

一种面向物联网应用的底层网络通信模块设计 *

杨丰玉, 陈 英, 吴振华

(南昌航空大学 软件学院 软件工程系, 南昌 330063)

摘 要: 通过引入接收线程池、转发对象池、文件缓冲区等技术, 构建了一个面向大规模物联网应用的底层网络通信模型。实验结果表明, 该模型的效率与稳定基本能满足大规模的底层通信, 适用于物联网应用。

关键词: 物联网应用; 底层通信; 线程池; 对象池; 缓冲区

中图分类号: TP391.4

文献标志码: A

文章编号: 1001-3695(2012)03-1057-03

doi:10.3969/j.issn.1001-3695.2012.03.070

Design of underlying communication model for application in Internet of Things

YANG Feng-yu, CHEN Ying, WU Zhen-hua

(Dept. of Software Engineering, School of Software, Nanchang Hangkong University, Nanchang 330063, China)

Abstract: This paper constructed a communication model adapted to large-scale applications in Internet of Things with following techniques, receiving thread pool, relaying object pool and file writing buffer. Plenty of experiments prove that the model is much efficient and stable to work for large-scale applications in Internet of Things.

Key words: applications in Internet of Things; underlying communication; thread pool; object pool; buffer

0 引言

物联网(Internet of Thing)^[1]是把具有标志、感知和智能处理能力的物体借助通信技术互连而成的网络, 这些无须人工干预即可实现协同和互动, 目的在于为人们提供智能服务。随着物联网基础网络的构建及软件的开发, 物联网正在进入快速发展的时期, 已广泛应用于军事航空、城市管理、定位导航、智能交通、数字家庭等领域^[2,3]。

物联网将各种设备资源连接到一个网络中, 并从标志和感知设备(也称为传感器)中获取信息, 从传感器获取的信息由网关转发到应用层处理。因此, 消息在各种设备间的转发速度、接收正确率将直接决定物联网的工作效率和作用, 设计一个快速、高效、可扩展性强的底层网络通信模块就显得尤为重要。

目前, 物联网领域的研究工作主要集中于感知层、节点状态的监控、节点信息的整合、通信协议格式等的研究^[4], 对传输层的通信效率方面研究较少。本文拟研究物联网大规模应用情况下传输层的通信效率问题。

1 网络通信模型

物联网通常可以概括为三层: a) 感知层^[5], 通常是基于物理、化学、生物等技术的传感器, 采集所需的各种数据(如温度、湿度等); b) 传输层^[5], 其主要传输方式分为有线和无线两种; c) 应用层^[5], 对采集上来的数据进行智能分类、查询处理及决策支持等。

通常的物联网应用中, 传输层的通信模型负责将感知层传感器采集的数据转发到应用层进行应用分析和处理。通信模型的输入为分布在各地的传感器按照一定的周期采集发送的

数据, 随着物联网应用领域和范围的扩大, 采集发送的数据量也在呈现指数级的增长, 特别是实时性要求较高的领域, 因此对通信模型的要求也越来越高。一个典型的物联网通信结构如图 1 所示。

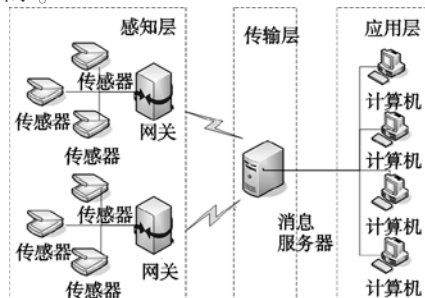


图1 典型的物联网应用结构图

由图 1 可以看到, 底层传感器节点采集数据后, 会将数据传送到与其相连的网关, 各个网关对数据封装后发送到系统的消息服务器中, 消息服务器根据配置好的路由表转发所接收的消息到应用层的计算机中, 应用层按实际需求解析处理数据。

