

基于传感器的 GPU 集群功耗收集监控系统

李超凡, 陈庆奎

(上海理工大学 光电信息与计算机工程学院, 上海 200093)

摘要: 图像处理器(GPU)集群因其高性能的特性而被广泛应用,但随着 GPU 规模的增大,其高功耗问题会降低系统的可靠性。为此,提出一种 GPU 集群功耗收集系统,并设计基于 ZigBee 无线传感器网络的 GPU 集群功耗收集监控网络,同时构建收集通信协议和数据库存储系统,通过运行该系统可有效避免通信冲突。实验结果表明,该监控系统可以精确地测量集群中各个 GPU 的功耗,系统测量误差和丢包率分别低于 1% 和 0.005%。

关键词: 图像处理器集群; ZigBee 无线传感器网络; 通信协议; 数据库存储; 低误差低丢包率

中文引用格式: 李超凡, 陈庆奎. 基于传感器的 GPU 集群功耗收集监控系统[J]. 计算机工程, 2019, 45(3): 65-72.

英文引用格式: LI Chaofan, CHEN Qingkui. GPU cluster power consumption collection and monitoring system based on sensor[J]. Computer Engineering, 2019, 45(3): 65-72.

GPU Cluster Power Consumption Collection and Monitoring System Based on Sensor

LI Chaofan, CHEN Qingkui

(School of Optical Electrical and Computer Engineering,
University of Shanghai for Science and Technology, Shanghai 200093, China)

[Abstract] Graphics Processing Unit(GPU) clusters are widely used for their high performance characteristics, but with the enlargement of GPU scale, their high power consumption reduce system reliability. Therefore, a GPU cluster power consumption collection system is proposed, and a GPU cluster power collection and monitoring network based on ZigBee Wireless Sensor Network(WSN) is designed. At the same time, a collection communication protocol and a database storage system are constructed. By running the system, communication conflicts can be effectively avoided. Experimental results show that the monitoring network can accurately collect the power consumption of each GPU in the cluster, and the system measurement error and packet loss rate are less than 1% and 0.005% respectively.

[Key words] Graphics Processing Unit(GPU) cluster; ZigBee Wireless Sensor Network(WSN); communication protocol; database storage; low error and low packet loss rate

DOI: 10.19678/j.issn.1000-3428.0049676

0 概述

随着大数据、云计算、人工智能等新一代互联网技术的发展,大量的数据对计算机的计算速度、计算效率、计算可靠性提出更高的要求,随着数据的不断增加,其规模已经达到了 TB 甚至 PB 量级^[1],同时面临的图形或者数据的计算复杂度也越来越大,这对于自身计算能力的要求也越来越高。图像处理器(Graphics Processing Unit, GPU)主要的功能是对图像的计算,同时也有通用计算的特点^[2]。另外,在

成本方面, GPU 的计算成本较低,结合自身的可靠性高、计算能力强、性价比高等优势,因此被市场所广泛应用,从而为一些高计算复杂度、大数据计算提供了新的解决方案。但是随着 GPU 集群规模的增大, GPU 功耗是一个不可忽略的问题,并且随着功耗的增加会降低系统的可靠性。因此,测量 GPU 的功耗进而进行功耗优化具有重要的意义^[2]。ZigBee 是基于 IEEE802.15.4 标准的低功耗局域网通信协议,同时, ZigBee 技术是一种短距离、低功耗的无线通信技术,具有近距离、低复杂度、自组织、低功耗、低数据

基金项目: 国家自然科学基金(61572325, 60970012); 高等学校博士学科点专项科研博导基金(20113120110008); 上海重点科技攻关项目(14511107902, 16DZ1203603); 上海市工程中心建设项目(GCX14014); 上海智能家居大规模物联网共性技术工程中心项目(GCX14014); 上海市一流学科建设项目(XTKX2012); 沪江基金研究基地专项(C14001)。

