

# 云计算平台中面向车联网应用的能耗感知调度算法

邓聃婷<sup>1</sup> 滕飞<sup>1,2</sup> 杨燕<sup>1</sup>

(西南交通大学信息科学与技术学院 成都 611756)<sup>1</sup>

(计算机软件新技术国家重点实验室(南京大学) 南京 210023)<sup>2</sup>

**摘要** 针对面向车联网应用的云计算平台的高能耗问题,提出一种采用节能整合策略的能耗感知调度算法——任务集整合算法(Task Set Consolidation Algorithm)。该算法的主要思想是通过减少活跃物理服务器的数目,有效降低云平台的能量消耗。建立了云平台模型、车联网任务集模型和能耗模型,确定了云平台的节能目标函数和变量因子。仿真实验通过模拟多维资源多并发任务集的云平台环境,以物理服务器的活跃时间和活跃数目、云平台的能量消耗作为性能指标,将任务集整合算法与现有算法进行了比较。实验结果表明,TSC 算法能够在避免任务集资源发生冲突的情况下,使面向车联网应用的云平台激活的物理服务器数量达到最少,能耗降到最低。

**关键词** 车联网,云计算,能耗感知,节能整合

**中图分类号** TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.3.008

## Energy-aware Scheduling Algorithm for Internet of Vehicles on Cloud Platform

DENG Dan-ting<sup>1</sup> TENG Fei<sup>1,2</sup> YANG Yan<sup>1</sup>

(School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China)<sup>1</sup>

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China)<sup>2</sup>

**Abstract** For the issue of high energy consumption of the internet of vehicles on cloud platform, we put forward an energy-aware scheduling algorithm, called task set consolidation algorithm (TSC). The main idea of the algorithm is reducing the number of active physical servers to reach the goal of bringing the cloud platform's energy consumption down. We built the objective function and the variable factors through the energy consumption model, the internet of vehicles task set model and cloud platform model. Our simulation experiment compared TSC with the existing algorithms on cloud platform environment, with the performance index of physical servers' active time and number and the energy consumption of cloud platform. The experimental results show that TSC is able to minimize the number of activate physical servers on the cloud platform, as well as the energy consumption.

**Keywords** Internet of vehicles, Cloud computing, Energy-aware, Power-saving consolidation

## 1 引言

在互联网、物联网之后,车联网作为未来智慧城市的另一个标志,正在悄然崛起。车联网是一个高效的监管和综合服务系统,它能通过装载在车辆上的电子标签识别技术,提取和利用所有信息网络平台上的车辆的属性信息以及静态/动态信息,同时又能根据不同需求对所有车辆提供综合服务,并对其运行状态进行有效的监管和管理<sup>[1]</sup>。在此基础上,车联网技术还可以实现智能交通管理,减少交通拥堵和事故,并降低燃油消耗,同时还可以让各种新型的业务和应用成为现实<sup>[2]</sup>。车联网技术正逐步走进社会视野,引起越来越多人的关注和重视,先有百度官方宣布,百度车联网战略于 2015 年 1 月 27 日正式发布。紧接着,包括腾讯、阿里巴巴、百度在内的 3 大互联网巨头都卷入了一场关于车联网系统的争夺战。这样一

来,随着车联网业务量的增加,其对数据存储和计算能力的要求也会越来越高。

云计算是近年来高速发展的重要的 IT 技术之一,是基于互联网的相关服务的计算模式,其计算资源以服务的方式提供给用户。用户可以通过多种方式如电脑、笔记本、手机等接入到云计算平台,并且按自己的计算需求进行资源获取。面向车联网应用的云计算平台能够对采集的海量感知数据进行存储和综合加工分析,根据数据分析的结果,提供各种类型的综合服务,这将是面向车联网的云平台的核心组成部分,也是应用的基础<sup>[3]</sup>。