感知层与传输层以及应用层与传输层的通信均为典型的多对一结构。当系统的规模不断增长后, 传输层就成为系统的通信中心, 传输层消息服务器的通信模型好坏将直接影响系统的性能。因此一个良好的通信模型需要解决以下几个关键问题:

a) 数据接收。消息服务器接收网关发送的数据时, 首先启动监听, 在指定的端口监听数据, 当监听到有数据到达时, 读取并解析其中的目的数据, 根据初始配置的路由表, 将数据转发到目的地。当监听的端口有大量数据到达时, 如果处理连接的并发能力不够时, 就会造成丢失网络数据包, 导致业务层读取原始数据不准确, 影响正常的业务功能。

b) 数据转发。消息服务器的职责除了接收网关发送的数

收稿日期: 2011-09-08; 修回日期: 2011-10-30 基金项目: 江西省自然科学基金资助项目(2010BGA02100)

作者简介: 杨丰玉(1980-), 男, 江西九江人, 讲师, 硕士, 主要研究方向为软件工程、数据挖掘(frueyang@163.com); 陈英(1981-), 男, 江西抚州人, 讲师, 硕士, 主要研究方向为无线传感器网络; 吴振华(1978-), 男, 江西波阳人, 副教授, 博士, 主要研究方向为无线传感器网络。

据外,还需要将接收的数据转发至目的地。转发数据首先必须创建本地端口,与目的地建立网络连接,再发送数据。创建端口不仅耗费内存资源,还需要访问 I/O,访问 I/O 的速度远远低于访问内存资源,如果要发送大量数据,创建端口的速度将成为转发数据的瓶颈。

c) 日志读写。消息服务器在接收并转发数据时,需要记录转发消息的日志,以观测系统的性能。常用的方法是采用文件来进行日志记录,但文件的打开和关闭也涉及 I/O 操作,较为费时。如果每次转发消息都打开并关闭一次文件,将极大地耗费时间。导致日志操作的时间比转发消息的时间还要长,严重影响了系统的性能。

2 网络通信模块设计

由上文分析可知,一个性能优良的网络通信模块必须要解决以上几个关键问题才能应对大规模物联网节点通信的功能需求及性能需求,满足物联网应用的需要。针对以上问题,本文拟采用接收线程池、转发连接对象池和文件缓冲区等技术构建网络通信模块。

2.1 接收线程池

线程池^[6,7]的基本原理是先启动若干数量的数据接收线程,并让这些线程都处于睡眠状态,当某个传感器节点采集完数据后请求上传连接时,就会唤醒线程池中的某一个睡眠线程,让它来处理该连接请求;当处理完这个请求任务后,该条线程又处于睡眠状态。如果为每个连接请求分别创建一条新线程,那耗费的 CPU 时间和内存将是惊人的,而采用线程池技术将会节约大量的系统资源,使得更多的 CPU 时间和内存用来处理接收数据的解析任务,而不是频繁地创建与销毁线程。

线程池的基本参数有:

a) 基本线程数。线程池中的最小线程数,线程池在初始化时就创建的线程数量,并且在运行过程中保证池中存活线程的最低数量。

b) 最大线程数。线程池中的最大线程数,当线程池的基本线程数不能满足需求时,可不断创新新的线程,直至达到最大线程数。

c) 线程最长存活时间。线程池中每个线程的存活时间,即到达该存活时间的线程在完成自己的当前任务后,将会被线程池回收并释放,如果回收后线程池中不足基本线程数,将补充新的线程,直至达到基本线程数。

d) 任务队列长度。应对感知层发送的大量数据时,可采用多线程机制,将接收线程池化,每当监听端口有新的数据到来时,即从线程池中取出一个接收线程,接收数据并交给转发连接对象进行转发。

2.2 转发连接对象池

对象池的基本工作原理是将用过的对象保存起来,等下一次需要这种对象的时候,再拿出来重复使用,从而在一定程度上减少频繁创建对象所造成的开销。用于充当保存对象的“容器”的对象被称为“对象池”。

