 1 所示，在密码服务系统虚拟化中，密码卡属于核心设备，为方便管理和提高系统稳定性，需要建立专门的设备域对其进行管理和调度[14]。密码域将运算任务提交给设备域，设备域再将任务

https://dc-china-simulation.researchcommons.org/journal/vol29/iss8/8
DOI: 10.16182/j.issn1004731x.joss.201708008

第 29 卷第 8 期 系统仿真学报 Vol. 29 No. 8
2017 年 8 月 Journal of System Simulation Aug., 2017

提交给密码卡进行密码运算。设备域负责密码服务的调度和管理，密码域与设备域之间进行数据传递时，需要在内存中交换数据，当任务调度较为频繁、需要交换的数据量较小时，由于存在频繁的上下文切换[15]，易成为影响密码服务系统性能的主要瓶颈，大量学者对此进行了专门研究[16-19]，如通过负载平衡调节、调度优化等措施提高数据的传输性能。在满足各个任务延迟要求的前提下，将虚拟密码域的多个任务聚合为单个任务包(Task Packet，TP)再进行排队和处理，能够减少内存授权、通道中断和数据拷贝等需要大量时间开销的请求。相比于多个小数据量任务，系统具有更低的单字节延迟，综合性能和处理效率更高。因此，系统将小任务聚合成大任务的处理方法，即在各密码域中建立缓冲区，将多个小任务聚合为单个任务包，在时间窗口内进行任务包的传递和处理，减少了虚拟化密码服务系统中数据传递的通道准备时间、中断现场保存时间等较为耗时的环节，能够大幅提高系统性能。同时，由于域间任务特征差异性导致任务包处理时间不同，通过对非虚拟化环境下密码运算业务时间采集分析，并以此为依据调整各密码域内的包聚合长度，使各密码域任务包的处理时间趋于一致，能够降低任务处理的延迟抖动，有效提高系统稳定性。

由于在云计算环境中密码服务系统存在规模大、复杂度高，且与云计算基础设施相关联的特点，对其在云中的服务进行分析建模较为复杂。如果对模型进行不恰当的简化，对 QoS 具有重要影响的参数被漏掉，将导致模型失真。为解决该问题，本文提出了一种基于 ISSM 的模型分析方法，并依据提供服务的步骤，为复杂的密码服务系统虚拟化建立分立子模型，通过对子模型进行交互分析来获得整个系统的性能解决方案，如图 2 所示。

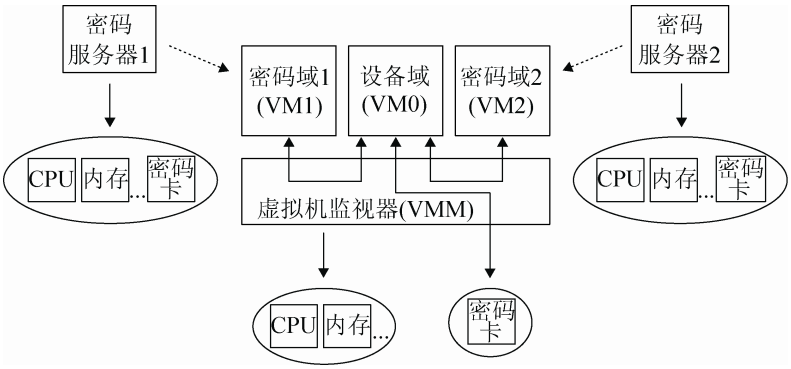


图 1 密码服务系统虚拟化的结构
Fig. 1 The architecture of virtualized cryptographic service system