但是,云计算平台一直以来都面临着高能耗问题,在已经到来的全球能源危机面前,能耗问题显得愈加严峻。为了更好地发展面向车联网应用的云计算平台,研究云平台的能耗感知调度算法有着至关重要的意义。目前,降低云平台能耗

到稿日期:2015-03-21 返修日期:2015-06-15 本文受国家自然科学基金资助项目(61202043,61170111),网络智能信息处理四川省高校重点实验室开放课题资助项目(SZJJ2014-049)资助。

邓聃婷(1990—),女,硕士生,主要研究领域为云计算节能调度;滕飞(1984—),女,博士,讲师,主要研究领域为云计算、并行计算、 workflow 调度等,E-mail:fteng@swjtu.edu.cn(通信作者);杨燕(1964—),女,教授,博士生导师,主要研究领域为数据挖掘、计算智能、集成学习等。

的主要方法有<sup>[4]</sup>: 关闭休眠技术(Resource Hibernation)、电压动态调整技术(Dynamic Voltage Scaling, DVS)和虚拟化技术(Virtualization)。相对来说,这3种技术中较常用的是虚拟化技术<sup>[5]</sup>。虚拟化技术有3种常见的部署整合方法: 第一种是节能整合方法, 根据现有任务集资源占用情况的分析结果进行整合, 使其占用尽量少的物理服务器资源; 第二种是节能部署方法, 根据现有的虚拟集群需求分配活跃的物理服务器, 并且让剩余的未使用的物理服务器处于低能耗的状态<sup>[6]</sup>; 第三种是节能迁移方法, 进行动态检测, 当虚拟机放置不合理时, 对放置不合理的虚拟机进行迁移。虽然迁移虚拟机后可以平衡工作负载并减少物理服务器的数量, 但在迁移虚拟机的过程中会让物理服务器承受极大的系统开销, 如网络传输开销、CPU消耗等, 有时迁移的成本甚至会超过关闭物理服务器所带来的收益<sup>[7]</sup>, 因此本文不考虑节能迁移技术。除此之外, 节能整合技术和节能部署技术各有优点, 本文主要考虑采用节能整合技术来降低面向车联网应用的云平台的能量消耗。

节能整合研究如何对云平台的任务集进行整合, 分配尽量少的物理服务器, 关闭空闲物理服务器, 以达到节能效果。Qian Zhu<sup>[8]</sup>等人提出一个基于虚拟环境下科学工作流的节能整合架构——pSciMapper。架构中包含一个节能整合算法, 其主要思想是计算不同工作流之间的相似度, 如果相似度小于一个阈值, 则对工作流进行整合。Lee和Zomaya<sup>[9]</sup>提出了两个节能整合算法——ECTC和MaxUtil, 两个算法的过程相似: 输入任务和每种云计算资源(如CPU、内存、磁盘等), 计算每种类型资源与任务的开销函数, 如果A类型资源与该任务的开销函数大于其他任何类型资源与任务的开销函数, 则对该任务分配A资源。Beloglazov<sup>[10]</sup>等人基于云计算应用程序的两个特点——多样性和动态可变性, 提出了对于虚拟机的部署问题应该采取在线持续优化的方法。Berra<sup>[11]</sup>等人从机器学习技术这一角度出发来研究虚拟机的动态整合问题, 从而同时实现能耗和SLA的双重目标。

本文针对面向车联网应用的云计算平台的特点, 提出了一种能耗感知调度算法——任务集整合算法(Task Set Consolidation Algorithm)。面向车联网应用的云平台的任务集是指拥有多个任务的执行集合, 任务类型很多, 比如油耗计算任务处理、实时交通任务处理、交通堵塞预测任务处理等。TSC算法的目标是在不降低计算性能的前提下, 最小化面向车联网应用的云平台的能耗。其主要思想是将任务集整合到较少的物理服务器上, 这样就可以减少活跃物理服务器的数目, 降低云平台的能量消耗。

