文章编号:1671-4598(2015)07-2559-04 DOI:10.16526/j.cnki.11—4762/tp.2015.07.094 中图分类号:TP3 文献标识码:A

基于多维评价模型及改进蚁群优化算法的云计算资源调度策略

蒋 华,张乐乾,王 鑫
(桂林电子科技大学 计算机科学与工程学院,广西 桂林 541004)

摘要:针对云计算环境下资源调度模型未充分考虑资源评价的问题,为更好适应不同节点计算性能和大规模数据环境的处理需求,提出了一种基于多维评价模型的虚拟机资源调度策略;首先,在云计算环境下建立包括网络性能在内的多维资源评价模型,在此基础上提出一种改进的蚁群优化算法实现资源调度策略;然后在云计算仿真平台 CloudSim 上进行实现。实验结果表明,该算法可以更好适应不同网络性能的计算环境,显著提高了资源调度的性能,同时降低了虚拟机负载均衡离差,满足了云计算环境下的虚拟机资源负载均衡需求。

关键词:云计算;蚁群优化算法;资源调度;评价模型

Cloud Computing Resource Scheduling Strategy Based on Multidimensional Evaluation Model and Improved Ant Colony Algorithm

Jiang Hua, Zhang Leqian, Wang Xin
(School of Computer Science and Engineering, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract: Aiming at the resource scheduling model fails to fully consider the resorce assessment in cloud computing environment, a virtual machine resource scheduling strategy based on multi-dimension evaluation model is Proposed to better adapt to the demand for computing performance and large data environment for different node processing. First of all, establish the evaluation model of multidimensional resources include network performance in the cloud computing environment, on this basis proposed an improved ant colony optimization algorithm for resource scheduling strategy; And then realized on the simulation platform of CloudSim. The experiment shows the algorithm in this paper can better adapt to different computing environments of network performance, significantly improve the performance of resource scheduling, and reduce the load balance deviation, meet the virtual machine resource in the cloud computing environment load balancing demand.

Keywords: cloud computing; ant colony optimization algorithm; resource scheduling; resource assessment model

0 引言

随着云计算商业化进程的发展,云计算的概念已经深入人心。在计算机发展过程中,人们在并行计算中实现了将同一个大问题交给多个计算机同时运算;在分布式计算中解决了对计算资源的有效整合和计算任务的合理分配;在网格计算中把众多计算机中的以计算资源为主的各种资源进行整合虚拟化,实现了在虚拟结构(VO)[1];为了适应普遍的信息化需要,人们对网格计算做出改进,但由于其诞生的目的便是解决科学研究问题,对于大众化的需求依然适应不足[2]。于是,基于商业模型应用的云计算逐渐走进人们的视野。

云计算就是在虚拟化技术的基础上,通过网络服务为用户提供对基础资源、应用平台、软件等的服务[3]。其核心是围绕用户需求为中心的虚拟资源调度。为了适应云环境下的虚拟资源调度,大量学者从多个角度对云计算环境下的虚拟资源调度进行深入的研究,并取得一定成果。文献[4]在蚁群优化算法的迭代过程中引入遗传算法加快其收敛速度,并通过逆转变异策略解决局部最优问题,降低了任务的平均执行时间,提高了任务的执行效率。文献[5]在计算节点资源优劣的度量标准里考虑执行时间、网络带宽、网络延迟等因素,利用蚁群优化算法对节点进行计算节点的寻找。

为了尽可能接近数据中心实际网络环境的复杂性,本文在资源调度模型建模过程中引入评价模型,对节点、网络进行评价建模,并在模型中根据节点情况预测作业执行速度,最终对3个模型的进行综合评价,特别考虑在网络评价模型中加入网络距离、带宽、负载等因素,并在蚁群优化算法的每次迭代完成后,通过在计算过程中积累的负载信息,在保持较高计算能力的条件下寻找更有利于负载均衡的路径方案,最终通过仿真实验对模型中的各权重参数进行调整,在提高资源调度效率的同时保证节点的负载均衡,并最终证明本文算法策略的优越性。

1 资源调度模型数学建模

1.1 资源评价模型

在云计算环境下的资源调度,首先需要将计算机系统物理资源通过虚拟化技术抽象为相互独立的虚拟资源,然后将计算任务分配到符合调度策略的最佳虚拟机节点上去。如何将虚拟资源进行抽象并通过节点属性对节点的计算能力进行有效表示,是云计算资源调度研究的一个关键问题。本文将一个虚拟机资源可以表示为:$VM_m(m,EvaCPU_m,EvaMEN_m,EvaSTO_m)$,通过对节点虚拟资源的固有资源(cpu、内存、