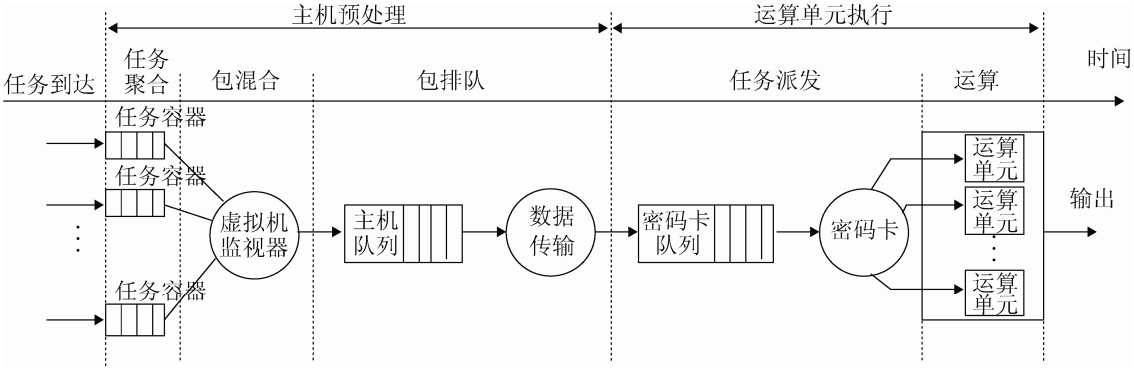


图 2 服务步骤与对应延迟
Fig. 2 The service steps and delay

系统响应时间延迟主要由两部分组成：一部分是主机预处理阶段的任务聚合与传递，另一部分是密码卡调度自身运算单元执行密码运算任务。在主机预处理阶段，密码域对密码服务请求进行前期处理，首先将多个小数据量任务聚合到单个任务包，当任务包中的任务数量或任务包聚合时间达到阈值时，将任务包提交给设备域，设备域将各个密码域提交的任务包进行排队，提交密码卡处理。在密码卡调度运算单元执行阶段，密码卡作为主(MASTER)设备在主机中运行，与设备域的通信均由其主动发起，密码卡通过读/写主机内存与主机交换数据。由于查询操作能够获得比中断更小的上下文保存开销，密码卡采用查询方式获取主机任务队列信息，查询速率等参数均由密码卡控制。密码卡中部署有多个硬件运算单元，当密码卡获取到任务包后，调度硬件运算单元执行密码运算。

本文设计的两个随机子模型：主机预处理子模型(Host Preprocessing Sub-model，HPSM)和运算单元执行子模型(Arithmetic-module Calculating Sub-model，ACSM)，分别用来分析主机预处理和密码运算单元执行两个阶段的详细特征。通过所建模型可分析得到负载与系统配置变化对性能指标和系统能力的影响。由于运算单元完成密码运算后，将运算结果返回各密码域的过程与运算任务输入过程相似且时间开销基本一致，故本文不再对其进行重复的分析讨论。

2 系统建模

2.1 主机预处理子模型(HPSM)

在密码域中建立任务容器，用于聚合任务包，新产生的密码运算任务进入任务容器时，在容器末尾排队，当容器中任务数量达到最大值或包聚合时间达到阈值时，密码域将任务包提交给设备域。设备域调用密码卡完成运算后，将运算结果返回给密码域。

设第 i 个密码域中容器容量为 k_i，运算任务的到达时间间隔服从指数分布，运算任务的到达速率为 λ_i，则 k_i 个服从指数分布的独立随机变量的运算任务，服从参数为 λ_i 和 k_i 的 Erlang 分布，且 N 个密码域的任务包到达设备域共享内存的过程是 N 个 Erlang 过程 $Er(\lambda_i,k_i),i=1,\ldots,N$ 的混合分布(a Mixture of Generalized Erlang distribution，MGE) [20]。

2.1.1 任务包到达过程分析

Erlang 分布是 PH 分布的一种特殊形式[20]，随机变量的 m 阶 PH 分布表示状态集 $\{1,2,\ldots,m-1,m\}$ 上的 Markov 过程吸收时间的分布，其分布函数为 $F(t)=1-\boldsymbol{\alpha}\exp(\boldsymbol{T}t)\boldsymbol{1}$，$m$ 维行向量 $\boldsymbol{\alpha}$ 是 Markov 链的初始分布，$\boldsymbol{T}$ 是无穷小生成元矩阵，$\boldsymbol{1}$ 为 m 维元素全为 1 的列向量。每个用来建模任务包到达过程的 Erlang 过程 $Er(\lambda_i,k_i),i=1,2,\ldots,N$，可以用 $(\boldsymbol{\alpha}_i,\boldsymbol{T}_i),i=1,2,\ldots,N$ 来表示，其中 $\boldsymbol{\alpha}_i$ 是 k_i 维的行向量，$\boldsymbol{T}_i$ 是 k_i*k_i 维的生成元矩阵：

$$\boldsymbol{\alpha}_i=(1\ 0\ \ldots\ 0),\ i=0,1,\ldots,N \tag{1}$$

$$\boldsymbol{T}_i=\begin{pmatrix} -\lambda_i & \lambda_i & 0 & \cdots & 0 \\ 0 & -\lambda_i & \lambda_i & \cdots & 0 \\ 0 & 0 & -\lambda_i & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -\lambda_i \end{pmatrix},\ i=0,1,\ldots,N \tag{2}$$

由于 PH 分布具有封闭性，N 个参数为 $(\boldsymbol{\alpha}_i,\boldsymbol{T}_i)$，$i=1,2,\ldots,N$ 的 PH 过程的混合，仍然服从 PH 分布，有 PH 表示 $[\boldsymbol{\alpha},\boldsymbol{T}]$ [20]，$\boldsymbol{\alpha}$ 表示初始概率向量，$\boldsymbol{T}$ 表示到达过程瞬时状态的生成元。

2.1.2 任务包服务过程分析

主机中任务包的服务过程，即为密码卡读取任务包的过程。设密码卡按指数时间间隔读取任务包，速率为 λ_{TP}，则主机中任务包的服务时间间隔服从参数为 μ_h 的指数分布，$\mu_h=\lambda_{TP}$。密码域容器容量为 k_i，对包含密码域、设备域和共享内存的虚拟化密码服务系统，可以用 PH/M/1 排队模型建模。模型由单 FIFO 服务台构成，具有 PH 到达过程和指数服务时间。分析该排队模型并对其性能进行评估，需要使用基于矩阵几何解的分

第 29 卷第 8 期 系统仿真学报 Vol. 29 No. 8
2017 年 8 月 Journal of System Simulation Aug., 2017

