因特网上基于节点角色的计算资源共享平台——RB-CRSP

张学锋¹, 徐胜超²

(1. 惠州学院 数学系, 广东 惠州 516007; 2. 广东邮电职业技术学院 计算机系, 广州 510630)

(isdooopen@126.com)

摘要:提出了因特网上基于节点角色的计算资源共享平台——RB-CRSP。设计时充分考虑节点的角色性和功能性,把因特网上的网络资源按照角色划分为服务器端节点、协调节点、工作机节点与客户机节点四类实体,通过配合 RB-CRSP 的应用编程模式,完成并行分布式计算。分析了 RB-CRSP 中的自适应资源调度策略,该策略考虑了节点的硬件信息与可信誉机制,实现了平台的负载均衡性;在动态的因特网环境下,利用面向工作机的容错方式保证了平台的可靠性。案例程序选择了典型的并行 BenchMark 程序: *N* 皇后问题,测试结果表明, RB-CRSP 可以方便聚集异构环境下的空闲计算资源,平台的性能与机器硬件条件和可靠性密切相关。

关键词:分布式计算;因特网计算;资源共享;并行处理;编程模式

中图分类号: TP338.8 **文献标志码:** A

RB-CRSP: Node role-based computing resources sharing platform over Internet

ZHANG Xue-feng¹, XU Sheng-chao²

(1. Department of Mathematics, Huizhou University, Huizhou Guangdong 516007, China;

2. Department of Computer, Guangdong Vocational College of Posts and Telecom, Guangzhou Guangdong 510630, China)

Abstract: A role-based computation resource sharing platform called RB-CRSP was proposed in this paper. The node roles and functions were strongly considered in design topology of RB-CRSP. All the nodes in RB-CRSP were assigned as server, coordinator, worker, and client. By using the united programming model, it could be in harness with the immense computational resource available in the Internet for parallel and distributed computation efficiently. RB-CRSP platform was also load balancing and fault-tolerant, which was strongly supported by hardware and credit standing based scheduling algorithm and the workers oriented fault tolerance policy. In order to demonstrate the effectiveness of RB-CRSP, a serial of simulation experiments were done. The results obtained from performance analysis show that RB-CRSP is feasible and efficient which can provide a new way for computing resource sharing over the Internet. The hardware conditions and availability of machines are important factors in this test.

Key words: distributed computing; Internet computing; resources sharing; parallel processing; programming model

0 引言

计算资源共享平台(Computing Resource Sharing Platform, CRSP)的研究目标是构造一个在大规模因特网环境下的虚拟计算环境,该环境应该具有科学应用的无关性、可扩展、跨平台运行、界面友好、方便程序员开发等特点。为了达到这些目标,近年来很多相关的项目都取得了有价值的研究成果^[1-5]。该问题至关重要,其研究技术点包括计算资源共享平台的网络拓扑架构、节点的组织与通信方式、任务与工作单元的调度方式、应用的编程模型、数据的传输协议研究等方面。

目前国内外在实现 CRSP 时,往往是通过中间件软件的方式把各种异构的、多域网络环境下网络资源充分利用与聚集起来,完成某种分布的、计算量大的、计算密集型的任务,某些文献上也称其为桌面网格(Desktop Grid)^[6-8]。它利用 Java 语言的对象序列化技术与字节码技术而开发,具有容易把规模扩大到因特网上与应用推广的特点。

RB-CRSP (Role-based Computing Resource Sharing Platform)是本文提出的一种按照节点角色功能划分的计算资源共享平台。和目前的 CRSP 比较起来, RB-CRSP 的节点层

次结构比较适合并行计算任务的划分与汇总;采用统一的并行应用作业描述手段及编程模型,使它更容易应用推广;该平台的可扩展性得益于建立在应用层上的层次消息通信网络,使其可以方便组织因特网上的节点;高效率性建立在针对硬件信息与可信誉机制的自适应调度算法及面向工作机的容错策略。