若初始化、实例化的代价高,且需要经常实例化,但每次实例化数量较少的情况下,使用对象池可以获得显著的效能提升。从池中取得对象的时间是可预测的,但新建一个实例所需的时间是不确定。

转发连接对象的主要工作过程是,在接收线程接收到数据后,分析数据要发送的目的地,根据配置的路由表获取要发送的目的地址,然后调用转发连接对象发送数据,转发连接对象首先创建到目的地址的连接,然后发送数据,发送成功后释放该连接。在这一过程中,创建到目的地址的连接及释放连接最为耗时,因为建立和释放连接需要执行 I/O 操作。

将转发连接对象池化,首先创建转发连接对象的对象池,建立一定数量的连接对象,当转发数据时,从池中取出一个转发连接对象,发送数据后,再将连接对象还回对象池,从而避免了大量创建和释放连接所消耗的资源,提高了系统性能。

2.3 文件缓冲区

缓冲区是为了提高操作性能、特别是需要频繁地执行某种耗费时间的 I/O 操作而设立的临时区域,将数据先写到临时区域而不直接写到目标设备中,直到缓冲区充满再整体写入到目标设备,从而提高系统的 I/O 操作性能。

文件缓冲区是在将内容写入文件时,不直接写入到文件,而是先将内容写入到设置的缓冲区内,直接缓冲区充满再将缓冲内的内容整体写入文件,避免多次访问文件导致大量的 I/O 操作,从而提高系统性能。

设置文件缓冲区是因为转发连接对象在成功转发数据后,需要记录转发日志信息到日志文件,为避免频繁的文件操作而进行的改进。

2.4 通信模型

采用了以上关键技术的通信模型结构如图 2 所示。

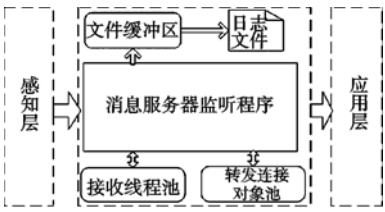


图2 通信模型结构图

通信模型的处理算法如图 3 所示。

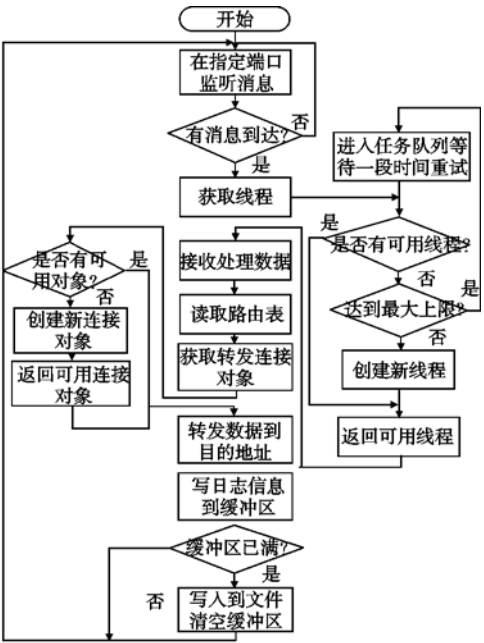


图3 通信模型处理算法流程图

3 实验分析

在设计了通信模型后,采用 Java 语言对模型进行了实现,并实现了模拟网关的数据发送程序和应用层终端数据接收程序,对模型进行了多次实验,验证模型的性能和工作效率。

3.1 实验环境

1)开发测试环境
实验中所使用的软件与硬件环境如表 1 所示。

表 1 实验环境

子系统	CPU	内存	操作系统	JRE	硬盘空间
模拟网关	主频 2.0 GHz	512 MB	Win XP	1.6	1 GB 以上
消息服务器	主频 2.66 GHz 双核服务器	2 GB 以上	Win 2003 Server	1.6	2 GB 以上
应用终端	主频 2.0 GHz	512 MB	Win XP	1.6	1 GB 以上