收稿日期:2015-02-01; **修回日期**:2015-03-24。
基金项目:国家自然科学基金重点项目(61262074);桂林电子科技大学研究生教育创新计划资助项目(GDYCSZ201422)。
作者简介:蒋 华(1964-),男,河南信阳人,教授,硕士研究生导师,主要从事数据库系统与信息安全技术方向的研究。

物理存储)、网络状况（带宽、距离)、当前运行状态进行统一建模，从多角度对计算节点的计算能力进行表达，尽可能准确的对计算节点的计算能力进行表达，并从中找出最符合计算要求的计算节点，以高效的完成资源调度任务。

相比网格计算，云计算对网络环境的要求更低，在云计算资源调度过程中面临的网络环境更加复杂，本文考虑在建模过程中考虑根据网络带宽，带宽使用率，网络质量参数进行建模，更加符合云计算环境下的资源调度特点。

在物理服务器上运行的各虚拟机的计算能力是通过多方面来评估的，根据在实际场景中的约束条件，我们建立以下节点评价模型，如公式（1）所示：

$$EvaNode_m = \rho_{cpu} EvaCPU_m + \rho_{men} EvaMEN_m + \rho_{storage} EvaSTO_m + \rho_{bandwidth} EvaBAN_m \quad (1)$$

其中：$EvaNode_m$ 代表第 m 号节点固定资源评价值，$EvaCPU_m$、$EvaMEN_m$、$EvaSTO_m$、$EvaBAN_m$ 分别代表虚拟机 CPU、内存、硬盘存储、路径带宽的评价值，Eva_{node}、Eva_{cpu}、Eva_{men}、$Eva_{storage}$、$Eva_{bandwidth}$ 分别代表虚拟机 CPU、内存、硬盘存储、路径带宽的评价值的权重，且 $\rho_{cpu} + \rho_{men} + \rho_{storage} + \rho_{bandwidth} = 1$。

1.2　节点作业执行速度的预测模型

我们利用在文献［5］中的执行速度预测方法和模型，针对在云计算环境下虚拟机计算节点的计算特征，根据节点积累的任务执行速度的历史数据对下一次任务执行速度进行预测评估。在云计算系统中的每个计节点的当前负载程度可以查询，并且在历史记录中，上一次作业完成的平均负载程度可以查询，根据这两项数据，我们利用公式（2）建立模型对执行速度进行预测：

$$EvaV_{mk+1}^{a}(k+1) = (a_{k+1}/a_k) \times [\rho \times EvaV_{mk+1}^{a}(k) + (1-\rho) \times ReaV_{mk}^{a}(k)] \quad (2)$$

其中：a_k 代表第 k 次执行速度预测时系统的实际负载程度，$EvaV_{mk}^{a}(k)$ 代表对序号为 m 的计算节点的第 k 次执行速度的预测执行速度，$ReaV_{mk}^{a}(k)$ 代表序号为 m 的计算节点的第 k 次作业实际的执行速度，执行速度的单位为 MIPS，ρ 为预测执行速度在预测模型中的权重系数。参与作业执行的每一个节点在作业完成后会将本节点的本次作业执行情况进行记录，记录中包括作业执行速度以及系统负载程度。而在节点负载程度中我们采用几个量化的指标进行衡量，系统负载程度的度量模型如公式（3）所示：

$$B_m = N_{1m} \times C_m + N_{2m} \times M_m + N_{3m} \times D_m + N_{4m} \times Net_m \quad (3)$$

其中：m 代表一个虚拟机（VM）的编号，u 表示选作参考的基准虚拟机的编号，N_{1m}、N_{2m}、N_{3m}、N_{4m} 分别代表第 m 个虚拟机中的 CPU 性能参数、内存参数、硬盘参数和带宽参数，且满足 $N_{1m}+N_{2m}+N_{3m}+N_{4m}=1$，$C_m$、$M_m$、$D_m$、$Net_m$ 分别代表第 m 个虚拟机中的 CPU 利用率、内存利用率、硬盘传送率、网络吞吐量 。