最后对 RB-CRSP 的性能进行了实验与分析,通过运行典型的并行应用案例程序,实验结果证明了 RB-CRSP 是一种可行的、高效的计算资源共享平台。

1 RB-CRSP 平台的设计

RB-CRSP 把因特网上加入到平台的计算机分成四类:服务器端节点、协调节点、工作机节点和客户机节点。服务器端节点是一个比较稳定的节点,用于启动计算网络的形成及节点的加入。协调节点作为平台的调度器,负责对工作单元进行分发和调度。工作单元是粗粒度并行的一个概念,它是高于指令级并行与循环级并行的大块的并行程序。大规模因特网环境下的空闲资源构成了工作机节点,具体实现工作单元的计算。设计把服务器端节点、协调节点和工作机节点形成

收稿日期:2010-09-13;修回日期:2010-11-03。

作者简介:张学锋(1968-),女,甘肃民勤人,讲师,主要研究方向:计算机网络、网络安全;徐胜超(1980-),男,湖北武汉人,讲师,硕士,CCF 会员,主要研究方向:并行与分布式计算、对等计算。

了一个可扩展的层次网络体系结构, RB-CRSP 平台的若干个协调节点必须依附于一个服务器端节点, 若若干个工作机节点必须依附于一个协调节点。这种有点类似于树型结构的设计既可以减少节点功能组内部与节点功能组之间的消息通信量, 也方便了任务及工作单元的划分与调度, 因为很多并行应用的任务之间的关系都是树型结构。各种角色的节点之间的作业执行流程如图 1 所示。

图1 RB-CRSP 平台的节点角色组织

客户机节点主要实现作业描述文件的形成及提供可执行代码文件与数据文件。主从编程模式是这类平台的主要应用。客户机节点与采用主-从模式编程的主控节点可以是一个节点。

2 应用开发及编程模型

2.1 应用的执行流程

在利用 RB-CRSP 平台进行并行应用程序的执行时, 平台中的各种实体的节点合作与交互过程如图 2 所示。

图2 RB-CRSP 平台中节点执行任务的过程

1) 当一个协调节点加入到 RB-CRSP 平台时, 首先与平台中的任意一个服务器端节点进行 TCP 连接, 连接的过程就是

注册的过程, 把自己的硬件信息 (CPU 主频、磁盘空间、内存大小和 CPU 利用率等) 与软件信息 (节点的 ID 号、IP 地址、端口、可以执行的工作单元数和预计的空闲时间等) 提交给服务器端节点, 这个过程需要一些基本的安全验证。

2) 服务器端节点在收到协调节点的注册请求后, 将检查协调节点所提供的信息的合法性, 如果协调节点通过检查, 将被增加到该服务器端节点下属协调节点队列。

3) 当一个工作机节点加入到 RB-CRSP 平台时, 它首先与服务器端节点进行 TCP 连接, 服务器端节点根据下属的协调节点的状态, 为其分配一个协调节点, 该工作机节点再与刚刚被分配好的协调节点进行连接, 提交其硬件信息与软件信息, 这些信息与上面的协调节点一样, 其中节点的 ID 号是自动产生的, 用于以后调度时执行工作单元及与协调节点之间的通信时作为身份检查, 工作机随后也被增加到协调节点下属的工作机节点队列中。

4) 服务器端节点返回一个其下属的已经向它注册过的活跃的协调节点的列表。

5) 当一个客户机节点加入到 RB-CRSP 平台时, 与任意一个活跃的协调节点进行连接, 提供节点的硬件信息与软件信息, 软件信息中最重要的是其需要的工作机节点的数目。如果该活跃协调节点下属的工作机节点数目能够满足其要求, 就不再与其他协调节点进行连接; 如果不能满足, 它将与网络中的其他协调节点连接, 直到其寻找到所有工作机为止。

6) 活跃协调节点在接收到客户机节点的请求后, 协调节点将检查客户机提供的信息是否合法。如果客户机通过了合法性检查, 产生一个客户机的 ID 号, 协调节点将为其分派空闲的工作机的列表, 客户机即获得了计算资源, 协调节点将通知其下属的工作机节点为具有此 ID 号的客户机服务。