作者简介: 李超凡(1990—), 男, 硕士研究生, 主研方向为 GPU 集群、并行计算、模式识别; 陈庆奎(通信作者), 教授、博士、博士生导师。

收稿日期: 2017-12-12 **修回日期:** 2018-02-04 **E-mail:** chenqingkui@usst.edu.cn

速率等特点^[3]。因此,ZigBee 可以应用于自动控制 and 远程控制领域,同时也可以嵌入到一些智能设备和物联网等诸多领域,具有较大的应用前景。

本文设计一种基于 ZigBee 技术的 GPU 集群功耗收集监控网络。将传感器节点部署在 GPU 上,且每个 GPU 部署一个终端采集节点,把采集到的 GPU 实时功耗数据通过 ZigBee 构建的无线传感器网络传输到协调器,经 USB 串口通信发送到监控终端进行实时监测分析。

1 相关工作

目前,测量 GPU 功耗的方法主要是软件测量方法和预测方法^[2],这些方法基本是以估算和预测为基础,不能精确地反映出 GPU 的实时功耗。GPU 硬件仿真方法也可以得到其功耗,但是这种方法对 GPU 硬件架构的认知能力要求高,并且 GPU 仿真方法通用性较低。

1) GPU 功耗估算模型方法^[4]:其主要思想是给出一种基于硬件性能技术事件的 GPGPU 功耗估算方法,通过分析 GPGPU 一些程序在运行时的功耗参数分布情况^[5],选择一组与这个应用程序运行时功耗密切相关性能技术事件的集合,通过 BP 神经网络来分析硬件性能技术事件和实时功耗之间的关系建立功耗估算模型。这种方法虽然能够大致估算出 GPU 的功耗数据,但是精度较低,而且要用到反向人工神经网络分析法,需建立在多组数据基础上,数据精度可信度不高。

2) 高精度体系结构级功耗模型^[6]:这种方法的主要思想是计算不同 native 指令及每次访问存储器消耗的功耗,然后根据应用在硬件上面的执行指令及采样工具得到采样结果,通过 13 个基准测试应用对事迹测试与功耗模型测试结果进行对比,分析预测其功耗。这种方法的弊端同样是精度较低,而且不能够获得实时的功耗数据。

3) 基于蓝牙 4.0 的 GPU 功耗检测方法^[7]:其思想是通过传感器采集 GPU 的电流,经过蓝牙 4.0 无线传感器网络将数据实时地发送到远程终端,但是其主要的缺点是蓝牙 4.0 无线传感器网络只有发送节点和接收节点,没有中间的路由节点,一旦其中的某个节点瘫痪,就可能使整个无线传感器网络中断,影响数据的实时发送。

考虑上述方法的特点,本文设计一种基于 ZigBee 技术的 GPU 集群功耗收集监控网络。利用 ZigBee 主芯片 CC2430 内含的 51 微处理器,将传感器采集到的模拟电压转化成数字电压,经串口通信将 GPU 功耗数据发送到本地电脑进行监控和分析,另外远程和本地终端都连接到数据库,实现了数据存储、回放的功能。通过在远程控制终端界面设定无线或者有线模式和发送周期命令,采集终端就会

实时地按照指定的采集周期采集 GPU 的功耗数据,并发送到远程或者本地监控终端进行实时的显示回放^[8]。本文系统在实验室的 GPU 集群上进行测试验证,实验结果表明,该系统可以精确地测量 GPU 集群的实时功耗,并且系统稳定性好、丢包率低、误差低,为 GPU 功耗优化提供了可靠的参考数据。另外设计的监控终端易于安装,方便维护,成本合理,适用于 GPU 集群功耗的测量,同时本文系统具有较好的可扩展性,如果通过连接路由器利用其端口映射实现数据转发功能,可扩展到智能手机等多移动终端进行实时监控,具有较高的实用价值^[7]。

2 GPU 的供电及计算任务

2.1 GPU 的供电分析

无论是 CPU 还是 GPU,如需获得更高的性能,则要付出较大的代价,即功耗。如果 GPU 的供电不足,当运行一些 3D 渲染的程序时很可能导致电脑死机、黑屏,甚至是电源和显卡烧坏。为解决上述问题,早期的 GPU 供电采用 AGP 来供电,3.3vAVX 辅助电源,它的作用是当系统挂起显卡停止工作时,能够提供很小的电源,进而使显卡迅速地恢复工作的状态,但是不管是 AGP、PCI,还是 PCIE 等接口,都需要 3.3vAUX 支持^[9]。

