

# 多代理技术在移动计费监控系统中应用

王友慈 谭立球 李建华 胡湘陵

(长沙中南大学铁道校区信息技术研究中心 410075)

**摘要:** 移动计费系统是移动业务的核心系统, 监控报警系统实现对移动计费系统的监控。传统的监控系统存在着过于集中、维护困难、灵活度不够等不足。本文采用AGENT技术, 实现了一个基于Multi-agent的移动报警监控系统, 与传统的监控系统相批, 具有较好的先进性。

**关键词:** 移动计费 报警监控 多代理 JADE开发环境

## Application for the Mobile Communication Billing Monitor System Based on Multi-agent

### 1 引言

移动计费监控系统(以下简称 MCBMS)主要是用于监控全省移动通信计费系统的运行情况, 以图文形式为用户提供计费系统中各个环节的处理状态信息、数据库信息以及计费网络信息; 当系统出现异常时, 能够寻找出错的原因, 直至找到错误发生的根源; 根据错误类型系统自动予以声光报警, 并提供一定的维护和纠错手段, 以保证计费系统的正常运行。传统移动监控系统采用客户/服务器方式, 在各个环节驻留客户端程序进行监控, 这种方式存在着过于集中、维护困难、灵活度不够等不足。

多Agent系统是由一组独立的但又协同工作的Agent构成的。本文采用Agent技术, 实现了一个基于Multi-agent的移动监控系统。

### 2 Multi-agent 移动报警监控结构描述

监控系统提供主要包括数据库监控、计费网络监控、交换机计费文本采集监控、计费文本传输监控、计费文本分拣监控、批价监控、详单处理监控、流向监控、分摊结算监控、计费文本错误更正监控、文本合并备份监控、文本上下传监控等。对于在计费系统批价之前的监控环节, 一旦检测到系统中的错误后, 可以根据系统运行的

规则和系统中的知识来进行智能查错, 使得错误根源可以迅速确定, 为管理人员提供解决问题的依据。

系统的框架结构如图1所示, 服务器端包括两种应用Agent, 一种是服务器主容器内的Agent: 任务管理Agent、错误管理Agent、数据库监控Agent和网络监控Agent, 它们在其生命周期中一直滞留在主容器内实时运行, 不必移动, 另一种是根据监控需要, 从主容器中迁移到指定位置运行的Agent, 如采集日志监控agent、分拣日志监控agent和传输文件监控agent。

该体系结构包括位于省计费中心的监控中心、计费系统环境监控点和分布于计费各环节的监控点组成:

(1) 监控中心包括完成系统功能的一些Agent, 如问题处理Agent、任务调度Agent和信息处理Agent等。它的功能包括系统整体故障的分析、处理, 其它Agent的产生、调度, 监控中心消息处理等。由于我们采用JADE作为Agent开发平台, 因此监控中心实质上还包含JADE平台的一些系统Agent;

(2) 计费系统环境监控点包括计费系统运行环境监控的Agent, 如数据库监控Agent以及网络监控Agent等。它的功能包括对计费系统的运

行环境的监控, 也就是说对系统的数据库系统和网络系统的状况进行实时检测, 对其产生的故障进行处理或者汇报;

(3) 而计费各环节的监控点则包括计费各环节故障检测、故障处理Agent, 如各环节的监控Agent、出错点分析Agent等。它的主要功能包括对于非运行环境的故障查找其出错点, 对出错点进行实时监控, 以及对各环节出现的故障进行实时处理或者汇报。这里需要说明的是, 由于计费系统中各地局的计费流程完全一样, 且处理过程从逻辑上为相互独立的, 所以事实上我们按地局将监控点划分开来, 每个地局都包含各计费环节的监控点。

监控中心的Agent都存活于系统唯一的主容器(Main-Container)中, 计费系统环节监控Agent、计费各环节的监控Agent和出错点分析Agent分别存活于分布式的容器(Container)中, 其中出错点分析Agent可以在多个计费环节监控点容器中自主移动。

