

一种云计算操作系统 TransOS: 基于透明计算的设计与实现

张尧学, 周悦芝

(清华大学计算机科学与技术系, 清华信息科学与技术国家实验室, 北京 100084)

摘 要: 云计算正成为计算机研究和应用的热点. 在云计算研究的许多问题中, 云计算操作系统的研究受到了广泛重视. 但是, 至今为止, 人们对什么是云计算操作系统? 怎样研制云计算操作系统? 以及为什么要有云计算操作系统等问题并没有给出答案, 也没有看到成功的云计算操作系统实例. 本文从透明计算, 也就是用户控制的云计算 (Customer Controlled Cloud Computing) 角度出发, 以用户端提供用户服务、网络服务器提供程序和数据的存储、对网络软硬件资源进行管理来定义云计算操作系统. 我们把传统操作系统, 例如 Linux、Windows 等也定义为资源, 把云计算操作系统定义为运行在传统操作系统与计算机主板 BIOS 之间, 对包含各种传统操作系统在内的网络资源进行管理的超级操作系统 (Meta OS). 本文首先介绍什么是透明计算和基于透明计算的云计算操作系统, 然后给出该系统的结构和相关实现实例.

关键词: 透明计算; 云计算操作系统; 云计算

中图分类号: TP302

文献标识码: A

文章编号: 0372-2112 (2011) 05-0985-06

A New Cloud Operating System: Design and Implementation Based on Transparent Computing

ZHANG Yao-xue, ZHOU Yue-zhi

(Department of Computer Science and Technology, Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing 100084, China)

Abstract: Cloud computing has become a hot topic recently. Among many research issues of cloud computing, CCOS (cloud computing operating system) has attracted extensive attention. However, to date, there is no answer to such related issues: What is a cloud computing operating system? Why we need it? How to develop a CCOS? And no successful instance of CCOS has been witnessed. In this paper, we introduce a CCOS from the viewpoint of transparent computing, i.e., a customer-controlled cloud computing, in which the client machines serve users' computing and services, and the networked servers provide the storage of programs and data. The CCOS manages all the networked and virtualized hardware and software to provide a unified service for users. Traditional operating systems, such as Linux and Windows are also defined as resources in the CCOS, which runs between the traditional operating system and the BIOS of computer motherboard. Therefore, the CCOS is a super operating system (Meta OS), which manages a variety of networked resources, including traditional operating systems. This paper first introduces the concept of transparent computing and the Meta OS based on transparent computing, and then gives the layered structure of Meta OS and one of its implementation examples.

Key words: transparent computing; operating systems; cloud computing

1 引言

云计算^[1]正成为新的计算机研究和产业发展重要热点. 这是一种由谷歌等公司于 2008 年首先提出的网络计算新模式. 这种模式的主要特点在于通过网络, 把

原来在单台计算机中软件的运算和存储分开到用户终端机和网络服务器上完成. 即, 这种模式把主要软件 (含系统程序、应用程序和数据) 存储在网络上, 而用户的终端机只负责和用户的交互以及获取服务, 为用户提供服务的运算既可在网络上的某台计算服务器上完成, 也可

在用户终端机上完成。

至今为止,谷歌、IBM、亚马逊等公司都推出了自己的云计算模式和产品^[2]。特别是谷歌的 GFS^[3]和 MapReduce^[4]分别给出了其大规模网络文件系统的存储结构和快速搜索算法。但是,在网络化的云计算操作系统方面,则还未见有相关研究成果报道。尽管谷歌公司于2009年推出了面向终端机的操作系统 Chrome OS^[5],但该系统只是为终端机提供相应的网络服务,并不管理网络软硬件资源。

本文从用户服务和以网络为基础管理系统软硬件资源的角度出发,提出一种把传统操作系统、应用程序与文件/数据也看做资源,通过网络对这些资源进行调度、分配、回收和控制,把资源的存储和运算分开管理的云计算操作系统(TransOS)。由于 TransOS 是在各种传统操作系统的底层分布式运行,因此,它使得用户可以自主选择自己需要的操作系统(例如 Windows 或 Linux),以及其上的各种应用服务,使不同厂家和不同标准的操作系统可在同一用户终端上兼容运行;同时,这种模式减少用户终端的存储负载、管理难度以及增加用户使用的方便性和安全性。