另外还有 VCC + 3.3、VCC + 5、VCC + 12 作为显卡主要的供电接口,三路电流分别是 6 A、2 A、1 A,而最新 AGP3.0 接口所能提供的最大功率也仅为 46 W。但这只是理论值,受种种条件的限制,并不能达到 46 W,实际只有 25 W 左右。因此,出现了 AGP-PRO 接口。AGP-PRO 接口类似于内存插口,分为 4 段,中间的一段是 AGP 插槽的 3.3 V 和 12 V 接口,它的目的主要是用来增强电压。AGP-PRO12 V 的供电功率为 110 W,AGP-PRO3.3 V 的供电功率为 25 W,所以主要供电方式为 12 V,PCIE 和 AGP-PRO 基本上也为 12 V 供电,去掉 + 5 V,保留了 3.3 V,以上基本上是低端显卡^[9]。

随着显卡的更新换代,对于稍微高端一点的显卡,或者用显卡运行一些通用计算和图像计算时,功率供给就较低,因此,目前显卡基本上都需要外接电源接口,主要是 6 Pin 或者 8 Pin 接口的外接 + 12 V 电源。如图 1 所示,对于 6 Pin 的接口,主要是由 3 个 + 12 V 和对应的 3 个 GND 构成,因此,主要通过测量外接电源接口的功率即可对 GPU 功耗进行监控^[9]。

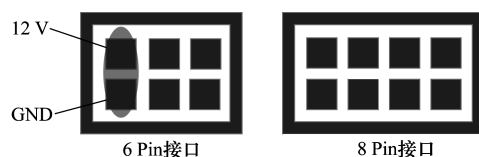


图 1 GPU 的供电接口

2.2 GPU 的计算任务

GPU 的计算分为图形计算和通用计算,而 GPU 的加速计算是指同时利用 GPU 和 CPU 进行计算的方式,以此来提高应用程序的运算速度^[10]。GPU 加速计算能够将应用程序计算密集型的部分加载到 GPU 进行计算,同时 CPU 负责应用程序剩余部分的计算^[11],它的重点是将 CPU 作为 Host, GPU 作为 Device 或者 Server,使 GPU 运行那些能够高度并行化的程序^[12]。

NVIDIA 推出统一计算设备架构 (Compute Unified Devices Architecture, CUDA)^[13],结合 GPU 可以大规模并行计算的特点^[9],使并行计算即可在普通的显卡上面进行扩展,用户只要有一台含有 NVIDIA 显卡的计算机就能进行并行计算。计算机能够将并行的任务分发到 GPU 的各个流多处理器 (Streaming Multiprocessor, SM) 上^[14],每个 SM 中含有多个流处理器 (Streaming Processor, SP)。统一计算设备架构线程模型如图 2 所示,每个核函数对应一个 Grid,每个 Grid 根据需要配置不同的 Block 数量和 Thread 数量^[12]。CUDA 包含 2 个并行逻辑层,分别是 Block 层和 Thread 层。在执行时 Block 映射到 SM, Thread 映射到 SP。而这也是造成 GPU 集群功耗变大的一个原因,因此能够准确实时地测量到 GPU 实时功耗数据,对于 GPU 功耗优化研究具有重要意义^[12]。

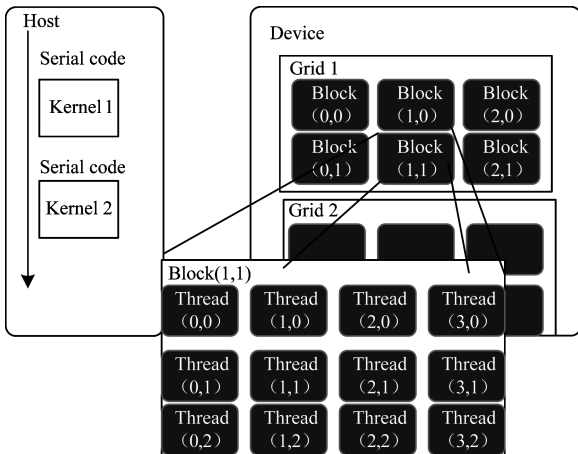


图 2 CUDA 线程架构

3 系统设计

GPU 集群中有很多 GPU 节点,每个节点的任务是不一样的,所以,每个 GPU 节点的功耗也是不一样的,预测出每个 GPU 节点的实时功耗对于 GPU 功耗优化研究是一个非常关键的步骤。

