

# 一种应用于物联网的实时数据存储技术

张琴利 马耀华 肖坤 谢敬丰 潘俊  
(中核兰铀公司,甘肃兰州 730065)

**【摘要】**该文以飞邻研发中心飞邻平台为例,着重论述了一种实时数据存储服务解决方案。

**【关键词】**物联网 哈希表 TimesTen 时间戳

**【Abstract】**This paper take the Feeling Research Center SMC platform as example, mainly discuss a technical solution of real-time data storage.

**【Key words】**Internet of Things Hash Table TimesTen Timestamp

## 1 引言

当物联网与互联网和移动通讯网相连,就可随时随地全方位“感知”对方,这就是物联网带来的变化。飞邻产品SMC系统从底层终端设备采集数据到中间服务数据解析、处理到客户端UI的动态监控,是一个完整的监控系统。数据缓存模块提供以a.h(携带时间戳的ReadKey)为关键字的特定时间点/时间段的历史数据服务,UI可以利用数据缓存服务实现历史数据展示,也可以进行二次包装进一步提供数据回放服务等功能。

## 2 基本概念

### 2.1 物联网

物联网指的是将各种信息传感设备与互联网结合而形成的一个巨大网络,其中非常重要的技术是RFID电子标签技术。这个网络中,系统可以自动的、实时的对物体进行识别、定位、追踪、监控并触发相应事件。

### 2.2 时间戳

是一种时间表示方式。

### 2.3 哈希表

有序HASH内存表是目前速度最快功能最强大的数据结构,能保持超大规模数据。

### 2.4 TimesTen

TimesTen应用于需要快速响应时间的数据库。

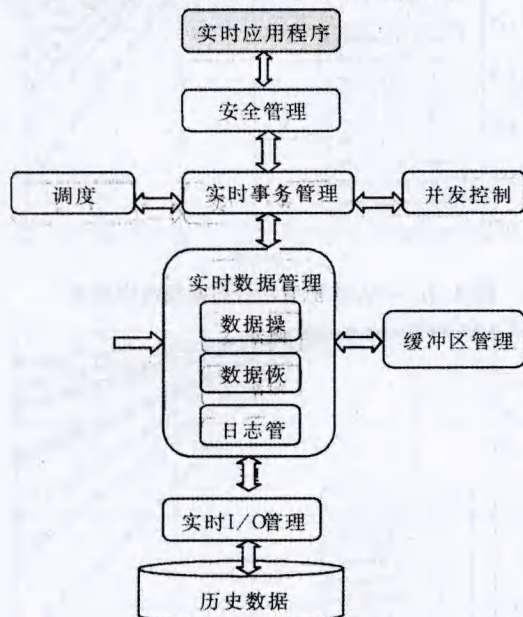


图1 实时数据库的结构

## 3 一般实时数据存储模型

在我国,控制层和管理层的计算机应用逐步形成了规模,但是这两个层次之间的数据交换仍是人工的,管理者不能及时掌握工厂的整体信息。可见,我们需要一种系统使生产过程控制和管理相结合。这种系统具有生产数据自动采集、数据过滤储存和智能分析等功能,使用户可以收集、储存、查看、分析、控制、管理工厂信息、使生产更有效率。具体连接过程见图1。

## 4 飞邻平台数据存储方案

### 4.1 飞邻中的位置

存储中实时数据的保存就是模型中缓存管理的位置。在飞邻平台里,缓存管理主要针对实时数据的历史数据的存储。数据存储服务在飞邻平台中位置见图2。

飞邻平台中有一个非常重要的高级服务DataCenter;它提供给各基础服务通讯接口,负责协调各模块之间的数据通信和管理。

### 4.2 使用模式

数据缓存的用户分为三类:

(1)飞邻内部用户如诊断服务,工作流等;(2)行业应用软件开发商;(3)行业系统集成开发商。

针对数据缓存服务的三类用户,提供独立通道服务使用模式和命令交互使用方式。独立通道服务使用模式见图3。

数据缓存服务与UI以命令方式交互见图4。

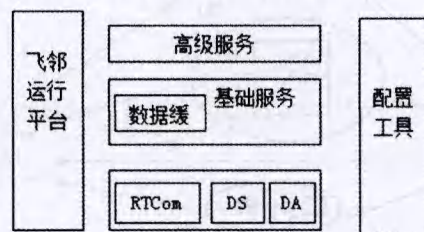


图2 数据存储服务在飞邻系统中位置

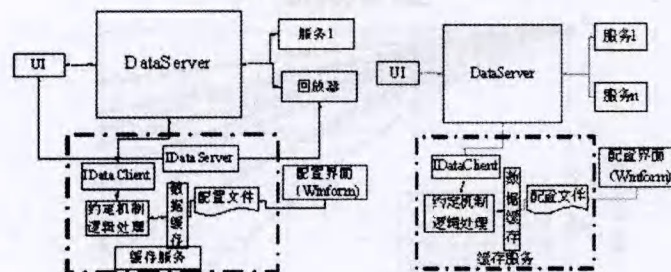


图3 独立通道模式

图4 命令方式

.....下转第40页

作者简介:张琴利(1977-),女,汉族,陕西人,工程师,中核兰铀公司。

马耀华(1978-),女,汉族,甘肃人,工程师,中核兰铀公司。

谢敬丰(1971-),男,汉族,甘肃人,高级工程师,中核兰铀公司。

与完全获得频谱情况相同,  $P_k = \prod_{l=1}^k A_l \bar{P}_d \prod_{r=1, r \neq l}^{M-1} (1 - A_r \bar{P}_d)$ , 所以

$$P[C=w|K=k] = \sum_k P_k \quad (16)$$

更进一步, 若所有从源节点出发到达中继节点的链路均相同, 那么  $\forall l \in [1, M]$ ,  $A_l$  都相同, 则上式简化为

$$P[C=w|K=k] = \binom{k}{w} (A_l \bar{P}_d)^w (1 - A_l \bar{P}_d)^{k-w} \quad (17)$$

此外, 与完全获得频谱情况一致, 该情况下

$$P[I < R | C=w, K=k] = \sum_{l=0}^w \sum_{j=1}^{N_r m_l} \frac{(-1)^j \beta_{j,l} \left( \frac{\lambda_l}{m_l} \right)^j}{\Gamma(j)} \Gamma_{inc} \left( j, \gamma \frac{m_l}{\lambda_l} \right) \quad (18)$$

相同地, 若到达宿节点的信号经历了相同的 Nakagami-m 信道,  $\forall l \in [0, w]$ ,  $\lambda = \lambda_l$ ,  $m = m_l$ , 上式简化为

$$P[I < R | C=w] = \frac{\Gamma_{inc}(m N_r (w+1), \gamma m / \lambda)}{\Gamma(m N_r (w+1))} \quad (19)$$

将式(14)、(16)和(18)代入式(13), 就得到不完全频谱获得情况下的  $P_{out}$ 。

#### 4.3 基于内部合作获得频谱

假定参与合作的相邻节点均采用与潜在认知中继节点相同的判决规则, 且参与合作节点数目记为  $N$ , 则该情况下的平均检测概率和虚警概率为<sup>[1]</sup>

$$\bar{C}_d = 1 - (1 - \bar{P}_d)^{N+1}, C_f = 1 - (1 - P_f)^{N+1} \quad (20)$$

式中  $\bar{P}_d$  和  $P_f$  分别由式(3)和(5)给定。

将不完全获得频谱下得到中断概率式中的  $\bar{P}_d$  替换为  $\bar{C}_d$ , 得到基于内部合作获得频谱的中断概率。完全获得频谱下得到的中断概率式与该式相比得到<sup>[1]</sup>

$$\Omega(M, SNR, \bar{C}_d, N, N_c, N_r) \approx 1 \quad (21)$$

给定  $M$ 、 $SNR$ 、 $N_c$ 、 $N_r$  和  $\bar{C}_d$ , 如果  $N$  满足上面的等式系统就能达到全分集。可以看到中断性能随着合作节点数目的增加而接近完全获得频谱的情况。

### 5 性能仿真

为仿真分析简便, 假定信道增益的特征值  $\lambda = 1$ , 固定目标速率  $R = 1$ , 时间-带宽积  $\Lambda = 6$ , 虚警概率固定为  $\bar{P}_f = C_f = 10^{-3}$ 。

图2给出的是认知中继完全获得频谱的情况下中断率性能的对比结果。图中分别设  $N_c = N_r = 1, 2, 3$ ,  $m = 1, 2, 3$  共有9个组合的中断率性能仿真。中继簇数目  $M = 1$ 。显然, 给定  $m$ , 接收天线个数越多, 中断率性能越好。固定天线个数, 中断率性能随着  $m$  的增大而表现出

更好的中断率性能。

图3是将图2中继簇数目改变为  $M = 4$ , 可以看到接收天线数目和 Nakagami 参数  $m$  对系统中断率性能的影响与图2类似, 不同的是, 在低信噪比情况下, 越大  $M$ , 中断性能越差。图4为认知中继不完全获得频谱的情况下中断率性能的对比结果。中继簇数目  $M = 4$ , 同时给出了分别设  $N_c = N_r = 1, 2, 3$ ,  $m = 1, 2, 3$  共为9个组合的中断率性能仿真。可以观察到, 不完全获得频谱情况下在接收天线数目和 Nakagami 参数对系统中断率性能的影响与图2也类似。但很明显, 不完全获得频谱的情况比完全获得频谱情况下的中断率性能要差很多。