7) 工作机节点发送一个确认消息给其所依附的协调节点, 客户机节点从活跃协调节点处获得的计算资源信息后, 客户机节点将把任务描述文件 (作业说明书) 等信息提交给工作机节点, 这些信息还包括客户机 ID 及将要执行的工作单元的可执行代码及数据文件。

8) 工作机节点验证这些信息的正确性, 并具体地执行代码, 把计算子结果返回给客户机。从这个过程可以看出, 工作机节点属于计算资源的提供者, 客户机节点属于计算资源的消费者。通过工作机的大量参加, 可以使 RB-CRSP 平台的计算速度提高很快。

2.2 应用的编程模型

RB-CRSP 平台能够广泛推广得益于其可以提供给开发员一套应用编程接口调用, 让其开发并行应用如同串行设计一样简单, 适合多种应用, 编程语言也是程序员所熟悉的 Java 或者 C++, 不需要有 MPI 或者 OpenMP 的并行程序设计思想。下面的例子程序显示了用户是如何进行并行编程的, 这种编程模型包含以下步骤:

```
public static void main (String[] args) { ...
//创建并行应用程序环境
1) RB-CRSP_Env env = new RB-CRSP_Env ( className,
    appClassNames, "SecretKey.ser", director,
    password ); ...
//产生并行任务到工作单元池
2) for ( int i=0; i < numWorkers; i++ ) { ...
    tidarray[i] = env. AddToTaskPool((Object) task) } ...
//分发工作单元给工作机计算
3) env. SendTasks(); ...
```

```

//收集计算完的临时子结果
4) for (int i=0; i<numWorkers, i++) {...
    result[i] = env.getResult(tidarray[i], waitTime); }...
//停止所有未计算完的工作单元,结束计算。
5) env.StopTasks();}

```

1) 创造并行应用环境。在该应用环境下,客户机节点与服务器端节点连接, RB-CRSP 平台为其分配可利用的协调节点,同时产生一个空的链表,用来存放将来要执行的工作单元,链表在某些文献上把其称为工作单元池。

2) 生成工作单元到任务池。具体一个大的任务划分为多少个工作单元由应用程序开发人员所决定,工作单元产生以后,都存储在任务池中。对于每一个新生成的工作单元,应用开发人员为其分配工作单元 ID 号。这个 ID 号在平台中执行中的调度过程与容错都会使用到。

3) 分配工作单元给工作机。完成上面两步以后,客户机节点连接协调节点,申请获得其下属的工作机的计算资源,然后把工作单元的数据文件、参数文件、执行文件分配给工作机节点进行执行,这个功能可以利用一个线程来完成。当客户机获取不到足够的工作机节点计算资源时,或者当有工作机节点异常退出平台时, RB-CRSP 平台将采用一个新的线程 WorkerHouter 来处理,以便获得新计算资源。探测工作机是否异常退出平台可以采用分布式系统的心跳机制 (Heartbeat), 以一个常规的时间间隔来监视其是否异常加入或者退出,这个过程对于应用开发人员是透明的,也是 RB-CRSP 平台的容错机制的一个部分。

4) 收集计算结果。工作机节点计算完的子结果将由一个 ResultCollector 线程来负责接收与存储。协调节点也将通知客户机应该释放先前被分配的工作机计算资源。为了汇总这些计算完的子结果,程序员应该利用第 2 阶段时候生成的工作单元 ID 号进行后续处理。

5) 停止所有未完成的任务。这一步是可选的,因为它依赖于应用程序的需求。如果应用开发人员已获得必要的结果,它可以通过使用一个 StopAllTasks() 方法终止现有的工作单元的执行,如果所有的计算子结果很容易分析,这一步骤可以不需要。

3 调度策略与容错

3.1 自适应调度策略

前面已经提到,面向因特网的分布式计算的并行单元是工作单元,这是一种粒度比较粗的应用,而且大规模因特网环境下节点很容易频繁地加入与退出,如何能够有效地实现调度,把工作单元高效地分配到空闲的工作机节点上运算是这种计算模型的核心问题^[9]。