1.3　资源节点综合评价模型：

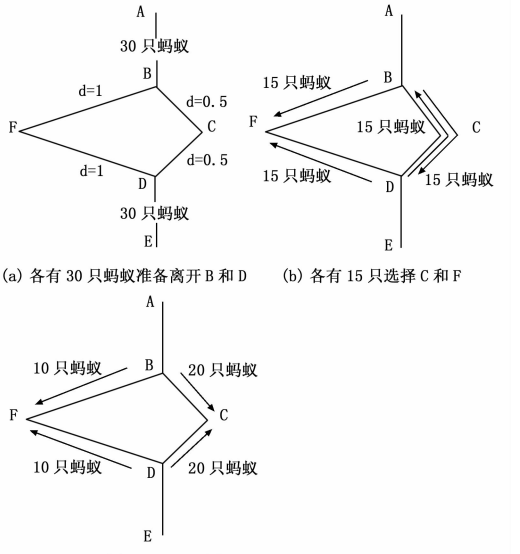
用户提交的计算任务在寻找合适的计算资源节点过程中，首先，根据节点评价模型和节点作业执行速度的预测模型建立节点的计算能力整体评价模型，模型公式为：

$$ENode_m = \lambda EvaNode_m + (1-\lambda) EvaV_m \quad (4)$$

其中：λ 为调节系数，通过调节该系统，对模型中节点固定资源计算能力以及该节点对计算任务的实际执行能力进行适应性调整，以达到最优的计算效果。

2　改进的蚁群优化算法

在 20 世纪 90 年代，Dorigo 等人根据蚂蚁觅食的过程提出的蚁群算法在解决复杂的组合优化问题方面具有很多优点，如正反馈、鲁棒性、收敛速度快等特点，能够较好应用于解决 NP－完全的组合优化问题。自然界中的蚂蚁在觅食过程中需要寻找巢穴与食物源之间的最优路径，蚂蚁在运动过程中通过分泌信息素（Pheromone）来记录其运动路径，并通过感知路径上信息素的浓度选择运动方向，信息素浓度会随时间逐渐挥发降低。通过信息素，蚁群形成一个通过蚁群集体智慧相互协作的高效搜索模型[6-7]。其中有两个关键的规则，一方面蚂蚁概率的倾向于选择信息素浓度较高的路径，另一方面全局路径上的信息素浓度是动态变化的，会随着时间而自然蒸发。对于基本的蚂蚁觅食的搜索模型如图 1 所示。



(a) 各有 30 只蚂蚁准备离开 B 和 D　(b) 各有 15 只选择 C 和 F

(c) 20 只选择 C，10 只选择 F

图 1　蚁群觅食搜索模型

其中，假设 A 为食物所在位置，E 为蚁巢，d 为单位距离，蚂蚁单位时间内匀速爬行单位距离，30 只蚂蚁从 A 和 E 之间往返运动，如图 1 中（a）所示。在 $T0$ 时刻，路径信息素浓度为 0，A 点和 D 点各有 30 只蚂蚁准备出发，蚂蚁以同等概率选择左右两条路，即在 B 点和 D 点分别有 15 只蚂蚁选择 C 点方向和 F 点方向，如图 1 中（b）所示。在 $T1$ 时刻，一个单位时间后，在整个路径 BCD 上来回共有 30 只蚂蚁经过并留下信息素，而在整个路径 BFD 上 30 只蚂蚁刚刚通过路径 BF 和路径 DF 到达 F 点，即整个路径 BFD 上只有 15 只蚂蚁经过并留下信息素。与此同时在 B 点和 D 点又有 30 只蚂蚁准备离开 B 点和 D 点，这个时候路径 BCD 上的信息素浓度是路径 BFD 上信息素浓度的两倍，所以按照蚂蚁的前进规则，在 B 点和 D 点分别有 10 只蚂蚁选择前往 F 方向，20 只蚂蚁选择前往 C 方向。

在算法完成一次迭代后，找到的最优解或许并不是全局最优解，而只是局部最优解[7]。根据文献［8］中的改进的蚁群优化算法，为提高算法的收敛速度，避免陷入局部最优问题，我们在蚁群优化算法的每一次迭代后，通过局部信息素更新规