2 操作系统

操作系统是管理和控制计算机系统中软硬件资源的系统软件^[6]。操作系统的发展主要有两条技术路线:

(1) **单机操作系统**。基于冯·诺依曼结构、以单机为单位的资源管理和程序运行平台。包括从最初的批处理系统,到后来的多道程序系统、分时系统和有网络功能的操作系统等。这类操作系统的典型代表有 Windows, Linux, Symbian 等。由于这类系统的资源管理以单机为单位,当应用需求增加时,其功能就会越来越复杂,系统程序就会越来越庞大,安全性降低。这导致用户操作复杂、管理维护困难。

(2) **网络操作系统**。这类操作系统以网络作为资源管理和程序运行的平台,大多是在单机操作系统基础上加上协议和管理软件构成。典型代表有 Windows 7^[7]、Android^[8]、Chrome OS^[5]等。与单机操作系统相比,网络操作系统具有前端功能轻载化、虚拟化、移动化和集中化等特点,方便了用户的管理和维护、提高了易用性和安全性。

但是,无论是单机操作系统,还是网络操作系统,它们都存在着系统过于复杂和庞大、不支持跨硬件和软件平台等问题。例如,支持桌面软件的操作系统一般不支持移动软件;支持 Windows 的应用软件也不能在 Linux 上运行。这给用户从网络上获得服务和管理带来了不便和风险,也提高了成本和功耗;不适合云计算模式的大规模异构存储的管理和为用户提供所需要

的主动服务。

Chrome OS 试图解决上述问题。但它只提供了一个类似于浏览器的简单桌面环境。它的启动速度很快,也能很快访问后端存储的大规模数据库和获得计算服务。但是,Chrome OS 不支持跨平台操作和访问,因而,还不能把 Chrome OS 看做云计算操作系统的解决方案。

3 透明计算

透明计算是一种为用户提供服务的计算^[9]。其目标是在系统为用户提供服务时,让用户不用知道计算过程和实现细节,只需关心系统能否提供所需服务和服务质量。为了实现这一目标,透明计算通过网络把运算和存储分离,服务器存储操作系统、应用程序以及用户数据等软件资源,终端接近裸机,只存储最底层的 BIOS 和极少部分协议和管理程序。用户服务需要的操作系统和应用程序则通过本文后续的分布式超级操作系统(Meta OS)从服务器中以分块或以流的方式调度到终端上运行。由于操作系统、应用程序以及用户数据都存储在服务器上,终端是一个近乎裸机的运行平台,因此,终端可以做到多样、轻载、安全、易管理和维护。另外,网络服务器可以存储多种不同操作系统以及无数的应用程序,形成巨大的“云”,从而使用户可以自由地从网上跨软件和硬件平台获得所需服务。

透明计算的核心思想包括:(1)计算和存储分离;(2)提供跨终端、跨操作系统平台的操作和用户可选择的服务;(3)服务器和终端之间通过数据流和块调度交换指令,云计算的超级操作系统对整个网络资源进行一体化管理;(4)程序以流的方式被动态调度到指定的终端或服务器上执行^[10]。

基于透明计算理论,文献[11]提出了透明计算体系结构 4VP⁺,并研制了相应的终端系统。Intel 公司又进一步把新一代 BIOS:EFI (Extensible Firmware Interface)^[12]和透明计算结合,开始进行 BIOS 层开放的研究^[13]。

4 一种云计算操作系统 TransOS

4.1 TransOS 操作系统的概念

如前所述,单机操作系统只能管理单机的有限资源。现有的网络操作系统还只是在单机的基础上加上对网络的支持,因此它们不能为用户提供需要的主动服务。在云计算环境下,需要一种能够管理网络软硬件资源,并为用户提供服务的超级操作系统。为此,基于透明计算,我们设计了一种层次化、分布式内核的云计算操作系统 TransOS。与单机和现有的网络操作系统相比,TransOS 能够统一管理由各种异构终端、服务器、网络、以及传统操作系统等组成的网络化资源,并能主动为用户提供按需服务。TransOS 具有如下几个特点:

(1)将传统操作系统也作为一种资源进行管理

TransOS 管理的资源分为三类:第一类是指由终端(群)、服务器(群)和互连网络组成的硬件平台资源;第二类是指在上述硬件实体资源基础上,通过设备驱动和虚拟化,形成可供 TransOS 管理的虚拟化资源;第三类是软件/数据资源,包括传统操作系统、应用程序和数据等.因此,与传统操作系统只管理和调度应用程序和数据不同,在 TransOS 中,传统操作系统,如 Linux, Windows 等本身也是一个被管理和调度的对象.在用户进行选择后,相应的传统操作系统会被 TransOS 调度到用户终端上执行,完成用户所要求的计算任务. TransOS 的任务就是统一管理和调度上述的三类资源,形成一个基于透明计算的云计算超级操作系统.为方便起见,作如下定义:

定义 1 网络化资源 包括由 TransOS 管理的硬件资源,如终端(群)和服务(群)等计算和存储设备,路由器、交换机、基站等网络设备;以及传统操作系统、开发和管理工具、应用软件、结构化和非结构化数据等软件或数据资源.

定义 2 自治网络系统 由终端(群)、服务器(群)、以及网络交换机等硬件设备组成的能够在其上运行 TransOS 的网络系统.

定义 3 实例操作系统 指由 TransOS 管理的传统操作系统,如 Windows, Linux, Sybian, Android, Windows Mobile 等.

(2)分布式内核,动态调度运行

TransOS 的内核分布在用户终端和服务(群)上.如图 1所示,TransOS 内核的很小一部分,驻留在终端 BIOS 上,其它内核部分则驻留在服务器上.需要注意的是,图中只画出了一(终端)对一(服务器)的情形.在实际中,会存在多对多的情形.当终端启动后,驻留在 BIOS 中的 TransOS 内核部分负责获得终端控制权,并与服务器联系,获得系统运行所需要的更多内核代码,并动态以块流的方式加载到终端侧执行^[10].块流的含义是指每次只按需从网络服务器上获取一小部分代码或者数

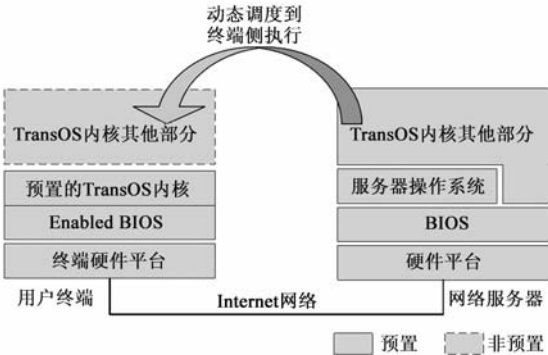


图1 分布式内核、动态调度执行

据块到终端上进行计算,如此循环反复,形成从网络服务器到终端的流.需要说明的是,TransOS 上面支持的实例操作系统、应用程序以及数据等也都是以块流的方式被动态调度到终端侧执行的.

(3)运行在 BIOS 之上,实例操作系统之下

如图 2 所示,TransOS 通过 EFI (Extensible Firmware Interface)等标准接口实现对不同终端硬件架构,如 x86、ARM、MIPS 等的支持. TransOS 在由 BIOS 负责启动运行后,会与服务器联系,将网络服务器能够支持的实例操作系统显示给终端用户进行选择.在用户选择后,再由它从服务器上以块流方式调度用户所选择的实例操作系统到终端上执行.

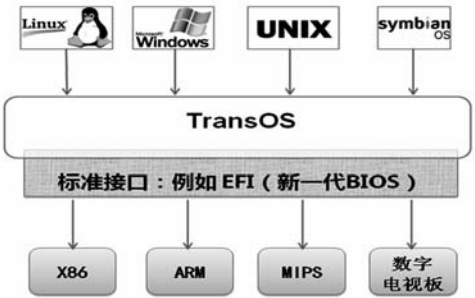


图2 TransOS与BIOS以及传统操作系统的关系

(4)分布式网络化资源的统一管理和调度

如图 3 所示,TransOS 管理的是一个跨网络、跨终端和服务(群)平台的自治网络系统.为了能够统一管理分布在不同地方的网络软硬件资源,TransOS 将其功能按照运行的地点,分为 TransOS 客户端和 TransOS 服务器端. TransOS 客户端和服务(群)端通过协议通信,共同维护和管理分布在不同地点的网络软硬件资源,包括终端硬件、服务器硬件、网络、实例操作系统、应用程序以及数据等,并按照用户需求,统一调度相关资源,以供用户使用.一般而言,TransOS 客户端运行在用户终端设备上,而 TransOS 服务器端则运行在网络服务器上.