RB-CRSP 平台的任务调度方式不仅考虑了机器的硬件环境,也考虑了机器的信誉机制,它的思想是尽可能将多的工作单元交给负载最轻的工作机节点。在这种调度方法中,节点的负载考虑了硬件信息与软件信息,硬件信息包括 CPU 的利用率、内存的大小、剩余的磁盘空间等,最直接的方式把 CPU 主频高、内存大、空闲磁盘空间多的机器投入运算。软件信息包括节点的 ID 号、IP 地址和端口等,其中最重要的软件信息是节点完成工作单元个数的历史信息,这些就是信誉机制。

RB-CRSP 平台中的自愿机节点的计算能力 (Computing Capabilities, CC) 用一系列参数来决定,包括 CPU 的主频 CPU_N 、内存的容量 MEM_N 、磁盘空间 $DISK_N$ 、自愿机自身的负载信息 NET_N 和网络延迟 $netDelay$,工作机连续成功完成计算的工作单元数 $WorkUnitCount_N$ 等。它设计了专门负责任务调

度的协调节点,它把平台中所有的执行工作单元计算的工作机节点利用队列来组织,然后按照每个工作机节点的计算能力的大小进行排序。

协调节点在执行工作单元调度时,每次都取出队列中的首部的节点来执行工作单元的计算(计算能力最强)。节点的 CC 由于是动态的,所以分成两种情况来处理:1) 当工作机节点刚刚加入平台时,CC 由节点的硬件信息和网络延迟共同决定。2) 当工作机节点加入平台后,节点计算能力可以定义为 $CC = W1 * CPU_N + W2 * MEM_N + W3 * DISK_N + W4 * NET_N + W5 * WorkUnitCount_N$ 之类的公式,其中 $W1 + W2 + W3 + W4 + W5 = 1$,表示工作机节点中各种硬件资源与信誉机制所占的权重, $W1 \sim W4$ 都是考虑的硬件因素, $W5$ 考虑的是信誉机制, $W1 \sim W5$ 可以由开发人员根据具体情况自己确定。

3.2 容错特性

RB-CRSP 平台的容错方式采用的是面向工作机节点的容错,假设服务器端节点与协调节点是稳健性的,将容错过程简化。

经实践证明导致工作机错误的原因有:1) 工作单元运行程序或数据错误;2) 工作单元运行工作机节点的硬件配置与工作单元运行所需不符,导致程序运行错误。总体来说,这些错误原因产生的根本原因是因特网上工作机节点情况的差异性。

协调节点在正常运行过程中,节点控制器会启动错误监控线程,监控线程不断地扫描工作机的 workerNodeList 队列,通过检查队列内每一个元素(对应一个 worker)的状态位 state (int 型),若状态位 $state < 0$,则表示节点出错,转入错误处理。错误监控线程同时处理工作机队列和容错队列。例如每一个工作机节点工作状态位 state 设计为:1 表示正在运行工作单元;0 表示空闲中;-1 表示出错。

工作机异常退出平台,其对应的工作单元的代码与数据会立刻转移到容错队列中的节点执行,使整个计算结果不受到影响。容错队列的容错状态位对应的位设计为:1 表示已经有新的工作机节点对其容错(即承担其未运行完毕工作单元);2 表示工作机节点需要执行容错(即有未运行完毕的工作单元需要新的工作机节点接替);0 表示不需要执行容错等。

4 实验与性能分析

4.1 评测环境

在构建 RB-CRSP 平台的测试场景时,理想的环境是使用因特网上大规模的各种类型的机器形成一个基于节点角色的计算资源共享平台,但是这种环境很难构造,而且测试的时间很难持续太长,因为不可能让陌生的机器强行运行 RB-CRSP 平台的客户端的 Jar 包,这样某些用户会认为 RB-CRSP 平台是病毒程序。只通过 200 多台 PC 机构建了 RB-CRSP 平台,这些 PC 机都来自于教育网中的某个科研机构,如表 1。各个机器上的 JVMs (Java Virtual Machines) 启动的优先级比较低,往往是通过屏幕保护程序才启动。RB-CRSP 平台的建立过程主要决定于其基于节点角色的层次消息通信网络,完成的是主-从编程模式。