析方法和拟生灭过程(Quasi Birth and Death processes，QBD)。下面对主机预处理子模型中使用 PH/M/1 排队的 QBD 过程进行分析。

2.1.3 基于 QBD 对 PH/M/1 排队系统建模

首先对到达过程为 PH 型分布、服务过程为指数分布的 PH/M/1 型排队系统进行分析。到达过程参数$(\boldsymbol{\alpha},\boldsymbol{T})$，阶为 m，$\boldsymbol{\alpha}$ 表示初始概率向量，$\boldsymbol{T}$ 表示到达过程瞬时状态的生成元，则 $\boldsymbol{\alpha 1}=1$，$\lambda^{-1}=-\boldsymbol{\alpha T}^{-1}\boldsymbol{1}$。PH/M/1 系统使用 Markov 过程 $\{(N(t), a(t):\ t\geqslant 0)\}$建模，其中 $N(t)$表示时刻 t 的任务包数，$a(t)$表示时刻 t 到达过程所处的状态，状态空间为 $\{(n,a): n\geqslant 0, 1\leqslant a\leqslant m\}$。主机预处理子模型的 Markov 过程是一个连续时间的 QBD，其无穷小生成元 $\boldsymbol{Q}$ 为：

$$\boldsymbol{Q}=\begin{pmatrix} \boldsymbol{T} & \boldsymbol{T}^0\boldsymbol{\alpha} & 0 & \cdots & 0 \\ \mu_h\boldsymbol{I} & -\mu_h\boldsymbol{I}+\boldsymbol{T} & \boldsymbol{T}^0\boldsymbol{\alpha} & \cdots & 0 \\ 0 & \mu_h\boldsymbol{I} & -\mu_h\boldsymbol{I}+\boldsymbol{T} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -\mu_h\boldsymbol{I}+\boldsymbol{T} \end{pmatrix} \tag{3}$$

则 HPSM 的稳态概率分布矩阵 $\boldsymbol{R}$(简称率阵)为非线性矩阵等式(4)的最小非负解：

$$\mu_h\boldsymbol{R}^2+\boldsymbol{R}(\boldsymbol{T}-\mu_h\boldsymbol{I})+\boldsymbol{T}^0\boldsymbol{\alpha}=\boldsymbol{0} \tag{4}$$

此过程的稳态分布是一个矩阵几何分布，将稳态概率向量 π 按水平表示为分段子向量：

$$\pi=(\pi_0,\pi_1,\ldots,\pi_n),\ \pi_k=(\pi_{k1},\ldots,\pi_{km}),\ k\geqslant 0 \tag{5}$$

其中 π_k 是阶 k 的子向量，满足：

$$\pi_k=\pi_0\boldsymbol{R}^k,\ k\geqslant 1 \tag{6}$$

稳态概率向量 $\boldsymbol{\pi}_0$ 通过求解式(6)边缘分布获得：

$$\pi_0\boldsymbol{T}+\mu_h\pi_1\boldsymbol{I}=0 \tag{7}$$

正规化条件为：

$$\sum_{k=0}^{\infty}\pi_k\boldsymbol{1}=\pi_0(\boldsymbol{I}-\boldsymbol{R})^{-1}\boldsymbol{1}=1 \tag{8}$$

式中：$\boldsymbol{I}$ 表示相应维数的单位阵，由式(7)、(8)可得到 π_0，从而获得全部概率向量。

2.1.4 排队模型 PH/M/1 的平均等待时间

任务在主机预处理时的平均等待时间 $E(T_{\text{wait\_host}})$ 为平稳状态下任意时刻到达任务的等待时间 $T_{\text{wait\_host}}$ 的一阶矩，$T_{\text{wait\_host}}$ 由下式得到：

$$T_{\text{wait\_host}}=1-\pi_1(\boldsymbol{I}-\boldsymbol{R})^{-1}\exp[-\mu_h(\boldsymbol{I}-\boldsymbol{R})x]\boldsymbol{1},\ x\geqslant 0 \tag{9}$$

平均等待时间为：

$$E(T_{\text{wait\_host}})=\frac{1}{\mu_h}\pi_0\boldsymbol{R}(\boldsymbol{I}-\boldsymbol{R})^{-2}\boldsymbol{1} \tag{10}$$

平均逗留时间为：

$$\overline{st_{\text{host}}}=\frac{1}{\mu_h}+E(T_{\text{wait\_host}}) \tag{11}$$

当运算单元执行子模型处于状态 n 时，密码卡拒绝服务，拒绝服务的概率为：

$$BP_{\text{host}}=\pi_n\boldsymbol{1}=\pi_0\boldsymbol{R}^n\boldsymbol{1} \tag{12}$$

2.2 运算单元执行子模型(ACSM)

密码卡中部署有多个硬件运算单元，密码域提交的任务包由密码卡调度空闲运算单元执行，如：杂凑运算、对称加密、对称解密、公钥签名和公钥验签等。密码卡在主机中作为主设备(MASTER)运行，密码卡通过读写主机内存中的任务队列完成与设备域交换数据，密码卡将任务取到卡内处理的读操作，以及将结果数据发送到主机内存的写操作，均服从指数分布，因此，可以用 M/M/n/N 排队模型进行建模[7]。子模型状态转移图如图 3 所示。