接下来的图5展示了合作节点数目对中断性能的影响。看到随着合作接点的增多, 中断率性能逐渐逼近完全获得频谱情况。假定中继簇数目  $M = 4$ ,  $m = 2$ ,  $N_r = N_c = 3$ 。显然, 随着  $N$  的增加, 中断率性能在不断提高,  $N = 6$  系统到达全分集。

图6给出了  $\Omega(M, SNR, C_d, N, N_c, N_r)$ 。展示了达到全分集需要的合作节点数目。给定中继簇数目  $M = 4$ ,  $SNR = 30\text{dB}$ 。分别画出了  $m = 1, 2$ ,  $N_r = N_c = 1, 3$  共4个组合的  $\Omega(M, SNR, C_d, N, N_c, N_r)$ 。从图中观察到给定天线个数,  $m$  越大所需合作节点数目越多。例如  $N_r = N_c = 3$  情况下,  $m = 1$  时,  $N = 6$  系统达到全分集,  $N > 6$  对系统中断性能的改善不再起作用。这与图3得到的结果一致。而  $m = 2$  时, 欲达到全分集所需合作节点数为  $N = 11$ 。再者, 给定  $m$ , 天线个数越大所需节点个数越小。例如,  $m = 2$  时,  $N_r = N_c = 1$ , 需要合作节点大约为  $N = 45$ ,  $N_r = N_c = 3$ , 需要合作节点大约为  $N = 11$ 。特别地,  $N_r = 1$  对应单天线情况,  $m = 1$  是瑞利衰落情形。

### 6 结语

本文研究了基于 Nakagami-m 衰落的多天线认知中继协作中断性能。根据认知中继节点的频谱获得能力分三种场景分析其性能: 完全获得频谱、不完全获得频谱和基于内部合作的频谱获得。可以看到, 在 Nakagami-m 衰落信道条件下, 与单天线情况相比多天线情况能达到更好的中断率性能, 而且 Nakagami 参数  $m$  越大, 性能表现也越好。特别是引入认知节点内部协作检测进一步改善了中断率性能, 通过选择合适的合作数目, 系统可达到全分集。

#### 参考文献:

- [1] Kyoungwan Lee and Aylin Yener. Outage Performance of Cognitive Wireless Relay Networks[C]. Proc. of IEEE GLOBECOM 2006. Nov.27 2006-Dec.2006, pp.1-5.
- [2] Xiaomao Mao, Peiliang Qiu. Cognitive Relay[C]. Future Generation Communication and Networking (FGCN'07), Dec.2007, pp. 264-269.

.....上接第37页

#### 4.3 服务模块功能

(1) 用户注册模块: 使用 DataServer 统一接口, 由用户动态注册需要进行缓存的数据标志, 用户注册时要输入唯一标识键值便于用户读入; 创建索引表, 并根据输入确定初始化内存存储区数据结构。(2) 用户数据写入模块: 提供写入数据的方法, 对写入的键进行筛选, 把注册键的值写入到对应的内存缓存区中。(3) 数据存储方式可配置: 数据缓存最重要的技术点是缓存写入的存储模型, 如可采用 HASH TABLE, 实时数据库等方法。为实现不同场景对性能、功能的不同需求, 缓存写入模型应做到灵活可配置。初步规划实现基于 Hash map 和实时数据库两种存储模型。(4) 用户数据读取模块: 用户提交要从缓存读取的数据键和条件, 模块根据键查找数据存储区, 然后根据条件筛选后把结果集返回给用户。(5) 缓存数据表维护、监控模块(预留): 提供用于对大量存储区数据的维护、监控数据的状态等功能。

#### 4.4 二次开发接口

从实现考虑, 缓存存储接口定义为抽象类, 用户继承后可以灵

活二次开发, 生成自己的数据缓存服务dll文件。

#### 4.5 方案测试

编程人员基于二次开发接口完成以哈希表作为存储载体的驱动和基于 TimesTen 实时数据库为存储载体的驱动, 加载服务进行了测试, 性能基本符合设计要求。

### 5 结语

本数据存储技术初步测试结果表明这种存储方案能够满足实时数据存储要求。且服务配置灵活、提供丰富的二次开发接口; 因此, 针对物联网复杂、实时性较高的应用领域, 也是一个不错的数据存储方案。

#### 参考文献:

- [1] 郎锐, 罗发根. Visual C++ 网络通信程序开发指南. 北京: 机械工业出版社, 2004年.
- [2] W. Richard Stevens. The Socket Networking API. Addison-Wesley, 2004.