RB-CRSP 平台的所有工作机分为三类子工作组,称为工作组 1、工作组 2、工作组 3,这些机器的台数目为 50 ~ 100。这样来划分是为了更好地分析 RB-CRSP 平台的资源调度性能。工作组 1 中的机器是根据 CPU 的主频大小来选择,即这些都是计算速度比较快的。测试中如果用户觉得计算非常耗 CPU,他们能够手动终止该计算过程。图 3 显示了 RB-CRSP 平台的网络结构,分别包括服务器端、客户机、协调节点与工作机。

表1 RB-CRSP 平台评测环境

节点角色	处理器/GHz	网卡速度/Mbps	机器数目	操作系统	虚拟机 JVMs
服务器端节点	Dual-P4 3.2	100	1	GNU/Linux	J2sdk1.5.0
协调节点	Dual-P4 3.2	100	5	Windows Vista	J2sdk1.5.0
客户机节点	Dual-P4 3.2	100	1	Windows Vista	J2sdk1.5.0
工作组1	Dual-P4 3.2	100	50	GNU/Linux	J2sdk1.5.0
工作组2	Intel P4 3.0	100	80	Windows XP	J2sdk1.5.0
工作组3	Intel Celeron 2.66	100	110	Windows Vista	J2sdk1.5.0

图3 RB-CRSP 平台的网络拓扑结构

4.2 应用实例

并行应用的案例程序选择了 N 皇后问题。该问题是在 $N \times N$ 格的国际象棋上摆放 N 个皇后,使其不能互相攻击,即任意两个皇后都不能处于同一行、同一列或同一斜线上,计算出有多少种摆法(例如 8 皇后有 92 中摆法)。

实验中为了发现 25×25 的表格上皇后的不同摆法,利用网上该算法的 Java 源代码,把整个问题划分为了许多独立的工作单元,采用了主-从的编程模式来分派这些工作单元在 RB-CRSP 平台上执行。客户机启动主控程序为初始化棋盘,工作机分别用来把每个排列下皇后来匹配棋盘看是否合适。 N 皇后问题的测试结果如表 2 所示。

表2 N 皇后问题的计算结果($N = 25$)

计算结果	数值
工作单元总数	12 125 199
测试的整体时间	72 h(完成了大约 210 000 个工作单元)
单台机一个工作单元的 执行时间	132 s(单台机 72 h 只能 完成 1960 多个工作单元)
计算的加速比	$210\,000/1960 = 107.1$
三个工作组的总节点数	240

4.3 性能分析

图 4 显示了计算过程中各种工作机加入到 N 皇后计算问题的情况。在整个计算的过程中,一共有 200 多台 PC 机加入了 RB-CRSP 平台,最多的时候一共有 240 台机器同时工作。图中还表明,RB-CRSP 有时候只有很少的节点参与了计算,原因是网络或硬件失效。

实验中还发现工作组 1 的 20 多个成员完成的工作单元比较多,因为它们的处理器主频高、计算能力很强,而且计算时间较长,连续完成的工作单元多,可信誉机制好,平台中的调度策略正好把大量的工作单元分发给工作组 1 计算,但是这只占该工作组中 50 多个工作组 1 的一半的数目。其余的机器由于负载比较大,JVMs 被用户所强行关闭了。

工作组 1 中的机器是根据 CPU 快慢而选择的,所以其可

以完成大量的工作单元。但是 RB-CRSP 平台中还有许多其他的节点,表 1 中显示了有很多机器主频小于 2.8 GHz,这些机器只完成了很少的工作单元。所有这些实验结果表明,RB-CRSP 平台可以加快 N 皇后问题的计算速度,完成的工作单元的多少与机器的硬件有很大的关系,也与节点的信誉有密切联系。