本文第2节介绍有关的系统模型; 第3节提出节能整合策略; 第4节对节能整合策略进行性能测试; 最后总结全文。

## 2 系统模型

本节介绍提出的系统模型, 包含云平台模型、车联网任务集模型和能耗模型。

### 2.1 云平台模型

云平台模型中包含一个车联网任务集整合控制器和很多台同构的物理服务器。本文的云平台模型参考文献<sup>[12]</sup>。任务集整合控制器用于整合任务集, 使之占用尽量少的物理服务器资源。

定义1 云平台表示为集合  $CloudPlat = \{pm_1, pm_2, \dots, pm_m\}$ 。CloudPlat中包含  $m$  台物理服务器,  $pm_i$  代表云平台中任意一台物理服务器。

本文的云平台模型中的每台物理服务器  $pm_i$  都是同构的, 都配置有相同的物理资源。

定义2 一台物理服务器的资源用一个四维向量  $(r_1, r_2, r_3, r_4)$  来表示,  $r_1$  代表CPU的个数,  $r_2$  代表Memory的大小,  $r_3$  代表Disk的大小,  $r_4$  代表Network的大小。

每台物理服务器拥有一定数量的物理资源, 每台物理服务器为了满足任务集的需求, 向虚拟机提供资源。

### 2.2 车联网任务集模型

在解释车联网任务集模型之前, 先给出车联网模型的概念。如图1所示, 本文的车联网模型中包含3个实体, 分别是信息处理中心、路测基站、汽车。信息处理中心与路测基站进行信息交换, 信息处理中心与汽车之间进行信息获取, 路测基站监控汽车的运行。

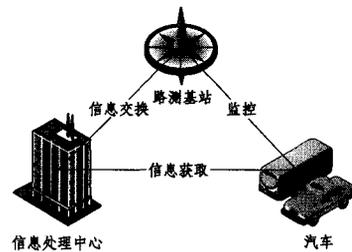


图1 车联网模型

车联网任务集指的是拥有一个或多个车联网任务的执行集合。车联网拥有很多不同类型的任务, 比如汽车需要获取实时道路信息, 这是一个信息查询任务, 属于I/O密集型任务; 交通拥堵情况的预测, 这是数据处理任务, 属于计算密集型任务; 信息处理中心与路测基站进行信息交换, 这是信息交换任务, 属于通信密集型任务。当然, 车联网不同类型的任务对资源的需求是不一样的, 例如, 计算密集型与通信密集型这两种类型的任务对资源的需求是不一样的, 第一种任务需要高效的计算资源, 第二种任务需要快速的网络资源。因此, 计算密集型任务侧重于CPU和内存的大小, 通信密集型任务则侧重于网络带宽的大小。由此可见, 由不同车联网任务组成的车联网任务集对计算资源的需求也是不相同的。本文假设所有车联网任务集都是相互独立的。

定义3 车联网任务集的资源需求情况用一个四维向量  $(vr_1, vr_2, vr_3, vr_4)$  来表示。  $vr_1$  代表vCPU的个数, 即车联网任务集所需要的CPU个数;  $vr_2$  代表vMemory的大小, 即车联网任务集所需要的Memory大小;  $vr_3$  代表vDiks的大小, 即车联网任务集所需要的Storage的大小;  $vr_4$  代表vNetwork的大小, 即车联网任务集对Network的需求量。

在面向车联网应用的云平台上有很多不同类型的任务, 任务根据自身特点进行整合, 组成任务集, 然后对整合后的车联网任务集分配虚拟集群资源, 不同任务集使用的虚拟集群共享同一个云平台上的物理服务器。

定义4 车联网任务集在特定时刻对特定类型资源的需求量用一个向量  $(T_{1,r_1}, T_{2,r_1}, \dots, T_{10,r_1})$  来表示。  $T_{i,r_j}$  代表  $i$  时刻任务集对  $r_j$  类型资源的需求量。本文总共获取10个特定时刻的资源需求。