则，减少新解的组成路径上的信息素浓度，以此来提高蚁群求解的离散型。在蚁群算法建模过程中，我们使用禁忌表 Tabuk（$k=1, 2, 3, \cdots\cdots, m$）对蚂蚁的前进路径进行记录，并在蚂蚁前进过程中不断做动态调整，用 $P_{ij}^{k}(t)$ 表示处于节点 i 的第 k 只蚂蚁在 t 时刻前往节点 j 的状态转移概率，那么概率公式为：

$$P_{ij}^{k}=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum\limits_{s\in allowed_k}[\tau_{in}(t)]^{\alpha}[\eta_{is}(t)]^{\beta}}, & \text{当 } j\in allowed_k\\ 0 & \text{,其他}\end{cases} \tag{5}$$

其中：allowedk 表示蚂蚁 k 下一步的可选节点集合；α 是信息启发因子，表示蚂蚁的运动历史经验在决策中的权重；β 是期望启发因子，表示启发信息在决策中的权重；且限制条件为 $\alpha+\beta=1$；

在对 n 个城市爬行过程中模拟信息素的自然蒸发过程，在 $t+n$ 时刻（i，j）上信息素含量的规则公式为：

$$\tau_{ij}(t+n)=\rho\tau_{ij}(t)+\sum_{k=1}^{m}\Delta\tau_{ij}^{k}(t) \tag{6}$$

其中：ρ 表示信息素的残留系数，且限制 $\rho\subset[0,1]$；$\triangle\tau_{ij}^{k}(t)$ 表示在 t 时刻第 k 只蚂蚁的爬行路径(i,j)上信息素浓度增量。

本文对蚁群算法进行改进，在循环迭代后，通过审查资源节点的有效性，对解集进行修正，以得到一种能够查找最优的有效计算节点路径的蚁群优化算法。该蚁群优化算法的流程如下。

第一步：配置蚁群算法相关参数；

第二步：对资源节点路径上的信息素进行初始化，对蚁群进行初始化；

第三步：蚁群中的蚂蚁按照公式（1）的计算规则进行爬行，选择爬行路径，并将爬行过的资源节点加入到禁忌表 Tabuk 中，直到遍历所有资源节点；

第四步：根据禁忌表 Tabuk，对所有路径进行筛选得到迭代产生的最优路径；

第五步：重复执行第二到四步，直到达到最大迭代次数 I _ Max；

第六步：对路径节点有效性进行审核，检查是否满足计算条件，若满足，则继续下一步；若不满足，则重新选择路径；

第七步：根据节点负载程度 B 对所有的路径解进行比较，选择负载程度较低的解作为全局的最优解。

算法流程图如图 2 所示。

3　实验结果与分析

为了验证该优化算法在云计算资源调度实际应用中的优化效果，本文通过对算法进行仿真实验，并与普通算法进行比较，证明本文改进的蚁群优化算法的优越性。仿真实验环境选用 CloudSim 仿真平台。CloudSim 是澳大利亚墨尔本大学的网格实验室和 Gridbus 项目在 2009 年推出的一个云计算仿真软件，可以支持在云计算环境下的基础结构、虚拟引擎、资源调度等的模拟仿真功能[9]。

本文使用 CloudSim 1.0 版本，对平台中现有类和方法进行扩展，其中参考文献［10］中的虚拟机匹配策略，并通过编写 VmAllocationPolicy 类中的 allocateHostForVm 方法实现其算法功能 。通过重写 DataCenterBroker 和 Cloudlet 等类，对文中改进的蚁群优化调度算法进行模拟仿真实现，对代码重新

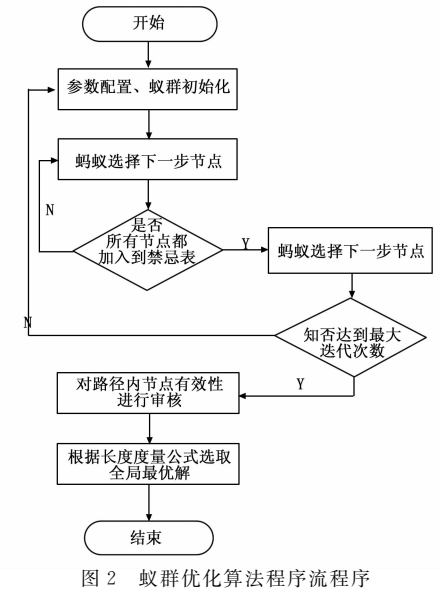


图 2　蚁群优化算法程序流程程序

编译后获得仿真环境。在仿真试验中，我们将改进的蚁群优化算法与 Min－min 算法、Max－min 算法执行相同任务调度的平均执行时间和负载均衡程度进行对比。如图 3 所示，本文改进的蚁群优化算法的执行时间比两种基本调度算法执行时间要小，并且随着任务数量的增加，该算法的任务执行时间上升幅度要小于两种基本调度算法，显然该算法具有较好的执行效率。