图4 72 h 内 RB-CRSP 平台节点的情况

5 相关工作

为了实现大规模因特网环境下的计算资源共享平台,目前国内外开发了一些很重要的系统。这些系统可以认为是介于因特网上空闲硬件资源与用户并行应用程序之间的中间件软件,具体包括面向因特网的计算资源共享平台、基于网格的计算资源共享平台、基于对等网络的计算资源共享平台等。

目前比较成熟的是因特网上的大规模计算资源共享中间件软件,包括 XtremWeb 和 BOINC^[3] 等项目。它们已经成功地运用于搜索外星球文明、全球气候预测、生物蛋白质折叠、密码破解等很多类型的应用问题,获取了很多数据与结果。这类平台要成功运行依赖于工作机的广泛参加,而且运行时间相当长,需要研究出某种机制刺激用户的加入。

基于网格的计算资源共享平台包括 Globus^[10]、CoreGRID^[11] 等,这些中间件软件可以共享与聚集异构环境下的计算资源、数据资源、存储资源,但是其对应用问题的限制、访问权限、计算的规模扩大方面往往略差于面向因特网的中间件软件。

另外一些文献是类似于基于 P2P 的科学计算环境 P2P-Grids,或称为桌面网格(Desktop Grid)^[4-8],如 OurGrid^[12]、Alchemi^[13]、ShareGrid^[14] 和 WSpeer^[15] 等项目。该类项目聚集的资源主要是 PC 机,通过运行计算量大的、分布式的应用,来充分利用大规模因特网上的计算资源。但是大部分 P2P 项目的应用领域是实现文件共享与视频播放等,所以其缺少比较完善的资源管理服务,存在着节点的信誉机制与搭便车(Free riding)问题。本文的 RB-CRSP 平台在易用性、通用性、可扩展性及高效调度等方面都有比较突出的优点与性能。

6 结语

本文描述了一个因特网上基于节点角色的计算资源共享平台 RB-CRSP,该平台具有易用性、通用性、可扩展性及高效率性。通过 RB-CRSP 的客户端软件就可以把因特网下的自愿机组织起来,实现大规模计算资源共享,利用 RB-CRSP 平台的编程模型即可以完成应用的分布式计算。通过对 RB-CRSP 平台上的并行 Benchmark 程序的测试,取得了比较好的性能。目前在实验中运行的应用都是主-从模式的并行应用,范围还比较小,如何优化编程模型扩大应用范围是下一步的工作。

参考文献:

- [1] AMORETTI M, ZANICHELLI F, CONTE G. SP2A: A service-oriented framework for P2P-based grids [C]// MGC'05: Proceedings of the 3rd International Workshop on Middleware for Grid Computing. New York: ACM Press, 2005: 1-6.
- [2] SHUDO K, TANAKA Y, SEKIGUCHI S. P3: P2P-based middleware enabling transfer and aggregation of computational resources [C]// Proceedings of 2005 IEEE International Symposium on Cluster Computing and the Grid. Washington, DC: IEEE Computer Society, 2005: 259-266.
- [3] ANDERSON D P. BOINC: A system for public-resource computing and storage [C]// GRID 2004: Proceedings of the Fifth International Workshop of Grid Computing. Washington, DC: IEEE Computer Society Press 2004: 4-10.
- [4] MATOSSIAN V, PARASHAR M. Enabling peer-to-peer interactions for scientific applications on the grid [C]// Proceedings of the 9th International Euro-Par Conference, LNCS 2790. Berlin: Springer-Verlag, 2003: 1240-1247.
- [5] PATOLI Z, GKION M, AL-BARAKATI A, *et al.* How to build an open source render farm based on desktop grid computing [C]// IMTIC 2008: Proceedings of the 2008 International Multi-topic Conference on Wireless Networks, Information Processing and Systems. Berlin: Springer, 2008: 268-278.
- [6] PETROU D, GIBSON G, GANGER G. Scheduling speculative tasks in a compute farm [C]// Proceedings of 2005 ACM/IEEE Conference on Supercomputing. Washington, DC: IEEE Computer

Society, 2005: 37-48.