3.1 系统整体设计

如图 3 所示,系统的整体结构框架由终端传感采集节点、协调器数据收集、本地监控和远程监控终端组成,整个系统是一个无线传感器网络,实现对数

据的收集和转发的功能。本文系统的无线传感器网络采用了网状的拓扑结构,增强了无线传感器网络的稳定性和可靠性, GPU 节点通过终端采集节点将数据经过无线传感器网络发送到远程监控终端,或者是通过 USB 接口和本地的 PC 相连实现向本地发送数据并实时监控的目的。另外在本地和远程 PC 监控终端上都连接了数据库,实现数据的存储和回放,通过在远程终端软件设计实现了无线和有线可切换的发送方式。

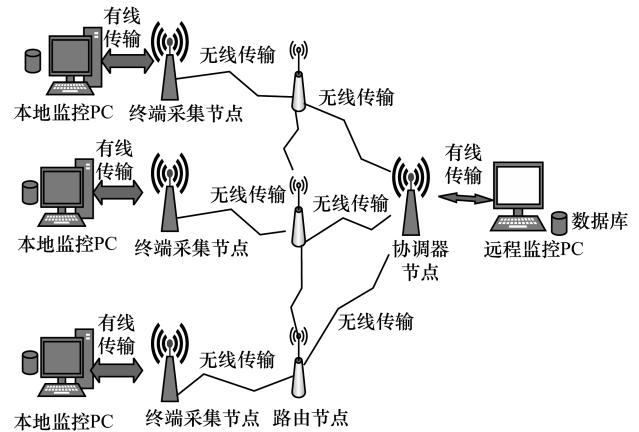


图 3 系统整体结构框图

3.2 电流感知传感器设计

电流感知传感器设计包括 2 个部分:感知部分与通信部分。感知部分即感知电流的大小;通信部分即将传感器感知到电流数据,通过 ZigBee 无线传感器网络将数据发送到监控终端。图 4 所示为电流感知设计框图,包括电流传感器部分和无线通信部分 2 个部分。传感器部分用的是霍尔电流传感器 WHB-LSP5S2H(量程 20 A)测量^[15],它的基本原理是将其传导通路中电流产生的电磁场转化为输出电压^[15],为了测量方便,通过显卡电源延长线将显卡供电线穿过霍尔电流传感器对其电流进行测量。