### 3 系统实现过程

系统在启动时, 系统的主容器和存活于主容器中的三个平台管理Agent: AMS, DF, ACC随着启动, 同时开始启动实时监控的Agent, 主要包括在主容器中启动实时运行的TaskManage

Agent (任务管理 Agent)、ProblemSolverAgent(错误管理 Agent)、DBMAgent (数据库监控 Agent) 和 NetMAgent (网络监控 Agent), 同时移动 PickMAgent (分拣日志监控 Agent) 到分拣机上运行。启动过程结束后, MCBMS 的处理过程分为如下两种情形:

(1) 系统响应用户处理过程: 用户在监控系统客户端选择监控的地点、对象, 用户的请求被发送到服务端, 服务端的 TaskManageAgent (任务管理 Agent) 接收请求, 分析解释请求, Task ManageAgent 的智能控制模块根据知识库决定问题的解决方案, 决定如何将任务分解成子任务, 由哪些 Agent 分别执行子任务, TaskManage Agent 在制定执行计划后, 询问 DF (目录管理 Agent) 将要使用到的 Agent 的情况, 如果指定地点还没有生成需要的 Agent, 则由服务器容器生成 Agent, 并将 Agent 移动到指定的主机上运行, Agent 在主机上运行后, 将需要实时显示但无需

记录的信息, 直接发送 UDP 给客户端, 客户端将接收到的数据在界面上显示, 对于需要留下历史记录的信息, 在发送 UDP 给客户端的同时, 发送消息给服务端, 服务端将信息分类记录到 Oracle 数据库。

(2) 系统异常问题处理过程: 也就是, 实时监控的 DBMAgent (数据库监控 Agent)、NetMAgent (网络监控 Agent) 和 PickMAgent (分拣日志监控 Agent) 监控到计费系统发生错误异常时, 首先检查自身处理环节是否有问题(比如, PickMAgent 监控到计费系统发生分拣延时), 再把监控到的系统异常信息发送给 ProblemSolver Agent(错误管理 Agent), 由其来分析错误异常的产生根本原因, 如果不能最终确定, ProblemSolverAgent 将按照 NetMAgent、Pick MAgent、FileTransMAgent、CollMAgent 的顺序依次指派这些检查自身处理环节是否有问题, 等待它们的检查结果, 直到最终确定错误异常的根

源为止, 把此结果在客户端显示出来, 当然对于 FileTransMAgent 和 CollMAgent, 先要询问 DF (目录管理 Agent) 指定地点是否存在这些 Agent, 如果不存在, 则由服务器容器生成 Agent, 并将 Agent 移动到指定的主机上运行。

#### 4 结束语

本文采用 Multi-agent 技术, 实现了一个移动报警监控系统, 与传统的监控系统相比, 有如下一些优点:

(1) 它不需要各监控节点长期与监控中心保持联系, 可以很好地适应网络状况不稳定的状况, 而且减少了许多网络开销;

(2) 没有日常维护客户端程序的要求, 这样不仅极大程度上减少了系统管理开销, 而且很容易达到监控系统本身很高的可靠性和稳定性要求;

(3) 它可以充分利用结构化工作流的特点, 只在工作流的最后环节进行固定的监控, 在发生错误或者可能将要发送错误的时候在生成相应的移动代理到前面环节检查错误根源, 这样充分利用了结构化工作流的特点, 有效地减少了系统的开销和复杂度;

(4) 移动代理本身具有一定的智能性, 可以根据在移动过程收集的信息判断出错原因, 也可以自主进行一些相应的错误处理, 这样有效地将监控中心的负载分布到系统的各结点中去, 避免了过于集中的情形的出现;