### 2.3 能耗模型

面向车联网应用的云平台的能量消耗主要分为3个部分:物理服务器能耗、网络能耗、冷却设备能耗。其中,物理服务器是云平台的关键设备,所以物理服务器能耗是云平台能耗的主要部分。因此,本文暂不考虑网络能耗和冷却设备能耗。

能耗指的是计算机系统一段时间内的能量消耗<sup>[13]</sup>,单位是焦耳(J),其定义式如下:

$$E = \int_t^{t+\Delta} P dt$$

其中, $t$ 为时间, $P$ 为功耗, $E$ 为系统的总能量消耗。功耗是指单位时间内能量的消耗,反映计算机系统消耗能量的速率,单位是瓦特(W)。

本文采用文献<sup>[14]</sup>中功耗的一般定义式,如下:

$$P(u(t)) = kP_{\max} + (1-k)P_{\max}u(t)$$

其中, $P_{\max}$ 为物理服务器满负载时的功耗,文中将其设为 $250W^{[6]}$ 。 $k$ 取值为0.6,物理服务器在空负载时的功耗为满载时功耗的60%<sup>[14]</sup>。这说明,物理服务器在空闲状态时,如果不将其关闭或休眠,物理服务器的功耗也是很大的。 $u(t)$ 为资源利用率,资源利用率的大小主要与物理服务器中的资源占用量相关,由此可见物理服务器的功耗与CPU资源利用率、内存读写吞吐量、磁盘读写字节数以及磁盘读写时间等息息相关。

根据以上描述,面向车联网应用的云平台的能耗 $E_{cloudplat}$ 可以看作是车联网云平台内的所有物理服务器的能耗值之和,其定义式如下:

$$E_{cloudplat} = \sum_{i=1}^m E_{pm_i}$$

$E_{pm_i}$ 为一台物理服务器的能耗。

由上述分析可知,云平台开启的物理服务器越少,能耗越低。

### 3 节能整合策略

本节介绍的核心算法应用于云平台模型中的任务集整合控制器。任务集整合算法的主要过程是:首先,任务集整合控制器对面向车联网应用的云平台的任务集进行信息获取;然后对其进行分析计算,得到合理的任务整合方案,并对其进行整合;重复这个过程,直到任务集不能够被整合;最后对整合好的新任务集进行物理服务器的分配。

上述任务集整合算法的过程主要面临以下几个问题:第一是整合后的任务集是否会超出单个物理服务器的资源负载;第二是对于不同的任务集,任意对其进行整合是否会产生冲突。

对于第一个问题,算法在进行整合判断之前,首先对待整合的两个任务或任务集的资源需求量进行了判断,必须保证它们的资源需求之和小于或者等于物理服务器的资源负载,才能够进行后续的整合判断。

对于第二个问题,算法的目标是在不发生冲突的情况下,对不同类型的任务集进行整合。本文引用文献<sup>[13]</sup>中的dist测量标准来检测任务集之间的差异,衡量两个任务集是否能够被整合。

定义:dist用于描述两个任务集之间的差异。

$$Dist_{i,j} = \sum_{R_i^1 R_j^2} (aff\_score(R_i^1 R_j^2) \times Corr(peak_i^{R_1^1}, peak_j^{R_1^1}) \times Corr(peak_i^{R_2^2}, peak_j^{R_2^2}))$$

$R_i^1, R_j^2$ 代表来自于任务集*i*和*j*的任何类型的资源,对于CPU、内存、磁盘、网络,总共有10种类型的组合方式。 $Corr$ 是两个任务集对于特定资源的皮尔森相关系数。文献<sup>[13]</sup>认为考虑峰值的相关性比其他值的相关性更有必要,因为两个协同任务更有可能在峰值的时候发生冲突。 $aff\_score$ 是预先设定的值,其取值范围是0到1,0表示 $R_i^1, R_j^2$ 两个资源在结合到一起时冲突最小;反之,1表示冲突最大。比如,对两个网络密集型任务集进行整合会造成较大的计算开销,但是,如果这两个任务集是直接相关的,通信开销就会大大减小,所以,对于两个网络密集型任务集, $aff\_score$ 是很小的。