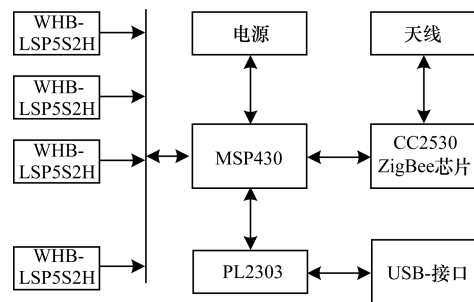


图 4 电流感知设计框图

当有电流通过时,根据线圈的电子感应会产生相应的电压,然后经过放大器对信号进行放大,输出模拟电压信号。霍尔电流传感器的工作原理如图 5 所示。

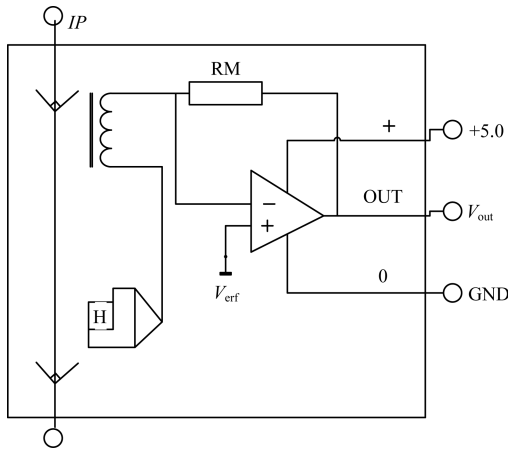


图5 霍尔电流传感器工作原理

从图5可以看出, I 是 GPU 供电线路的通路电流, V_{out} 是电流 I 经传感器转换后的模拟量输出, 由于模拟数字转换器 (ADC) 采样精度是 9 位, 因此参考 D_{out} 最大是 511, V_{ref} 是 ADC 的参考电压, 它通过微处理器的模拟数字转换器 ADC 转换后可得到数字量 $D_{out}^{[15]}$, V_{out} 与 D_{out} 之间的关系如式 (1) 所示。

$$D_{out} = \frac{V_{out} \times 511}{V_{ref}} \quad (1)$$

根据霍尔传感器的转换关系, 得:

$$V_{out} = 2.5 \pm \frac{2}{50} A \times I \quad (2)$$

所以, 电压和电流的关系为:

$$I = (V_{out} \pm 2.5) \times 25 \quad (3)$$

根据式 (1), I 和 D_{out} 的关系式为:

$$I = \left(\frac{D_{out} \times V_{ref}}{511} \pm 2.5 \right) \times 25 \quad (4)$$

其中, D_{out} 是 ADC 转换过后输出的数字量, I 是 GPU 的实际电流值^[1]。

无线通信部分用的是 ZigBee 无线通信协议, ZigBee 主芯片是 TI 公司的 CC2530^[16], 具有业界领先的 RF 收发器的优点, 并且是内含 8051 单片机的处理器芯片, 在 CC2530 的系统中进行编程和闪存, 有着 8 kB 的存储空间和多重运行模式, 非常适应于超低功耗的系统中, 多重模式之间可以迅速进行转换, 这样降低了功耗的消耗。而数据的处理部分采用的是 TI 公司的 MSP430 微控制器作为数据处理核心^[17], MSP430 里面有 12 个 12 位的模拟数字转换器, 可以更好地处理传感器节点的输出数据, 将 UART 接口与 ZigBee 芯片 CC2530 相连^[18], 实现将数据发送到 ZigBee 之后, 经过进一步处理, 再经过 ZigBee 协调器实现数据的无线发送和转发功能。ZigBee 协调器就是负责无线传感器网络的组件、数据包和指令的转发功能。

4 通信算法模型设计

本文系统的通信涉及到 2 个部分: 即无线通信部分和有线通信部分。整体的无线传感器网络由多个 ZigBee 终端节点 (由 ZigBee 和电流传感器采集节点组成) 和 ZigBee 协调器构成。另外, 还有一个远程监控终端和多个本地监控终端构成监控节点, 远程监控终端通过 USB 有线通信和 ZigBee 协调器相连并实现双向通信, 每个 GPU 节点又通过 USB 和自身计算机节点相连实现双向通信并构成本地监控节点。在远程监控终端设计监控界面来控制每个 ZigBee 终端节点向 ZigBee 协调器发送数据, 或者向本地计算机节点发送数据, 通过远程监控界面可以得到 ZigBee 终端节点的发送频率、数据采集频率和发送去向。另外, 远程监控和本地计算机监控都连接了数据库可以实现数据的存储、回放和界面的实时显示的功能。

4.1 无线通信算法设计

定义 1 (无线通信数据包) 结构为 $BGA(st, dev, value^{[3]})$, 其中, “ st ” 作为数据标准格式判断, 通常以字符 “&” 表示, “ dev ” 是由哪个终端所发出 (dev 取值: A, B, \dots 依次往下取决于所用到的终端数量), “ $value^{[3]}$ ” 存储由霍尔传感器所得到电流的数字量, 每一个 GPU 节点每一时刻都有对应的数据, 即数据的唯一性。