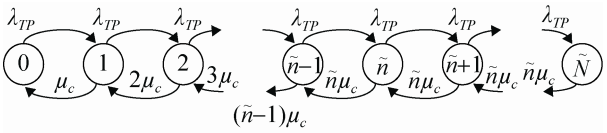


图 3 运算单元执行子模型状态转移

Fig. 3 The state transition of arithmetic-module calculating sub-model

图 3 中，运算单元数量为 $\tilde{n}$，密码卡内 FIFO 长度为 $\tilde{N}$，$\tilde{N}\geqslant\tilde{n}$，当任务包到达密码卡时，有以下三种可能的执行方法：

(1) 若有空闲的运算单元，则到达的任务包立即进入服务；

(2) 若 $\tilde{n}$ 个运算单元全部忙，且系统中排队的任务包数量小于 $\tilde{N}$，则该任务包排在队末等待服务；

(3) 若系统中已有 $\tilde{N}$ 个任务包，则表示密码

卡 FIFO 没有足够空间，拒绝任务包。

设 $1/\mu_c$ 为运算单元完成一个任务包中所有运算任务的平均服务时间。当任务包得到服务离开队列后，下一个排在队首的任务包即将获得服务。在该子模型中，任务包到达速率 λ_{TP} 由密码卡控制；密码域任务包长度依据处理时间进行设置，运算速率 $1/\mu_c$ 为密码卡运算单元固有特征参数，因此，λ_{TP}、$1/\mu_c$ 均为外部输入参数。模型的稳态概率满足：

$$\tilde{\pi}_1 = \frac{\lambda_{TP}}{\mu_c}\tilde{\pi}_0 \tag{13}$$

当系统状态 $i < \tilde{n}$ 时，

$$\begin{aligned}&\tilde{\pi}_i = \frac{\lambda_{TP}}{i\mu_c}\tilde{\pi}_{i-1} = \frac{1}{i!}\left(\frac{\lambda_{TP}}{\mu_c}\right)^i, i = 1,2,\ldots,\tilde{n}-1\\ &\tilde{\pi}_0 = \frac{(n\rho)^i}{i!}\tilde{\pi}_0\end{aligned} \tag{14}$$

当系统状态 $i \geqslant \tilde{n}$ 时，

$$\begin{aligned}&\tilde{\pi}_i = \frac{1}{\tilde{n}^{i-\tilde{n}}}\frac{1}{\tilde{n}!}\left(\frac{\lambda_{TP}}{\mu_c}\right)^i, i = \tilde{n},\tilde{n}+1,\cdots,\tilde{N}\\ &\tilde{\pi}_0 = \frac{(\tilde{n}\rho)^i}{\tilde{n}!\tilde{n}^{i-\tilde{n}}}\tilde{\pi}_0\end{aligned} \tag{15}$$

当系统处于平稳状态时，有：

$$\sum_{i=0}^{\tilde{N}}\tilde{\pi}_i = 1 \tag{16}$$

则：

$$\tilde{\pi}_0 = \left[\sum_{i=0}^{\tilde{n}-1}\frac{1}{i!}(\tilde{n}\rho)^i + \sum_{i=1}^{\tilde{N}-\tilde{n}}\frac{1}{\tilde{n}!\tilde{n}^i}(\tilde{n}\rho)^{i+\tilde{n}}\right]^{-1} \tag{17}$$

式中：$\rho = \lambda_{TP}/\tilde{n}\mu_c < 1$ 为系统利用率。

计算密码卡队列中的任务包平均逗留时间，首先需要得到系统到达平稳状态后系统中任务包的平均数，任务包平均数量为：

$$\overline{q} = \sum_{i=1}^{\tilde{N}} i\tilde{\pi}_i \tag{18}$$

当运算单元执行子模型处于状态 $\tilde{N}$ 时，密码卡拒绝服务，拒绝服务的概率为：

$$BP_{\text{card}} = \tilde{\pi}_{\tilde{N}} = \frac{\tilde{n}^{\tilde{n}}\rho^{\tilde{N}}}{\tilde{n}!}\tilde{\pi}_0 \tag{19}$$

任务包进入系统的概率为 $(1-\tilde{\pi}_{\tilde{N}})$，因此系统进入率为：

$$\overline{\lambda} = \lambda_{TP}(1-\tilde{\pi}_{\tilde{N}}) \tag{20}$$

根据 Little's 定律[20]，队列中的平均逗留时间 $\overline{st_{\text{card}}}$ 由下式得到：

$$\overline{st_{\text{card}}} = \frac{\overline{q}}{\overline{\lambda}} \tag{21}$$

因此，系统拒绝为用户提供密码服务的概率为：

$$P_{\text{reject}} = BP_{\text{host}} + BP_{\text{card}} \tag{22}$$

密码服务的平均单向延迟为：

$$\overline{st} = \overline{st_{\text{host}}} + \overline{st_{\text{card}}} \tag{23}$$