- [7] KONDO D, TAUFER M, BROOK C, *et al.* Characterizing and evaluating desktop grids: An empirical study [C]// IPDPS04: Proceedings of the 18th International Parallel and Distributed Processing Symposium. Washington, DC: IEEE Computer Society, 2004: 26-35.
- [8] KONDO D, CHIEN A, CASANOVA H. Resource management for rapid application turnaround on enterprise desktop grids [C]// Proceedings of 2004 ACM/IEEE Conference on Supercomputing. Washington, DC: IEEE Computer Society, 2004: 17-30.
- [9] 朱承, 刘忠, 张维明, 等. 自愿计算中重叠网络拓扑对面向结点能力的非集中式调度方法的影响分析[J]. 计算机学报, 2005, 28(10): 1638-1644.
- [10] Globus project 2010[EB/OL]. [2010-09-01]. <http://www.globus.org/toolkit/>.
- [11] ANDRZEJAK A, DOMINGUES P, SILVA L. Classifier-based capacity prediction for desktop grids [C]// Integrated Research in Grid Computing - CoreGRID Workshop. Pisa, Italy: [s. n.], 2005: 61-70.
- [12] ANDRADE N, CIRNE W, BRASILEIRO F, *et al.* Ourgrid: An approach to easily assemble grids with equitable resource sharing [C]// Proceedings of the 9th Workshop on Job Scheduling Strategies for Parallel Processing, LNCS 2862. Berlin: Springer-Verlag, 2003: 61-86.
- [13] LUTHER A, BUYYA R, RANJAN R, *et al.* Alchemi: A .NET-based enterprise grid computing system [C]// ICOMP'05: Proceedings of the 6th International Conference on Internet Computing. Las Vegas, USA: CSREA Press, 2005: 269-278.
- [14] ANGLANO C, CANONICO M, GUAZZONE M, *et al.* Peer-to-peer desktop grids in the real world: The sharegrid project [C]// CCGRID 2008: Proceedings of the 8th IEEE International Symposium on Cluster Computing and the Grid. Washington, DC: IEEE Computer Society, 2008: 621-626.
- [15] HARRISON A, TAYLOR I. Dynamic Web service deployment using WSPeer [C]// Proceedings of the 13th Annual Mardi Gras Conference on Frontiers of Grid Applications and Technologies. Baton Rouge, LA: [s. n.], 2005: 11-16.

(上接第833页)

图3 公共管理模块界面

参考文献:

- [1] 姜寒. 基于嵌入式 Linux POS 系统架构研究[J]. 科技信息, 2009

(5): 451-456.

- [2] 徐敏. 嵌入式 POS 的新机遇[J]. 信息与电脑, 2009(3): 75-76.
- [3] 陈景, 陈平华, 李文亮. Android 内核分析[J]. 现代计算机, 2009(11): 112-114.
- [4] 公磊, 周聪. Android 的移动终端应用程序开发与研究[J]. 计算机与现代化, 2008(8): 85-89.
- [5] 姚昱曼. Android 的架构与应用开发研究[J]. 计算机系统应用, 2008(11): 110-113.
- [6] 仲中方, 张华, 闫明松. MVC 模式研究综述[J]. 计算机应用演进, 2004(10): 254-257.
- [7] 肖云鹏, 李茜. 一种基于 J2ME 的轻量级 MVC 框架设计[J]. 重庆邮电大学学报, 2008(4): 491-494.
- [8] 余志龙, 陈昱勋, 郑名杰. Android SDK 开发范例大全[M]. 北京: 人民邮电出版社, 2009.
- [9] GAMMA E, HELM R, JOHNSON R, *et al.* 设计模式: 可复用面向对象软件的基础[M]. 北京: 机械工业出版社, 2006.
- [10] 张炜达. J2EE 模式于 Android 平台的应用及嵌入式零售终端 CRM 子系统设计实现[D]. 厦门: 厦门大学, 2010.