2)网络环境
为了获得真实的测试结果,结合通信模型的实际功能,设置了一个模拟网关、一个消息服务器、三个模拟应用终端(图 4)。模拟终端可以模拟多个线程同时发送数据,数据的目的地址随机产生,由消息服务器转发到三个模拟应用终端。

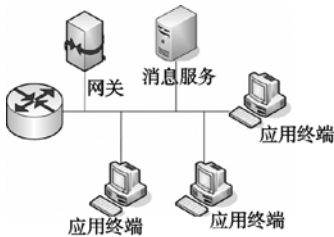


图4 应实验环境拓扑图

3.2 实验过程

软件主要分为三个部分:
a)模拟网关。主要模拟多个线程同时发送多个数据包到消息服务器中。模拟网关的伪代码如下:

```
/* 模拟网关主程序 */
获取初始化参数(线程数、间隔时间、持续时间、消息服务器地址);
while (运行时间 < 持续时间)
{
    创建本地 UDP 发送端口;
    组装要发送的数据;
    发送数据包到目标地址;
    记录日志;
    b)消息服务器。接收网关发送的消息,解析其中的目标地址,根据路由表转发到目标终端。消息服务器的伪代码如下:
    /* 消息服务器的主程序 */
    创建服务端监听端口;
    创建接收线程池;
    创建转发对象池;
    初始化文件缓冲区;
    while (有数据包到来) {
        从接收线程池中取出一个可用线程;
        解析数据包内容,获取目标;
        从路由表查到具体的转发地址;
        从转发对象池取出一个转发对象;
        转发数据;
    }
}
```

```
记录发送日志;
}
c)模拟终端应用程序。接收消息服务器转发的消息数据,并记录接收日志。模拟终端应用程序的伪代码如下:
/* 终端应用主程序 */
创建服务端监听端口
while (有数据包到来)
{
    接收数据;
    记录接收日志;
}
```

3.3 实验数据分析

实验中,模拟网关模拟 200 个线程,每个线程每隔 2 s 发送 150 个消息包,持续 1 min,共计 930 000 个数据包。

1)线程池使用前后的性能对比 使用线程池程和未使用线程池的性能对比分析如表 2 所示。

表 2 使用线程池前后性能对比分析

项目名称	未使用线程池	使用线程池
转发数据包数(成功率/%)	259 027(27.85)	419 756(45.14)
终端收到数据包数(成功率/%)	258 826(27.83)	419 756(45.14)
CPU 峰值/%	10	25
内存峰值/KB	36 366	67 480
转发能力/包/s	884	2 657
接收时间/s	293	158

采用线程池,提高了系统的并发能力和转发成功率,接收时间上也有一定的缩短,但内存值占用较大。这是由于线程池需要由 CPU 来管理,每个线程有一定的存活时间,线程的创建需要耗费较大的内存,线程的调度需要由 CPU 控制,因此耗费了一定的 CPU 和内存值,带来了一定程度上的性能提升。

2)转发对象池使用前后性能对比 使用转发对象池及未使用转发对象池的性能对比分析如表 3 所示。

表 3 使用对象池前后性能对比分析

项目名称	未使用转发对象池	使用转发对象池
转发数据包数(成功率/%)	259 027(27.85)	930 000(100)
终端收到数据包数(成功率/%)	258 826(27.83)	776 140(83.46)
CPU 峰值/%	10	21
内存峰值/KB	36 366	36 548
转发能力/包/s	884	10 814
接收时间/s	293	86

采用对象池,大大提高了系统的转发成功率,转发能力也得到了明显的提升,接收时间得到了有效地改善。由于对象池主要改进的是对象的创建,即 UDP 连接对象不再是每次需要再创建,而是在对象池中预先创建。由此可以看出,UDP 连接对象耗费时间巨大,因此,采用对象池后性能得到了极大提升。