2.3 子模型交互关系

运算单元执行子模型的处理能力直接关系到密码服务系统虚拟化的整体处理能力。当运算单元执行能力较低时，会导致系统拒绝服务的概率增加；反之，当运算单元执行能力较高、而主机部署的虚拟密码域较少时，则不能充分发挥密码卡的性能，导致资源浪费。当密码服务系统硬件配置固定时，运算单元执行子模型的运算能力确定，通过对系统进行建模分析，能够为主机预处理子模型中虚拟密码域的合理配置提供科学依据，从而有效提高系统利用效率。

各密码域将任务包提交给设备域，由设备域在 FIFO 中进行排队；密码卡将存放任务队列的内存映射到自身 CPU 内存地址范围，按照指数间隔读取速率，将任务包提交给运算单元进行运算。因此，密码卡中任务包的到达过程，也是主机中任务包的服务过程，即 $\mu_h = \lambda_{TP}$，服从指数分布。

当虚拟化密码服务系统处于稳定状态时，主机预处理子模型的平均逗留时间 $\overline{st_{\text{host}}}$ 与运算单元执行子模型的平均逗留时间 $\overline{st_{\text{card}}}$（包括等待时间和服务时间）趋于一致，系统具有最大化效能，即：

$$\overline{st_{\text{host}}} \approx \overline{st_{\text{card}}} \tag{24}$$

由以上分析可知，子模型之间存在依赖关系。在系统硬件配置一定时，通过对密码卡读取任务包速率的调整，可以实现对密码服务系统的最大化利用。设：

$$\begin{cases}\overline{st_{\text{host}}} = f(x,y)\\ \overline{st_{\text{card}}} = g(x)\end{cases} \tag{25}$$

https://dc-china-simulation.researchcommons.org/journal/vol29/iss8/8
DOI: 10.16182/j.issn1004731x.joss.201708008

式中：x 表示主机预处理子模型中的任务包服务速率 μ_h，y 表示虚拟密码域数量 N_{VM}，$f(x,y)$ 为主机预处理子模型中的任务平均逗留时间函数，$g(x)$ 为运算单元执行子模型中的任务平均逗留时间函数。根据实际系统运行特征，选取 x_i 作为任务包服务速率初始值，利用式(24)、(25)即可得到 y_i。

设要求系统的拒绝服务概率不高于 p_{sys}，利用 (x_i,y_i) 可计算得到 BP_{host}、BP_{card}，如果满足 $BP_{\text{host}}+BP_{\text{card}}=P_{\text{reject}}\leqslant p_{\text{sys}}$，则 (x_i,y_i) 为符合系统要求的参数值，如果不符合，则根据 $\overline{st_{\text{host}}}$、$\overline{st_{\text{card}}}$ 的单调性，对 x_i 按一定的步长进行修正，对获得的参数值使用 $P_{\text{reject}}\leqslant p_{\text{sys}}$ 进行验证，直到获得符合要求的 (x,y)。

3 验证与分析

采用软件仿真和实验测量两种方式验证模型有效性，并对性能特征进行分析。软件仿真使用 MATLAB_R2013a Linux 构建模型，操作系统为 Linux 3.13.0 内核的 Ubuntu14.04。实验服务器采用 Dell PowerEdge R720，2 颗 8-core 3GHz Intel Xeon E5-2690 CPU、224 GB DDR3 内存、3×600 GB 15K RPM 硬盘、1 Gb 网口，Xen 4.4.1 虚拟机监视器[21]，CentOS 6.5 操作系统；部署 16 台 Dell OptiPlex 790 作为客户端，用于模拟用户进行压力测试，4-core 3GHz Intel i5-2400 CPU、8 GB DDR3 内存、1Gb 网口、500 GB 7200RPM 硬盘，RedHat 5.0 操作系统，服务器与客户端通过 H3C 全千兆以太网交换机 S5120-48P-EI 连接。服务器中的密码卡采用实验室研制的 PCI-E x4 接口高性能密码卡，单卡配置 5 颗 Altera 公司 Cyclone V SE 型 FPGA 芯片，含 ARM Cortex-A9 硬核，工作频率 500 MHz，FPGA 片上逻辑工作频率 100 MHz。1 颗 FPGA 用作密码卡控制单元，负责与主机数据通信、卡上资源管理和运算单元调度；其余 4 颗 FPGA 用作密码卡运算单元，均独立支持对称密码算法、公钥密码算法和消息摘要等常用密码运算，与控制单元采用星形拓扑结构连接，如图 4 所示。