定义dist的意义是对两个拥有不同类型任务的任务集进行整合,即任务集的资源需求不同。但是对于拥有相同资源需求的任务集,只要在同类资源竞争可避免的情况下,也是可以进行整合的。

#### 算法 任务集整合算法

输入:任务集列表 TaskSet, 服务器集合 Server

输出:任务集整合方案 ConsolidationList

1. 初始化任务集列表和服务器集合
2. While TaskSet.size >= 2
3. Task\_i = TaskSet.get(0)
4. for (j=1; j <= TaskSet.size; j++)
5. if Task\_i + Task\_j <= Server
6. Dist\_ij = CalculateDist(Task\_i, Task\_j)
7. else
8. icount++
9. end if
10. end for
11. if icount >= TaskSet.size - 1
12. TaskSet.remove(Task\_i)
13. else
14. 将 Dist\_ij 最小的 Task\_i 与 Task\_j 整合
15. TaskSet.remove(Task\_i) //删除 Task\_i
16. TaskSet.remove(Task\_j) //删除 Task\_j
17. TaskSet.add(Task\_new) //增加 Task\_new
18. end if
19. end for
20. end while

任务集整合算法展示了如何获取任务集整合方案。首先对到达的任务集进行信息获取,取出任务集列表中的第一个任务集,遍历剩下的任务集列表中的任务集,找出与第一个任务集整合后不会超过物理服务器资源负载的任务集,并分别计算它们与第一个任务集的dist,选出dist最小的任务集,将其与第一个任务集进行整合,整合后形成新的任务集,更新任务集列表,删除两个整合前的任务集,增加整合后的新任务集。假如剩下的任务集与第一个任务集的资源需求之和都大于物理服务器资源负载,则从任务集列表中删除第一个任务集,因为它已经饱和,不能再进行整合。重复上述过程,直至任务集列表中只剩一个任务集,最后为剩下的这一任务集分配物理服务器。

## 4 性能评估

本节将任务集整合算法 (Task Set Consolidation Algorithm) 与最优拟合整合算法 (Best-Fit Consolidation Algorithm)、首次拟合整合算法 (First-Fit Consolidation Algorithm) 进行对比。最优拟合整合算法 (BF) 是一种针对云平台的节能整合算法, 其主要过程如下: 首先对所有任务集按照各自的资源利用率进行排序, 先对资源利用率最高的任务集进行整合, 选取资源利用率次高的任务集, 判断两个任务集整合后是否会超出物理服务器的资源负载, 如果不会超出, 则对其进行整合, 再计算整合后形成的新任务集的资源利用率, 将新任务集按照资源利用率的大小插入到任务集列表中的合适位置; 反之, 如果超出物理服务器的资源负载, 则按任务集列表顺序查找后面的任务集, 如果找到能够整合的任务集, 则重复上述过程, 假如遍历列表后还没找到能够整合的任务集, 则从任务集列表中取出该任务集, 对其进行物理服务器资源的分配。重复这个过程, 直到所有任务集列表为空。首次拟合整合算法 (FF) 的基本过程与 BF 算法相差不大, 唯一的区别在于 FF 算法不会对任务集列表进行排序, 每次都对任意顺序的任务集列表中的第一个元素进行整合的判断。

为了验证算法的有效性, 本文设计了一个仿真实验。TSC 算法的仿真框架由 Java 实现。本文的仿真框架模拟了一个面向车联网应用的云平台, 云平台上有很多车联网任务集, 任务集整合控制器对车联网任务集进行整合, 并对整合后的新任务集分配物理服务器。当物理服务器上的任务集全部执行完成时, 将物理服务器调整至休眠状态。本文用到的评价指标包括: 能量消耗、服务器开启时间和活跃物理服务器数量。本文通过能耗模型来估计能量消耗。