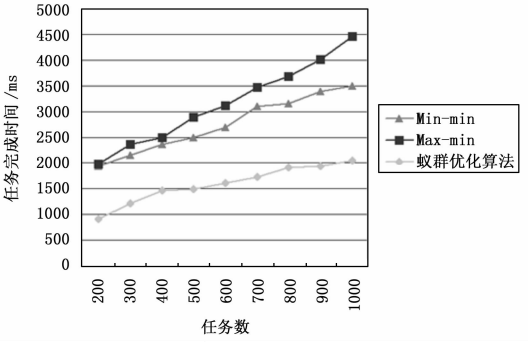


图 3　任务完成时间对比

同时，在仿真过程中，对节点的负载均衡离差进行统计并比较，得出各算法的负载均衡程度的比较以及负载均衡程度随着任务数增多的变化趋势。负载均衡离差 ε 计算公式为：

$$\varepsilon=\sqrt{\frac{\sum\limits_{i=1}^{m}(B_i-\overline{B}_i)^2}{m-1}} \tag{7}$$

从图 4 中我们可以看出，在提高作业执行效率，降低作业执行时间的同时，文中改进的蚁群优化算法具有更好的负载均衡效果。

综合上述实验对比结果可知，改进的基于云计算环境蚁群优化算法通过对资源质量的建模评价形成可信的资源描述，通过改进的蚁群优化算法在可信资源中进行查找，筛选出优化的资源路径，同时通过对路径节点的再次判断，从资源负载角度对最终资源节点进行优化并在保证作业执行效率的同时选取更有利于负载均衡的方案进行任务的分配和执行，保证了云计算环境下资源调度的高效性，加快了蚁群优化算法的收敛速度，也使得系统资源负载更加均衡，并最终提高了云计算系统资源

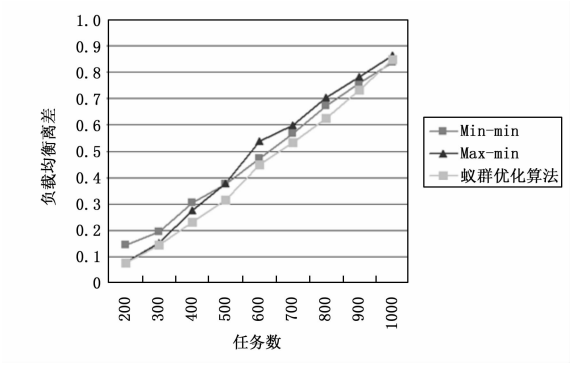


图 4　负载均衡离差对比

调度的执行效率，缩短了任务执行时间，适合应用于大规模计算任务的云计算环境下资源调度。

4　结论

针对资源调度策略，本文构建了资源评价模型，提出了一种蚁群优化算法实现对虚拟资源的高效调度，在仿真平台上通过重新编写实体类，搭建仿真环境对基于蚁群优化算法的资源调度策略进行仿真实验，并通过与传统最小负载优先调度算法、轮转调度算法的对比表明该策略在任务执行效率和负载均衡度方面具有较好表现。

参考文献：

[1] 王　鹏. 走进云计算出版社 [M]. 北京：人民邮电出版社，2009.
[2] MaoZhenLi M B. 网格计算核心技术 [M]. 王相林，张善卿，王景丽 译，北京：清华大学出版社，2006.
[3] 方　巍，文学志，潘吴斌，等. 云计算概念，技术及应用研究综述 [J]. 南京信息工程大学学报（自然科学版），2012，8：28－28.
[4] 王永贵，韩瑞莲. 基于改进蚁群算法的云环境任务调度研究 [J]. 计算机测量与控制，2011，19（5）：1203－1204.
[5] 华夏渝，郑　骏，胡文心. 基于云计算环境的蚁群优化计算资源分配算法 [J]. 华东师范大学学报：自然科学版，2010，1（1）：127－134.
[6] 李锋华. 基于蚁群算法的云计算资源负载均衡调度算法研究 [D]. 昆明：云南大学，2013.
[7] 史恒亮，白光一，唐振民，等. 基于蚁群优化算法的云数据库动态路径规划 [J]. 计算机科学，2010，37（5）：143－145.
[8] 田文洪，赵　勇. 数据中心资源优化调度：理论与实践 [J]. 计算机安全，2014（3）：56.
[9] 刘　波，刘青凤. 基于 QoS 和效用的云计算资源调度模型 [J]. 计算机测量与控制，2014（3）：56.
[10] 张　牧. 云计算和多维 QoS 环境中基于蚁群优化算法在虚拟机资源负载均衡问题中的研究 [J]. 计算机科学，2013，40（11A）：60－62.