定义 2 (串口缓冲区) 在协调器终端与主控节点计算机及从节点计算机的数据交互数据库存储中, 需要用到串口通信, 将串口缓冲区定义为 $MScom.buf$ 。

定义 3 (串口数据包) 结构与无线通信包类似, 为 $BGP(st, dev, value^{[3]})$, 其中, “ st ” 作为数据标准格式判断, 通常以字符 “&” 表示, “ dev ” 是由哪个终端所发出的 (dev 取值: A, B, \dots 依次往下取决于所用到的终端数量), “ $value^{[3]}$ ” 存储由霍尔传感器所得到电流的数字量。

定义 4 (串口命令数据包) 结构为 $BGC(wx, dm, ed)$, 其中, “ wx ” 是变量可取字符为 “Y” (表示有线模式的起始) 和 “N” (表示无线模式的起始), “ dm ” 存储四位数的采样间隔发送时间, 单位为毫秒, “ ed ” 代表该命令结束通常以字符 “!” 作为结束判断。

定义 5 (数据库存储数据包) 结构为 $BGS(tm, dev, current)$, 其中, “ tm ” 指当前时间, “ dev ” 作为数据存储标号, 说明电流数据来自哪个终端, “ $current$ ” 指数字量转换后的实际电流值。

定义 6 (延时时间) 采用一个变量 $Time_Delay$, 用户可以根据需要自己设定终端采样的间隔。

图 6 所示为远程终端监控模型, 其中虚线为无线通信, 实线为有线通信。一共包括 4 个过程, 分别为发送、接收、处理和存储。

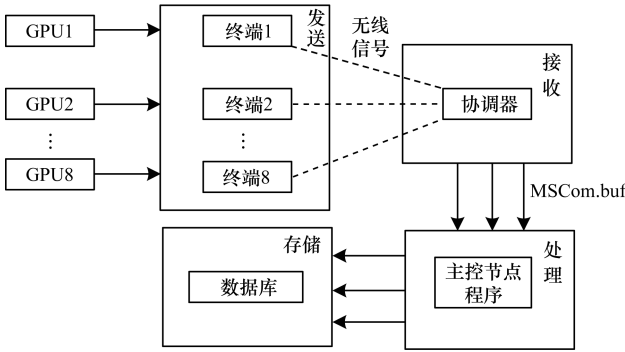


图 6 远程终端监控模型

过程 1 SEND 过程

初始化:初始化通信协议,初始化串口设置

- 1) 进入就绪状态,等待组网成功;
- 2) 判断组网是否成功;
- 3) 延时 $Time_Delay$ (前后采样间隔);
- 4) 霍尔传感器采集当前 GPU 电流;
- 5) 存储数据包 $BGA(st, dev, value^{[3]})$;
- 6) 将 BGA 数据包发送给终端。

过程 2 RECV 过程

- 1) 触发协调器接收无线信号事件(当有无线数据发送过来时);
- 2) 接收 BGA 数据包;
- 3) 判断数据包格式是否满足标准(如果满足,则将当前数据包的数据拆分后重新存入 BGP 数据包中,放入串口缓冲区 $MSCom. buf$,等待主控节点程序来读取该数据包)。

过程 3 PROC 过程

- 1) 选择连接协调器的串口号;
- 2) 打开串口;
- 3) 判断 $MSCom. buf$ (如果 $MSCom. buf > 0$,则触发主控节点控制程序向缓冲区读取数据包,如果 $MSCom. buf = 0$,则不做处理,直到过程 2 中有 BGP 传入 $MSCom. buf$ 中);

4) 剔除 $trans(BGP. value)$ 中小于 0 受干扰的异常值(通过数字量到模拟量传感器转换过程);

5) 判断 $BGP. dev$ 将正常数据描绘到所对应的图形界面上和数值显示界面上。

过程 4 SAVE 过程

1) 将该包发送过来的时间存储到 $BGS. tm$ (根据计算机本地时间获取该包时间);

2) 将存储的终端号存储到 $BGS. dev$ (根据 $BGP. dev$ 确定);

3) 根据 PROC 过程中的 $trans(BGP. value)$ 返回值存储到 $BGS. current$;