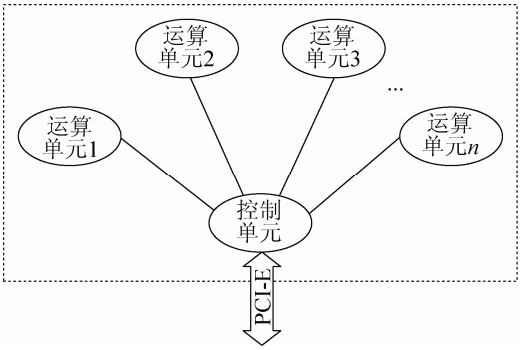


图 4 密码卡各单元连接拓扑
Fig. 4 The topology of encryption card units

3.1 模型有效性

在密码服务系统中建立 4 个虚拟密码域，每 4 个客户端分别绑定 1 个密码域，向密码域发起对称密码服务请求，通过控制客户端任务速率，即可控制任务到达各密码域的速率。密码任务单包最大为 16 K 字节，虚拟密码域中容器为 128 K 字节，容器中聚合任务的数量为 8，密码域数量为 4，密码卡中运算单元数量为 4，任务包聚合时间阈值设置为 200 ms，设备域中队列长度为 16，密码卡读取速率为 60 次/秒，各客户端发送计算量相同的任务到对应密码域。图 5 为将任务到达速率作为输入时的延迟结果。当密码域任务包输入低于 40 次/s 时，仿真延迟时间低于实验延迟时间。这主要是由于在较低输入速率时，容器中数据的传输通常是在达到包聚合时间阈值时进行，图 5 表明模型具有较高的有效性。

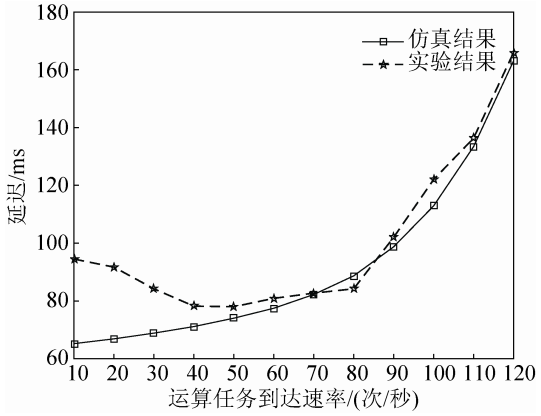


图 5 平均延迟时间的仿真与实验结果对比
Fig. 5 The comparison between simulated and measured results of the average delay

表 1 为将任务到达速率作为输入时的任务拒绝概率。任务负载较低时，到达速率的增加对拒绝概率没有明显影响；当任务到达速率接近系统服务能力时，拒绝概率开始随着到达速率的增加而相应增加，说明此时系统能力已经得到充分利用，可用于为新增加的密码运算任务提供服务的闲置资源减少。

表 1 任务拒绝概率

Tab. 1 The rejection probability of tasks

任务到达速率(次/秒)	拒绝概率(λ_{TP}=60)
10	6.683×10^{-20}
20	4.695×10^{-15}
30	4.5×10^{-12}
40	5.013×10^{-10}
50	2.451×10^{-8}
60	4.777×10^{-7}
70	6.912×10^{-6}
80	6.389×10^{-5}
90	4.646×10^{-4}
100	2.02×10^{-3}
110	5.823×10^{-3}

3.2 系统运算性能

密码域接收到服务请求后，采用立即提交和对任务聚合后提交两种方式。其中立即提交方式，即仅聚合 1 个任务、无聚合时间阈值要求的任务聚合方式。由于前面已经验证过模型的有效性，故可通过仿真方式来验证容器容量与响应延迟之间的关系，对包的聚合时间通过设定的时间阈值进行控制。如图 6 所示，当容器容量增加，即聚合长度增加时，平均响应延迟减少。采用对任务进行聚合处理的方式，单个任务的响应延迟时间更长，但系统具有更高的综合处理性能，即平均单任务延迟时间更短。因此，在密码服务系统虚拟化中，采用任务聚合处理的方式具有更高的性能。任务聚合的数量应当根据系统硬件能力进行设定，如果太高，则会导致密码卡运算单元服务时间变长，系统响应延迟增加。

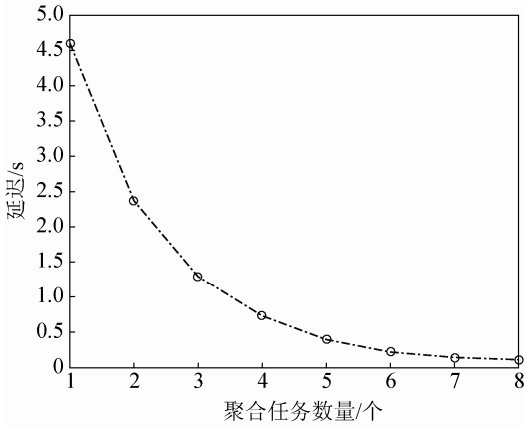
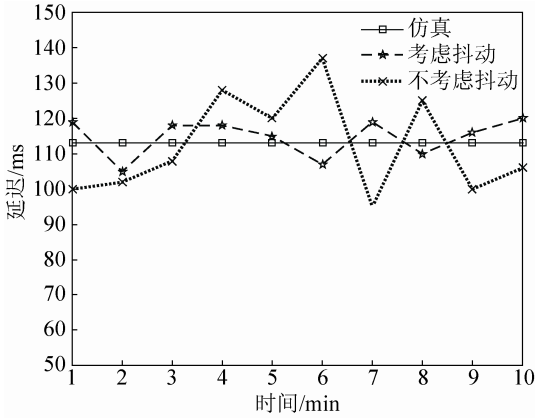