在仿真实验中, 假设云平台有足够的物理服务器资源, 以满足所有任务集的资源需求; 假设云平台的所有物理服务器都是同构的。

本文实验模拟了拥有不同类型任务的任务集, 任务类型主要有以下几种:

- 1) Traffic\_Analysis, 交通状况分析, 对采集的信息进行数据分析, 可用于交通路况预测, 属于计算密集型任务。
- 2) File\_Zip, 文件压缩, 由于数据信息在传输之前需要被压缩, 属于 I/O 密集型和计算密集型任务。
- 3) Information\_Exchange, 信息交换, 路测基站与信息处理中心之间存在信息交换, 属于通信密集型任务。
- 4) Information\_Inquire, 信息查询, 包括实时路况查询、附近停车场情况查询等, 属于 I/O 密集型任务。

对于不同类型的任务集, 有以下几个方面的描述:

1) 定义 3 中的四维向量  $(vr_1, vr_2, vr_3, vr_4)$  用来描述任务集对每种类型资源的需求情况, 本文实验假设每台服务器每种类型的总量为 100 个单位,  $vr_i$  的取值为 0 到 50 个单位之间的整数。表 1 给出了不同类型任务的资源需求取值范围。

表 1 各个应用的资源需求取值范围

	vCPU	vMem	vSto	vNet
Traffic_Analysis	[40, 50]	[30, 40]	[10, 20]	[0, 10]
File_Zip	[40, 50]	[10, 20]	[40, 50]	[0, 10]
Information_Exchange	[10, 20]	[0, 10]	[30, 40]	[40, 50]
Information_Inquire	[10, 20]	[0, 10]	[40, 50]	[30, 40]

2) 每个任务集的预计执行时间  $T$ , 在实验中,  $T$  的取值范围是 50 到 100 之间的整数, 单位是 s。

3) 定义 4 中的任务集在特定时刻的资源需求量为  $(T_{1r_1}, T_{2r_1}, \dots, T_{10r_1})$ , 本文一共获取 10 个特定时刻,  $T_{ir_j}$  的取值范围是 1 到 10 之间的整数。

实验过程如下: 首先获取面向车联网应用的云平台上的任务的信息, 分别按照 3 种算法对任务进行整合形成任务集, 根据整合方案开启物理服务器, 记录开启的物理服务器的数目以及各自的开启时刻。当任务集执行完成后, 关闭物理服务器, 记录关闭时刻。物理服务器的活跃时间为关闭时刻减去开启时刻。当提交的作业全部完成时, 仿真实验结束。

图 2 为当作业数量为 10 时 FF 算法、BF 算法、TSC 算法以及不进行整合 4 种情况下的物理服务器活跃时间的对比图。由图 2 可知, 当作业数量为 10 时, 不进行整合的情况下总共要开启 10 台物理服务器, 而 3 种整合算法开启 3 台物理服务器, 但是这 3 种算法的每台物理服务器的活跃时间是不相同的。相比而言, BF 算法的物理服务器活跃时间是最短的。虽然 TSC 算法的物理服务器活跃时间不是最短的, 但由图 3 可知, 其能量消耗是最低的, 这是因为物理服务器的能量消耗不仅与它的活跃时间相关, 还与 CPU 资源利用率、内存读写吞吐量、磁盘读写字节数以及磁盘读写时间等相关, 而 TSC 算法中设置的 *aff\_score* 用来控制相应的任务集整合, 使整合后能耗小的任务集更容易被整合。