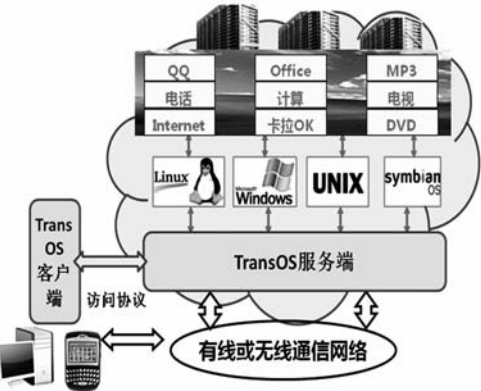


图3 分布式网络资源的统一管理

4.2 TransOS 操作系统的设计

为了实现对分布式网络软硬件资源,包括实例操作系统等的统一管理和调度,TransOS 被设计成一种层次式的模块结构,如图 4 所示,共由 7 个模块组成.需要说明的是,每个模块都是按照需要分布在一台或者多台设备上,共同完成某项功能.

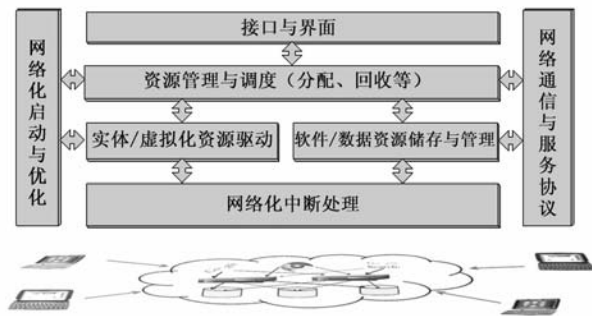


图4 TransOS的系统结构

下面分别对每个模块的功能进行说明.网络化中断处理模块是对来自虚拟设备(在一台计算机的硬件设备基础上形成的可供其他计算机操作的设备)的中断进行处理的模块.该模块能够接受来自自治网络系统中任何一台计算机的网络中断并进行处理,完成该中断请求的相应功能.图 5 描述了 TransOS 与传统单机操作系统在中断处理上的区别.对于传统单机操作系统,出现诸如缺页中断、键盘中断等中断后,CPU 保存当前的运行状态,开始运行相应的中断处理函数处理该中断事件,处理结束后恢复之前的运行状态,继续运行.对于 TransOS,如果是由位于 TransOS 客户端计算机上的虚拟设备发出的中断,则该中断事件并不在 TransOS 客户端上进行处理,而是需要通过网络传输给服务器,交由 TransOS 服务器端进行处理,处理结束后,再通过网络将处理结果返回给 TransOS 客户端.此后,TransOS 客户端 CPU 恢复中断前的运行过程.

实体/虚拟化资源驱动模块,是在网络化中断管理的基础上,完成对实体/虚拟化资源的驱动和管理.具体包括如下三个方面的功能:(1)对物理的硬件设备,如网卡、显卡等进行驱动,完成自治网络系统中所有硬件设备的驱动;(2)在实体硬件设备驱动基础上,进一步形成虚拟化资源,如对虚拟 CPU、虚拟内存、虚拟存储等虚拟设备资源进行构建、驱动和管理;(3)实现对大规模软件/数据的存储支持,其功能类似于 Google 公司的 GFS 和 Bigtable^[14],实现对大规模数据的网络化存储和检索.软件/数据资源存储与管理模块是在虚拟化网络存储的基础上,进一步将实例操作系统、应用程序以及数据进行组织、存储和管理,供用户进行高效访问.

TransOS 的核心功能是对网络化资源进行管理和调度,也就是进行资源的调度、分配和回收等工作.这是

由资源管理与调度模块来实现,该层功能类似于传统单机操作系统中的内核调度功能.为满足不同网络化资源的调度和管理要求,需要针对不同属性的资源,研究基于不同策略和方法的调度、分配和回收算法,实现资源的按需弹性流动.与单机操作系统类似,TransOS 也具有接口与界面模块,用于实现与应用程序以及用户的交互,接收并完成来自上层应用和用户的请求.