图 6 不同聚合长度时的平均响应延迟

Fig. 6 The average response delay of various combing length

3.3 任务抖动特性

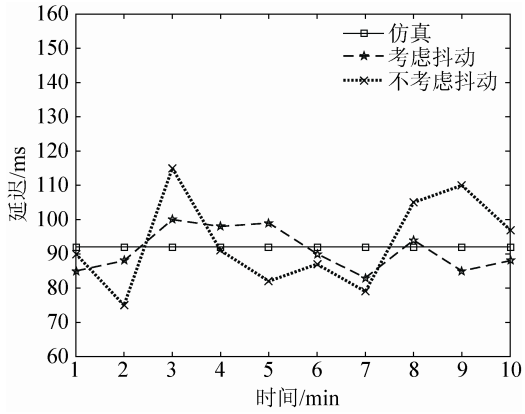
当客户端代表不同应用时，各客户端发送的密码任务需要的处理时间各不相同。实验时接收到的密码任务运算时间，假定在同一密码域内基本保持一致，各密码域之间可以不同。通过对非虚拟化状态下密码服务器的任务处理特征进行统计分析，在虚拟化环境下进行任务聚合时，各个密码域中的容器容量应依据其处理时间来进行设定，即保证各密码域容器内的任务总处理时间一致，从而使各运算单元执行任务包的时间一致，确保系统处理任务时具有较小的抖动幅度，提高系统 QoS。图 7(a)、7(b)为密码卡读取速率 λ_{TP} 分别取 60 次/秒、80 次/秒时，不考虑任务抖动与考虑任务抖动的延迟对比。



(a) λ_{TP}=60

DOI: 10.16182/j.issn1004731x.joss.201708008

第 29 卷第 8 期 系统仿真学报 Vol. 29 No. 8
2017 年 8 月 Journal of System Simulation Aug., 2017



(b) λ_{TP}=80

图 7 任务抖动特性

Fig. 7 The characteristics of task jitter

从图 7 可知，考虑任务抖动时系统具有更优异的性能。图中仿真结果与实验结果存在一定偏差，原因在于运算单元执行各任务包的时间不同，因此在运算单元执行子模型中的任务处理过程不完全服从指数分布，模型预测准确度相应降低。而考虑任务抖动时，任务包的处理时间服从指数分布，延迟抖动较小，且模型预测准确度较高。

4 结论

本文提出了一种适合于分析云计算环境下密码服务系统虚拟化的性能建模方法，该性能模型基于 ISSM 构建。通过仿真和实验两种方式，对密码域数量、包聚合长度、密码卡服务速率等参数在不同配置时，系统响应延迟、任务拒绝概率等性能特征进行了测试和分析，并对降低任务执行时间抖动的有效性进行了验证。结果表明，本文提出的性能模型能够对资源部署、服务总体延迟等各项特征进行准确度量，同时在精度和易用性两方面具有良好的平衡关系，能够有效降低云计算中密码服务系统虚拟化部署的难度。云服务提供商能够依据此模型对响应时间和拒绝服务概率进行可靠的评估，为用户提供高 QoS 的密码运算服务。

本文关于子模型服务时间服从指数分布以及每个密码域只提供某一类密码服务的假设具有一定的局限性，在后续工作中，我们将继续完善密码服务系统虚拟化的性能模型，细化子模型服务类型和定量分析方法。另外，我们将根据云计算环境下密码服务响应要求，对基于优先级的密码服务系统虚拟化进行性能建模分析。

参考文献:

[1] 林闯, 苏文博, 孟坤, 等. 云计算安全: 架构、机制与模型评价 [J]. 计算机学报, 2013, 36(9): 1765-1784. (Lin Chuang, Su Wenbo, Meng Kun, et al. Cloud Computing Security: Architecture, Mechanism and Modeling [J]. Chinese Journal of Computers, 2013, 36(9): 1765-1784.)

[2] 李林, 杨先文, 郑斌. 多任务密码服务系统设计与实现 [J]. 信息工程大学学报, 2013, 14(1): 96-102. (Li Lin, Yang Xianwen, Zheng Bin. Design and Implementation of Multi-Tasking Cryptography Service System [J]. Journal of Information Engineering University, 2013, 14(1): 96-102.)

[3] Vaquero L M, Rodero-Merno L, Caceres J, et al. A Break in the Clouds: Towards a Cloud Definition [J]. ACM SIGCOMM Computer Communication Review (S0146-4833), 2008, 39(1): 50-55.

[4] Sandhu R, Sood S. Scheduling of big data applications on distributed cloud based on QoS parameters [J]. Cluster Computing (S1386-7857), 2015, 18(2): 817-828.

[5] 程仲汉, 官水旺, 黄皓. 多核环境下一种支持动态预测性能损失的方法 [J]. 系统仿真学报, 2014, 26(11): 2803-2809. (Cheng Zhonghan, Guan Shuiwang, Huang Hao. Method to Support Degradation Prediction Online in Multicore Environment [J]. Journal of System Simulation (S1004-731X), 2014, 26(11): 2803-2809.)