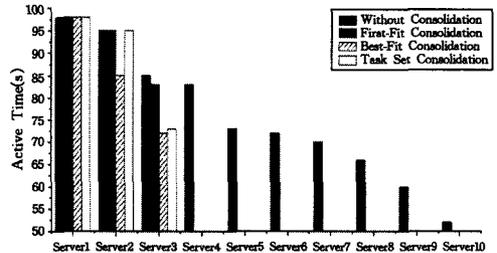


图 2 10 个任务集时物理服务器活跃时间对比

图 3 为 FF 算法、BF 算法、TSC 算法以及不进行整合 4 种情况下的能量消耗折线图。如图 3 所示, 随着面向车联网应用的云平台的任务数目增加, 这 4 种情况的能量消耗几乎呈线性增加。从图中可明显看出, 在刚开始即任务刚开始提交时, 除去不进行整合的情况, 其余 3 种算法的能量消耗差异不大; 随着时间的增加, 任务的提交量增加, 3 种算法出现差异。TSC 算法在相同条件下能量消耗较少, 比不进行整合的情况下减少了 67.6% 的能耗, 比 FF 算法减少了 23.12% 的能耗, 比 BF 算法减少了 6.48% 的能耗。这说明, TSC 算法是 3 种算法中最节能的。

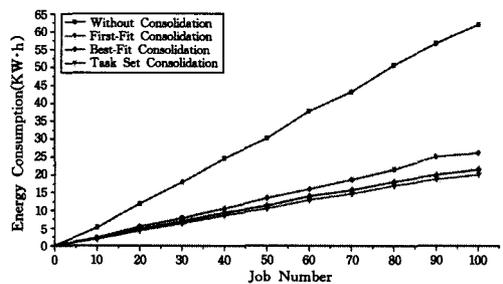


图 3 能量消耗折线图

图 4 为 3 种算法的能量消耗对比图。本文将不进行整合情况下的能量消耗视为 100%，与其他 3 种算法进行了对比。从图中可以看出，BF 算法和 TSC 算法随着任务集数目的增加，其能耗优势越来越明显，但 FF 算法由于每次都是从随机任务集列表中选取第一个任务集进行整合，因此其能耗优势并不会随任务数目增加而发生变化。从图 4 中也可以得出与图 3 相同的结论，即 TSC 算法是 3 种算法中最节能的。

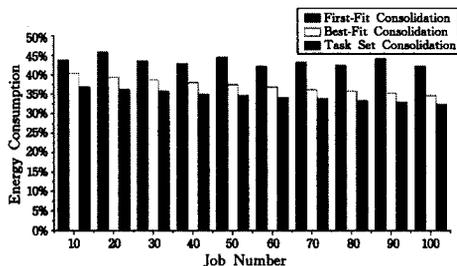


图 4 能量消耗对比

图 5 为活跃服务器数目图。如图 5 所示，活跃物理服务器个数随着任务数目的增加而增加。由于 FF 算法、BF 算法和 TSC 算法都会对有整合条件的任务集进行彻底的整合，因此这 3 种算法的活跃物理服务器数目是相同的。总的来说，任务集的整合能够大大减少物理服务器的使用数目。

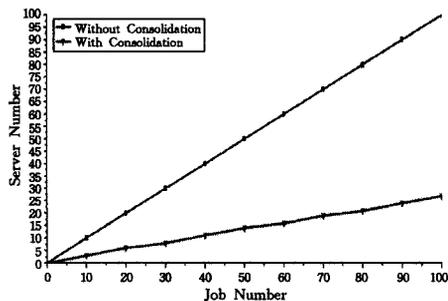


图 5 活跃服务器数目

**结束语** 本文分析了面向车联网应用的云平台的执行特点，建立了云平台模型；分析了车联网任务集的特点，建立了车联网任务集模型；分析了系统的能量消耗，建立了能耗模型，结合上述 3 种模型提出了一个节能整合算法——任务集整合算法(TSC)。本文结合实验证明了提出的 TSC 算法在保证性能的前提下，能够大幅降低车联网云平台的能量消耗，并且减少活跃物理服务器的数目。下一步的工作重心是将本文的任务集整合算法(TSC)应用到实际的面向车联网应用的云计算平台上。