（上接第 2558 页）

4）基于 COTS 器件开发，有效降低研制成本，缩短研制周期。

3　应用前景

基于三模冗余的低成本高可信微纳通用计算机易于实现标准化、模块化，易于技术更新，研制周期短，经济成本低，可应用于空间科学实验与新技术演示验证、分布式空间系统[10]及空间安全与快速侦察等应用领域。

3.1　空间科学实验与新技术演示验证

基于三模冗余的低成本高可信微纳通用计算机采用成熟的 COTS 器件设计，研制周期短、成本低、设计约束限制较小，支持和使用大量的设计新概念、新方法；提高广大航天爱好者的广泛参与度。

同时也可充分利用先进的微电子技术、MEMS 技术、FPGA 技术、片上系统技术和集成技术等进行太空演示验证，大大提高系统的可靠性和卫星功能密度，减少卫星体积、重量、功耗以及研制成本。

3.2　分布式空间任务应用

基于该微纳通用计算机搭建的微纳卫星体积小、特点突出，可以搭载成百上千颗一次发射，通过星间组网通信、编队飞行共同完成空间环境的科学测量、空间碎片监测等分布式空间任务；也可在太空中部署成毯状或者伞状星云，利用星座和集群特点实现分布式空间天线功能，增强通信链路通信能力，可用来探测宇宙、进行地球观测研究时作为通信系统。

3.3　空间安全与快速侦察应用

由于体积小、重量轻、成本低的特点以及批生产和快速研发部署等能力，使其组装的微纳卫星具有隐蔽性强、空间生产能力强、系统可重组和再生能力强等优势，难以追踪和摧毁，可用于海上目标侦察、多幅低分辨率高清成像、敌方局部区域探测，特别适合于执行空间安全任务，在空间军事应用上具有很好的前景。

4　结论

本文针对微纳卫星应用发展需求，基于 COTS 器件，提出了一种基于三模冗余的低成本高可信微纳通用计算机体系架构，通过三模冗余设计、硬件看门狗、看闩锁电阻设计等硬件加固设计增强系统可靠性，利用硬件接口标准化、软件分层化提高系统通用性，使其在科学实验与新技术演示验证、分布式空间系统、军事等方面具有良好的应用前景。

参考文献：

[1] 林来兴. 立方体星的技术发展和应用前景 [J]. 航天器工程，2013（6）：90－98.
[2] He J，Qiu Y. COTS－Based On－Board Computer Design and Reliability Study [A]. 2013 the 5thICAT [C]. 2013.
[3] 傅忠传，陈红松，崔　刚，等. 处理器容错技术研究与展望 [J]. 计算机研究与发展，2007，44（1）：154－160.
[4] 王　挥，潘海燕，沙李鹏. 嵌入式星载计算机故障注入系统 [J]. 计算机测量与控制，2011，19（10）：2335－2336.
[5] OhN，Shriven P，McCluskeyE. Error detection by duplicated in structions in super scalar processors [J]. IEEE Transactions on Reliability，2002，51（1）：63－75.
[6] 李爱国，洪　炳，王　司. 一种星载计算机数据流软故障纠正算法 [J]. 宇航学报，2007，28（4）：284－288.
[7] 李军予. 立方体纳卫星的发展及其启示 [J]. 航天器工程，2012（6）：80－87.
[8] Lai A. Space－ready，Radiation Tolerant Processor Modules：A COTS Technology Strategy，Military Embedded Systems Resource Guide [Z].
[9] Lee S，Toorian A，Munakata R，et al. CubeSat design specification [S]. Oklahoma：C. P. S. University，2009.
[10] 苏瑞丰，张科科，宋海伟. 甚小型卫星发展综述 [J]. 航天器工程，2013，22（6）：104－111.