[6] Ghosh R, Longo F, Frattini F, et al. Scalable Analytics for IaaS Cloud Availability [J]. IEEE Transactions on Cloud Computing (S2168-7161), 2014, 2(1): 57-70.

[7] 何选森. 随机过程与排队论 [M]. 湖南: 湖南大学出版社, 2010: 186-190. (He Xuansen. Random Process and Queuing Theory [M]. Hunan, China: Hunan University Press, 2010: 186-190.)

[8] Yigitbasi N, Iosup A, Epema D, et al. C-Meter: A Framework for Performance Analysis of Computing Clouds [C]// Proceedings of the 2009 IEEE/ACM 9th International Symposium on Cluster Computing and the Grid (CCGRID '09), Shanghai, China. USA: IEEE Press, 2009: 472-477.

[9] Salah K, Boutaba R. Estimating service response time for elastic cloud applications [C]// Proceedings of the 2012

IEEE 1st International Conference on Cloud Networking (CLOUDNET), Paris, France. USA: IEEE Press, 2012: 12-16.

[10] Mytilinis I, Tsoumakos D, Kantere V, et al. I/O Performance Modeling for Big Data Applications over Cloud Infrastructures [C]// Proceedings of the 2015 IEEE International Conference on Cloud Engineering (IC2E). Tempe, USA. USA: IEEE Press, 2015: 201-206.

[11] Khazaei H, Misic J, Misic V B. A Fine-Grained Performance Model of Cloud Computing Centers [J]. IEEE Transactions on Parallel and Distributed Systems (S1045-9219), 2013, 24(11): 2138-2147.

[12] 明仲, 尹剑飞, 杨巍, 等. 一种 Web 系统性能测试框架及其混合建模过程 [J]. 计算机研究与发展, 2010, 47(7): 1192-1200. (Ming Zhong, Yin Jianfei, Yang Wei, et al. A Web Performance Testing Framework and Its Mixed Performance Modeling Process [J]. Journal of Computer Research and Development, 2010, 47(7): 1192-1200.)

[13] Kraft S, Casale G, Krishnamurthy D, et al. Performance models of storage contention in cloud environments [J]. Software and Systems Modeling (S1619-1366), 2013, 12(4): 681-704.

[14] Pu X, Liu L, Mei Y D, et al. Who Is Your Neighbor: Net I/O Performance Interference in Virtualized Clouds [J]. IEEE Transactions on Services Computing (S1939-1374), 2013, 6(3): 314-329.

[15] Bourguiba M, Haddadou K, El Korbi I, et al. Improving Network I/O Virtualization for Cloud Computing [J]. IEEE Transactions on Parallel and Distributed Systems (S1045-9219), 2014, 25(3): 673-681.

[16] Guan H B, Ma R H, Li J. Workload-Aware Credit Scheduler for Improving Network I/O Performance in Virtualization Environment [J]. IEEE Transactions on Cloud Computing (S2168-7161), 2014, 2(2): 130-142.

[17] Chydzinski A, Adamczyk B. Analysis of a Scheduler for Virtualization of Links with Performance Isolation [J]. Applied Mathematics & Information Sciences (S1935-0090), 2014, 8(6): 2653-2666.

[18] Li D D, Jin H, Liao X F, et al. Improving disk I/O performance in a virtualized system [J]. Journal of Computer and System Sciences (S0022-0000), 2013, 79(2): 187-200.

[19] Tian K, Dong Y Z, Mi X, et al. sEBP: Event Based Polling for Efficient I/O Virtualization [C]// Proceedings of the 2012 IEEE International Conference on Cluster Computing (CLUSTER), Beijing, China. USA: IEEE Press, 2012: 135-143.

[20] Neuts M F. Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach [M]. Baltimore, USA: The Johns Hopkins University Press, 1981: 125-141.

[21] Barham P, Dragovic B, Fraser K, et al. Xen and the Art of Virtualization [C]// Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP '03). New York, USA: ACM Press, 2003: 164-177.

(上接第 1691 页)

[13] 刘俊芳, 张雪英, 宁爱平. PSO 和 ABC 的混合优化算法 [J]. 计算机工程与应用, 2011, 47(35): 32-34, 44. (Liu Junfang, Zhang Xueying, Ning Aiping. Hybrid optimization algorithm of PSO and ABC [J]. Computer Engineering and Applications, 2011, 47(35): 32-34, 44.)

[14] Xia Xuewen, Liu Jinnan, Hu Zhongbo. An improved particle swarm optimizer based on tabu detecting and local learning strategy in a shrunk search space [J]. Applied Soft Computing (S1568-4946), 2014, 23(1): 76-90.

[15] Yang X, Deb S. Multiobjective cuckoo search for design optimization [J]. Computer & Operations Research (S0305-0548), 2013, 40(6): 1616-1624.

[16] Yang Xinshe. Multi objective Firefly Algorithm for Continuous Optimization [J]. Engineering with Computers (S0177-0667), 2013, 29(2): 175-184.

[17] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm [J]. Journal of Global Optimization (S0925-5001), 2007, 39(3): 459-471.

DOI: 10.16182/j.issn1004731x.joss.201708008