4) 按照上述顺序将数值存储至数据库($BGS(tm, dev, current)$)。

4.2 有线通信算法设计

监控模型共分为本地监控和远程监控,图 7 所示为本地监控模型。

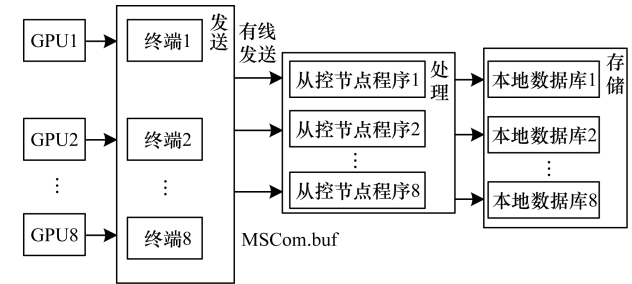


图 7 本地监控模型

过程 1 WSEND 过程

初始化:初始化通信协议,初始化串口设置

- 1) 进入就绪状态,等待组网成功;
- 2) 判断组网是否成功(虽然是有线发送,还是要保证终端在组网状态中,这样才能接收从协调器发来的模型转换命令);

3) 延时 $Time_Delay$ (前后采样间隔);

4) 霍尔传感器采集当前 GPU 电流数字量;

5) 发送电流转换的数字量 $value$ 到缓冲区。

过程 2 WPROC 过程

1) 选择本地终端的终端号;

2) 选择终端连接的串口号;

3) 点击开始采样;

4) 从节点控制程序每过 1 ms 向串口缓冲区检查,如果 $MSCom. buf > 0$,则主控节点控制程序向缓冲区读取数据包,如果 $MSCom. buf = 0$,则不做处理,直到过程 2 中有 BGP 传入 $MSCom. buf$ 中;

5) 剔除 $trans(value)$ 中小于 0 受干扰的异常值;

6) 将正常删选过的数值显示在数值栏和坐标系中绘图。

过程 3 WSAVE 过程

1) 根据计算机本地时间获取该包发送过来的时间存储到 $BGS. tm$;

2) 根据界面中所选择的端口号确定存储的终端号存储到 $BGS. dev$;

3) 根据 WPROC 过程中的 $trans(value)$ 返回值存储到 $BGS. current$;

4) 按照上述顺序将数值存储至数据库($BGS(tm, dev, current)$)。

5 实验结果与分析

本文主要从测量误差、丢包率、同一程序在不同任务下以及不同温度下的功耗情况进行多组实验和分析。

5.1 精度测量及丢包率分析

图8所示为测量系统在 GTX970、GTX680、GTX670 3种不同的显卡上面分别运行 14种不同的程序,以此来检验系统的测量精度,结果显示,测量精度基本在 1% 以下,中间存在个别的误差值。

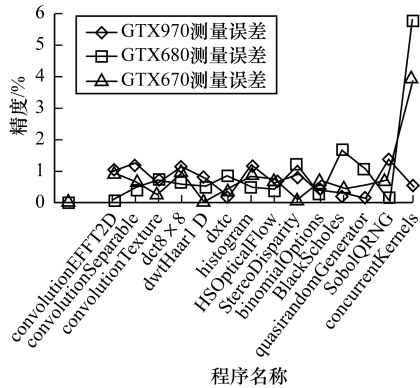


图8 系统测量精度比较结果

图9所示为系统的丢包率实验结果。为精确验证系统丢包率情况,本文采用控制变量法,通过改变采样周期以及终端节点个数来比较系统丢包率的变化情况。实验结果表明,当采样周期比较大时,系统的丢包率基本为 0%,在 8 个节点时,丢包率为 0.005%,随着节点个数和采样周期的减小,系统的丢包率会随之增加,并且当采样周期相对小时,随着节点个数的增加,丢包率的增长率越大。