3)文件缓冲区使用前后性能对比 使用文件缓冲区与未使用文件缓冲区性能对比分析如表 4 所示。

表 4 使用文件缓冲区前后性能对比分析

项目名称	未使用文件缓冲区	使用文件缓冲区
转发数据包数(成功率/%)	259 027(27.85)	261 256(28.10)
终端收到数据包数(成功率/%)	258 826(27.83)	261 256(28.10)
CPU 峰值/%	10	6
内存峰值/KB	36 366	35 104
转发能力/包/s	884	929
接收时间/秒	293	281

文件缓冲区只是在一定程度上降低了 CPU 及 (下转第 1075 页)

4 仿真实验

以式(2)描述的系统作为实验对象,其中 $\omega = 0.99\text{ Hz}$,被测微弱信号角频率 $\omega_1 = 1.02\text{ Hz}$,幅值 $A = 0.01\text{ V}$,信噪比为 -40 dB ,则有

$$\begin{cases} x' = y \\ y' + 0.5y + (-x + x^3) = r \cos(0.99t) + 0.01 \cos(\omega_1 t) + n(t) \end{cases}$$

系统策动力的角频率 ω 初值为 0.03 Hz ,循环步长为 0.06 Hz , $n(t)$ 为均值为 0 、方差为 1 的随机噪声,系统的初值为 $x = 0, y = 1$,仿真步长 $h = 0.1$ 。按照循环混沌检测方法的步骤计算实验。

图 6(a) 为系统输入被测信号时混沌现象的相图, (b) 是满足本实验的间歇混沌现象的时域图。循环 16 次后,检测器检测到系统发生了间歇性混沌现象如图 6(b)。可以明显看到,图 6(b) 中总时间 $2\,000\text{ s}$ 内有 9.5 个间歇周期,则间歇性周期 $T = 2000/9.5 = 210.5263\text{ s}$ 。此时系统跳出循环,按循环混沌检测法的步骤,利用式(6)计算出被测微弱信号角频率 $\omega_1 = \omega + 2\pi/T = 0.99 + 0.0298 = 1.0198\text{ Hz}$,与实验所设的真实值误差只有 0.0196% ,误差极小。因此循环混沌检测系统在低信噪比下检测出的目标微弱信号精度高,检测效果也很好,其运算量较传统的混沌阵法大大降低。

5 结束语

通过分析混沌振子系统在未知信号检测方面的缺点,提出了循环混沌检测系统。该方法在未知频率检测信号时,将传统的数目巨大的混沌阵阵元的建立变为了易于工程现实的循环设计(循环次数减少为混沌阵阵元个数的一半),只需调整振

子的参数,就可以实现信号检测的目的,使混沌未知信号检测法简单、可实现。最后仿真实验表明了这种方法的有效性。

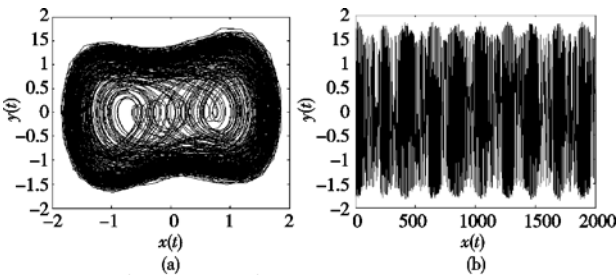


图6 低信噪比下微弱信号输入到循环混沌检测系统中后的相图与间歇混沌现象

参考文献:

[1] BIRX D L. Chaotic oscillator and CMFFNS for signal detection in noise environments [C]//Proc of IEEE International Joint Conference on Neural Networks. 1992;881-888.

[2] 陈维,孟晨,崔少辉,等. 基于杜芬振子的微弱信号频率测量[J]. 计算机测量与控制,2010,18(9):1982-1985.