与单机操作系统不同的是,TransOS 具有两个特殊的模块,网络化启动与优化模块以及网络通信与服务协议模块.其中网络化启动与优化模块的客户端主要完成终端启动中断的截获,并将此信号传递到网络服务器,由服务器来进行处理.网络化启动与优化模块的服务器端会响应来自客户端的请求,首先是给客户端分配一个网络地址,便于后面的通信;其次,会对用户的身份进行认证并将网络自治系统支持的实例操作系统列表返回给用户进行选择;在用户选择自己想要运行的实例操作系统后,再调用相关的传输协议将部分的实例操作系统内核加载到终端内存中执行,完成实例操作系统的启动.网络通信与服务协议模块主要完成实例操作系统启动以后,如何依照用户的使用,按需将其他实例操作系统模块以及应用程序代码和数据按需以块流的方式传送到终端内存中执行.与传统操作系统中通信协议模块只为应用程序间通讯服务所不同的是,TransOS 中的通信协议传送的不仅是数据,还有代码;同时,TransOS 中的通信协议在实例操作系统未启动的情况下就会工作.

5 TransOS 的实现示例

TransOS 有多种实现方式.下面介绍一种基于 4VP⁺ (4 Virtual Layers, 2 Protocols) 在普通 PC 计算机板上基于 EFI BIOS 的实现示例.在这种实现下,TransOS 客户端和服务器的功能均被划分为四个虚拟层和两个协议,如图 6 所示.

两个协议分别是多操作系统远程启动协议(MRBP, Multi-OS Remote Booting Protocol)和网络服务访问协议(NSAP, Network Service Access Protocol).MRBP 的功能是捕获客户端启动初始化的中断信号,发现服务器存储的实例操作系统列表,下载部分操作系统内核以及 NSAP 协议客户端部分.因此,MRBP 客户端代码必须提前驻留在终端主板 BIOS 上.由于 MRBP 工作在实例操作系统启动之前,因此 MRBP 客户端不能像传统协议一样利用操作系统的网络通信接口,而需要实现自己的网络设备驱动和网络协议栈.NSAP 协议的主要功能是传输 TransOS 以及上层实例操作系统的指令、数据、中断和响应等.

4VP⁺ 中的四个虚拟层分别是:虚拟网络 IO 管理(VNIOM, Virtual Network I/O Management),虚拟网络设备

管理(VNDM, Virtual Network Device Management), 虚拟网络文件管理(VNFM, Virtual Network File Management)以及

虚拟网络用户管理(VNUM, Virtual Network User Management). 下面分别介绍各个虚拟层的功能.

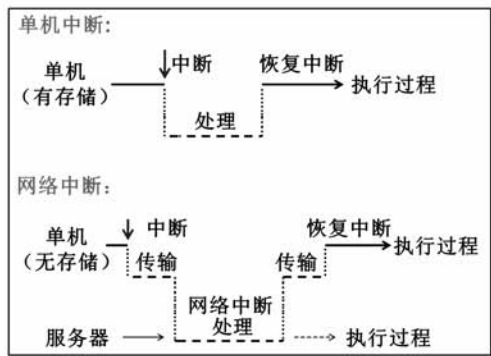


图5 单机中断与网络中断的处理

(1)VNIOM.其主要功能是接收终端上运行的实例操作系统的 I/O 中断请求,分析中断原因,并唤醒相应的处理程序进行处理.与单机环境下不同的是,VNIOM 需要将终端的中断请求送到服务器进行处理,并将处理结果返回给终端.由于服务器需要同时处理来自不同终端的中断请求,因此 VNIOM 需要管理多级队列,以快速响应高优先级的中断请求.

(2)VNDM.其主要功能包括:完成虚拟设备的分配与回收;完成客户端虚拟设备到服务器虚拟镜像的映射及管理;完成客户端与服务器的虚拟交换与调度.

(3)VNFM.这一层首先是完成类似于 Google GFS 的大规模分布式文件系统的存储、访问与搜索;其次,还需要完成如下辅助功能:完成用户文件空间、文件目录的分配与管理;实现用户文件访问的重定向与虚拟化;保证文件访问的一致性;提供用户文件的访问控制、搜索等功能.