### 参考文献

[1] 中国信息产业网——车联网 [OL]. [http://www.cnii.com.cn/lhrh/node\\_31121.htm](http://www.cnii.com.cn/lhrh/node_31121.htm)

[2] 沈建华. 基于物联网的车联网技术[C]//中国通信学会 2011 年光缆电缆学术年会论文集. 中国通信学会委员会, 2011: 82-89

[3] Yao Wei-hong, Huang Xiao-yuan, Fang Ren-xiao. Cloud Platform Task Scheduling Algorithm Based on Internet of Vehicles [J]. Computer Simulation, 2014, 31(10): 165-169(in Chinese)

姚卫红, 黄小远, 方仁孝. 基于车联网应用的云平台任务调度算法 [J]. 计算机仿真, 2014, 31(10): 165-169

[4] Tan Yi-ming, Zeng Guo-sun, Wang Wei. Policy of Energy Optimal Management for Cloud Computing Platform with Stochastic Tasks [J]. Journal of Software, 2012, 23(2): 266-278 (in Chinese)

谭一鸣, 曾国荪, 王伟. 随机任务在云计算平台中能耗的优化管理方法[J]. 软件学报, 2012, 23(2): 266-278

[5] Ye Ke-jiang, Wu Zhao-hui, Jiang Xiao-hong, et al. Power Management of Virtualized Cloud Computing Platform [J]. Chinese Journal of Computers, 2012, 35(5): 1262-1285(in Chinese)

叶可江, 吴朝晖, 姜晓红, 等. 虚拟化云计算平台的能耗管理 [J]. 计算机学报, 2012, 35(5): 1262-1285

[6] Beloglazov A, Abawajy J, Buyya R. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing [J]. Future Generation Computer Systems, 2012, 28(5): 755-768

[7] Liu H, Jin H, Xu C Z, et al. Performance and Energy Modeling for Live Migration of Virtual Machines [J]. Cluster Computing, 2013, 16(2): 249-264

[8] Zhu Q, Zhu J, Agrawal G. Power-aware Consolidation of Scientific Workflows in Virtualized Environments: Technical Report OSU-CISRC-6/10-TR14[R]. The Ohio State University, 2010

[9] Lee Y C, Zomaya A Y. Energy Efficient Utilization of Resources in Cloud Computing Systems [J]. The Journal of Super Computing, 2010, 60(2): 268-280

[10] Beloglazov A, Buyya R. Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers [J]. Concurrency and Computation: Practice and Experience, 2012, 24: 1397-1420

[11] Berral J L, Goiri I, Nou R, et al. Towards Energy-aware Scheduling in Data Centers Using Machine Learning [C]//Proceedings of the 1<sup>st</sup> International Conference on Energy-Efficient Computing and Networking (e-Energy'10). Passau, Germany, 2010: 215-224

[12] Deng Dan-ting, Teng Fei, Li Tian-rui, et al. An Energy-saving Algorithm for MapReduce-based Virtual Cluster [J]. Computer Engineering and Science, 2014, 36(11): 2054-2060(in Chinese)

邓聃婷, 滕飞, 李天瑞, 等. 基于 MapReduce 虚拟集群的能耗优化算法[J]. 计算机工程与科学, 2014, 36(11): 2054-2060

[13] Chase J S, Anderson D C, Thakar P N, et al. Managing energy and server resources in hosting centers [J]. ACM SIGOPS Operating Systems Review, 2001, 35(5): 103-116

[14] Li Qiang, Hao Qin-fen, Xiao Li-min. Adaptive Management and Multi-Objective Optimization for Virtual Machine Placement in Cloud Computing [J]. Chinese Journal of Computers, 2011, 34(12): 2253-2264(in Chinese)

李强, 郝沁汾, 肖利民. 云计算中虚拟机放置的自适应管理与多目标优化[J]. 计算机学报, 2011, 34(12): 2253-2264