[3] 吴永峰,张世平,孙金玮,等. 环形耦合 Duffing 振子间的同步突变[J]. 物理学报,2011,60(2):186-192.

[4] WANG Guan-yu, CHEN Da-jun, LIN Jian-ya. The application of chaotic oscillators to weak signal detection[J]. IEEE Trans on Industrial Electronics, 1999, 2(46):440-444.

[5] 汪学兵,张林华,李传东. 混沌同步及其在保密通信中的应用[J]. 计算机应用研究,2007,24(5):127-132.

[6] LOU T L. Frequency estimation for weak signals based on chaos theory [C]//Proc of International Seminar on Future BioMedical Information Engineering. Washington DC:IEEE Computer Society, 2008;361-364.

[7] 姜万录,吴胜强. 基于混沌振子的微弱信号检测方法及应用[J]. 机床与液压,2002,21(4):194-196.

(上接第 1059 页)内存使用的峰值,接收能力并未得到明显改善。由此可见,文件读写在此模型结构中所耗费的系统资源较小,主要是由于每次 I/O 读写的时间与建立网络连接相比较短,因而体现不出性能的提升。

4) 使用三种技术的组合前后性能对比 未使用任何技术与使用了三种技术组合的性能对比分析如表 5 所示。

表 5 使用三种技术组合前后性能对比分析

项目名称	未使用三种技术	使用了三种技术的组合
转发数据包数(成功率/%)	259 027(27.85)	930 000(100)
终端收到数据包数(成功率/%)	258 826(27.83)	928 898(99.88)
CPU 峰值/%	10	18
内存峰值/KB	36 366	36 626
转发能力/包/s	884	11 328
接收时间/s	293	82

由以上几组实验数据比较可知,系统最大的瓶颈在于转发时 UDP 连接对象的创建时间;其次是采用多个线程接收网关传输过来的数据;最后是文件的读写。三种技术的组合,能大幅度且有效地提升系统的工作效率。

4 结束语

本文针对物联网大规模应用中可能存在的通信效率问题,提出了一种可提高系统通信效率的底层通信模型。模型采用线程池、转发对象池及文件缓冲区技术,共同组合提升传输性

能。通过模拟实际的实验数据,分别对各种技术与原始传输方式的数据结果进行比较,得出结论:在网络应用中,网络连接对象所占用的时间是主要的时间耗费,线程池和文件缓冲区也在一定程度上缓解了内存和 CPU 的压力,三者共同组合可解决系统通信时存在的多种问题,大大提升系统性能。

在目前已取得的研究成果基础上,后续的研究工作将进一步研究分析对象的规模、对象的生存周期、初始对象数和最大对象数等因素对模型效果的影响程度,并给出在特定的应用规模条件下,各个参数如何配置才能使性能达到最优化。

参考文献:

[1] 孙利民,李建中,陈渝,等. 无线传感器网络 [M]. 北京:清华大学出版社,2005.

[2] 张峰,张晓鹏,吴高成. 基于物联网的机场集成行李处理系统及其应用研究[J]. 计算机应用研究,2010,27(10):3771-3774.

[3] 黄冬梅,方的苟,张明华,等. 物联网技术在救灾物资配送管理系统中的应用[J]. 计算机应用研究,2011,28(1):189-194.

[4] 顾晶晶,陈松灿,庄毅. 基于无线传感器网络拓扑结构的网络定位模型[J]. 计算机学报,2010,33(9):1548-1551.

[5] 周洪波. 物联网:技术、应用、标准和商业模式 [M]. 北京:电子工业出版社,2010.

[6] 周钦强. 一种利用并发提高数据处理吞吐率的模型[J]. 计算机系统应用,2010,19(11):190-192.

[7] LING Yi-bei, MULLEN T, LIN Xiao-la. Analysis of optimal thread pool size[J]. Operating Systems Review, 2000, 34(2):42-55.