(5) 由于移动代理都在监控中心产生, 因此系统的维护和升级只需更新监控中心的代码, 而无需一一到系统各结点进行, 这样极大地减少了维护和升级的工作量。 ■

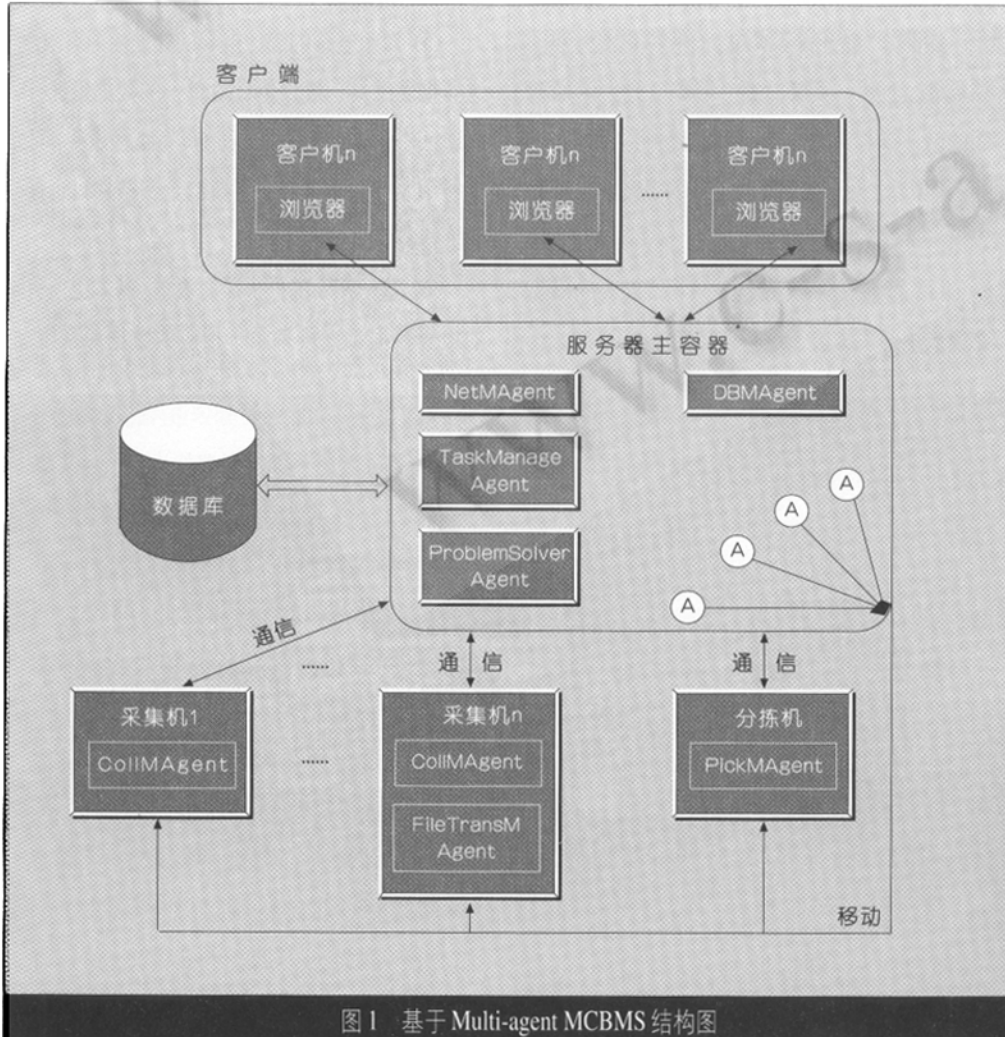


图1 基于 Multi-agent MCBMS 结构图

#### 参考文献

- 1 M. Wooldridge, "Agent-based software engineering", IEEE Proceedings on Software Engineering, 144(1), pp26-37, February 1997.
- 2 周永林、潘云鹤, "面向 Agent 的分析与设计", 计算机研究与发展, 1999年, 4月。