(4)VNUM.其主要功能包括:完成增加、减小、配置用户信息的工作;进行用户身份认证;对用户使用过程进行监管等.

我们已经在基于桌面终端和普通 PC 服务器集群上实现了局域网范围内的 4VP*,并已经进行了产业化

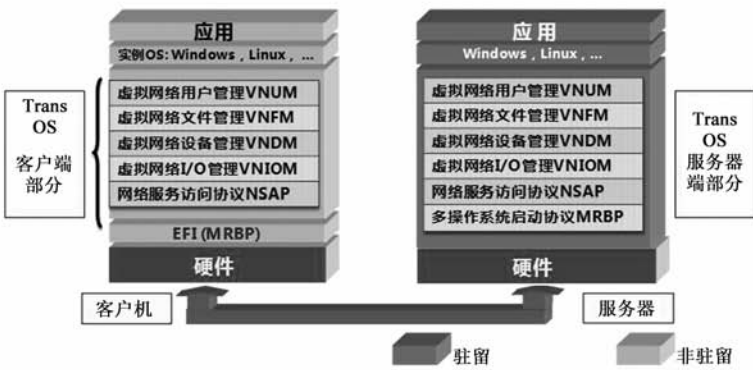


图6 4VP*的组成

生产和实际部署.目前,正在研制基于平板电脑、手机等终端和移动互联网的 TransOS.此外,透明计算模式和 TransOS 也可以在家电等其他设备上得到应用.

6 相关工作

近年来,为了支持云计算,谷歌公司开发了 GFS^[3]、BigTable^[14]和 MapReduce^[4],亚马逊公司开发了 EC2^[15]和 S3^[16].其他公司和研究机构也开发了相应的云计算技术和产品^[17,18].但是些工作都是在传统单机操作系统基础上来实现的分布式系统,并没有讨论用于云计算的操作系统问题.

表 1 展示了 TransOS 与 Chrome OS 的详细比较. Chrome OS^[5]以网络作为开发和运行的平台,通过轻载的前端界面,利用强大的后端“云”,给用户方便的服务,是云计算操作系统的一个示例.Chrome OS 强调前端轻载,后台支撑.在这一点上,TransOS 和 Chrome OS 的思路是类似的.Chrome OS 通过本地轻载的操作系统内核加上一个浏览器,将存储在网络上的 Web 应用程序下载到本地浏览器运行,实质上还是一个传统单机操作系统的开发思路.与 Chrome OS 不同,TransOS 利用开放的 BIOS 技术和一个轻载的固件层模块将存储在网络上的

表 1 TransOS 与 Chrome OS 的异同点

	TransOS	Chrome OS
硬件平台	x86, MIPS, 适用于 PC, 服务器, 上网本, 手机等	X86 和 ARM, 只适用于上网本
网络支持	LAN, WAN, 3G 等	WAN
类型	轻载操作系统, 分布式内核/组件 + Web 浏览器	轻载操作系统, 操作系统内核 + Web 浏览器
网络传输内容	操作系统环境 (包括操作系统, 应用程序, 数据, 状态等)	应用
操作系统存储位置	网络	本地
应用程序存储位置	网络	网络
支持的应用程序类型	本地应用, Web 应用	Web 应用
操作系统维护和管理	远程更新, 集中管理	本地只读
应用程序维护和管理	集中式管理	集中式管理
管理对象	操作系统	应用程序
编程模式	Web 编程 + 传统编程	Web 编程
向后兼容性	既可以支持传统应用, 又可以支持 Web 应用	只能支持 Web 应用
用户体验	可以保持用户体验	改变用户体验

实例操作系统及其应用程序调度到本地运行,因此是一种网络化操作系统的思路.其次,Chrome OS 只能支持 Web 应用,而 TransOS 支持多种操作系统及其应用程序.

7 结束语与未来工作

云计算操作系统是云计算领域研究的核心问题之一.针对目前操作系统只能管理单机软硬件资源的问题,本文提出了一种基于透明计算的云计算操作系统 TransOS. TransOS 内核分布在由终端、服务器和网络设备组成的自治网络系统上,能够集中管理包括传统操作系统在内的各种网络化软硬件资源,并通过按需调度用户所需资源,完成用户所需要的计算和服务.

未来的工作包括:(1)探索如何将基于透明计算的云操作系统扩展并应用到更多的设备(如数字家电)和网络(如 3G, LTE 网络)上去;(2)建立相应的云操作系统结构和接口规范;(3)在实际应用中进一步完善相关理论和技术.

参考文献

- [1] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, Matei Zaharia. A view of cloud computing[J]. Communications of the ACM, 2010, 53(4): 50 – 58.
- [2] 陈康, 郑纬民. 云计算: 系统实例与研究现状[J]. 软件学报, 2009, 20(5): 1337 – 1348.
- [3] Sanjay Ghemawat, Howard Gobioff, Shun-Tak Leung. The Google file system[A]. Proc of the 19th ACM Symposium on Operating Systems Principles (SOSP'03) [C]. New York: ACM, 2003. 29 – 43.
- [4] Jeffrey Dean, Sanjay Ghemawat. MapReduce: Simplified data processing on large clusters[A]. Proc of the Sixth Symposium on Operating System Design and Implementation (OSDI'04) [C]. San Francisco, CA: USENIX. 2004. 137 – 150.
- [5] S. Pichai, L. Upson. Introducing the Google Chrome OS[EB/OL]. <http://googleblog.blogspot.com/2009/07/introducing-googlechrome-os.html>, 2011-03-18.
- [6] 张尧学, 史美林, 张高. 计算机操作系统教程(第 3 版)[M]. 北京: 清华大学出版社, 2006.
- [7] Mike Nash. "Why 7". The Windows Blog[EB/OL]. <http://win-dowsteamblog.com/windows/archive/b/windowsvista/archive/2008/10/14/why-7.aspx>, 2011-03-18.
- [8] Sharon P. Hall, Eric Anderson. Operating systems for mobile computing[J]. Journal of Computer Sciences in Colleges, 2009, 25(2): 64 – 71.
- [9] 张尧学. 透明计算: 概念、结构和示例[J]. 电子学报, 2004, 32(12A): 169 – 173.

Zhang Yaoyue. Transparence computing: Concept, architecture

and example[J]. Acta Electronic Sinica, 2004, 32(12A): 169 – 173. (in Chinese)

- [10] Zhou Yuezhi, Zhang Yaoyue. Transparent Computing: Concepts, Architecture, and Implementation[M]. Singapore: Cengage Learning, 2009.
- [11] Zhang Yaoyue, Zhou Yuezhi. 4VP+: A novel meta OS approach for streaming programs in ubiquitous computing[A]. Proc of the IEEE 21st International Conference on Advanced Information Networking and Applications (AINA'07) [C]. Niagara Falls, Canada: IEEE Computer Society Press, 2007. 394 – 403.
- [12] Extensible Firmware Interface (EFI)[EB/OL]. <http://www.intel.com/technology/efi/>, 2011-03-18.
- [13] 清华大学. EFI 平台创新架构技术合作备忘录[Z]. 2007-02-08.
- [14] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, et al. Bigtable: A distributed storage system for structured data[A]. Proc of the Seventh Symposium on Operating System Design and Implementation (OSDI'06) [C]. Seattle, WA: USENIX, 2006. 205 – 218.
- [15] Amazon. Amazon Elastic Compute Cloud User Guide (API Version 2011-01-01) [EB/OL]. <http://docs.amazonwebservices.com/AWSEC2/latest/UserGuide/>, 2011-03-18.
- [16] Amazon. Amazon Simple Storage Service (Amazon S3)[EB/OL]. <http://aws.amazon.com/s3/>, 2011-03-18.
- [17] David Chappell. Introducing the Azure Services Platform[EB/OL]. http://download.microsoft.com/.../e/.../Azure_Services_Platform.docx, 2011-03-18.
- [18] Dhruba Borthakur. The Hadoop Distributed File System: Architecture and Design[EB/OL]. http://hadoop.apache.org/common/docs/r0.18.0/hdfs_design.pdf, 2011-03-18.

作者简介



张尧学 男, 1956 年生, 教授, 中国工程院院士, 主要研究方向为透明计算和主动服务.

E-mail: zyx@moe.edu.cn



周悦芝 男, 1975 年生, 副研究员, 主要研究方向为透明计算和云计算.

E-mail: zhouyz@mail.tsinghua.edu.cn