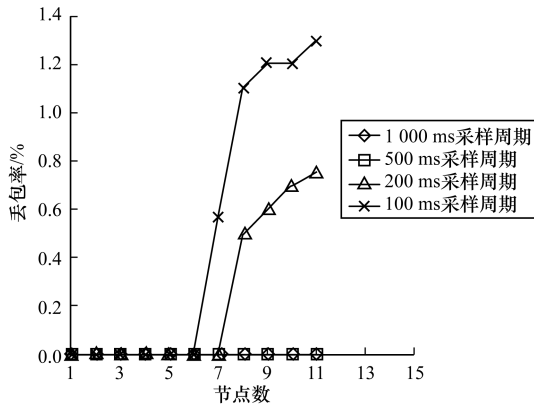
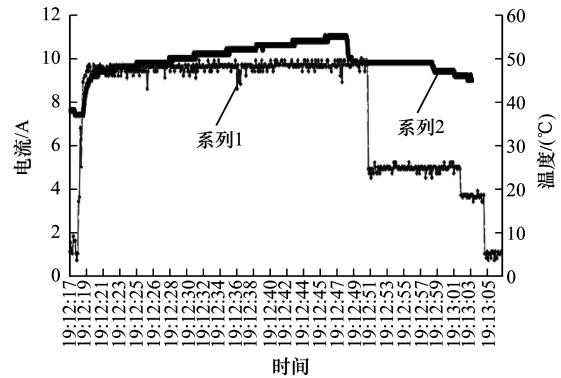


图9 系统丢包率折线图

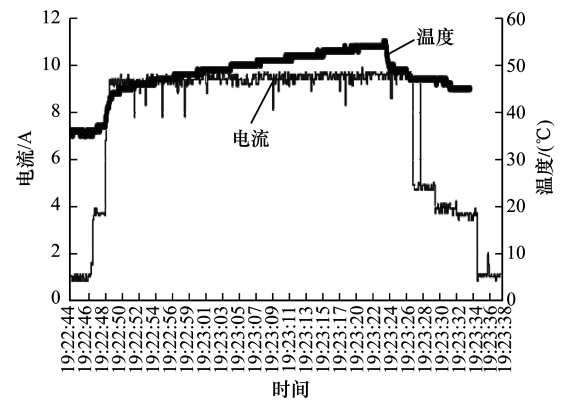
5.2 不同程序类型下的 GPU 功耗

图10~图12分别为存储、计算、延迟密集型3种不同类型的程序功耗变化曲线,可以看出,存储密集型程序的功耗变化情况为当程序开始运行时,功耗将很快达到峰值状态,当程序运行将要结束时功耗会慢慢下降直至为0;计算密集型程序的功耗变化情况和存储密集型变化情况相比,功耗的峰值相对小些,而且末尾电流变化下降速度更快;延迟密集型程序的功耗情况为功耗的峰值较以上2种类型更

小,而且峰值的周期相对较短,功耗的上升和下降较为缓慢。

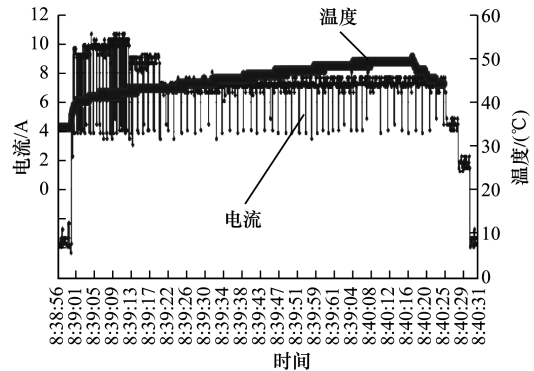


(a)程序功耗图1

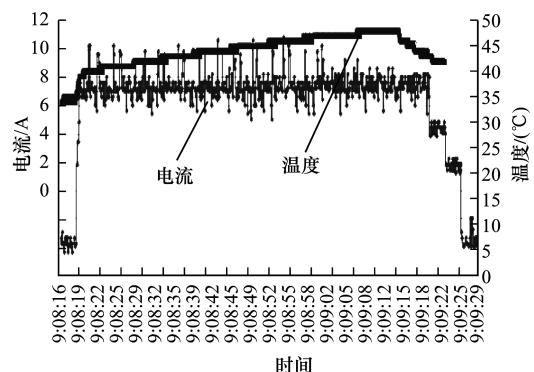


(b)程序功耗图2

图10 存储密集型程序功耗示意图



(a)程序功耗图1



(b)程序功耗图2

图11 计算密集型程序功耗示意图

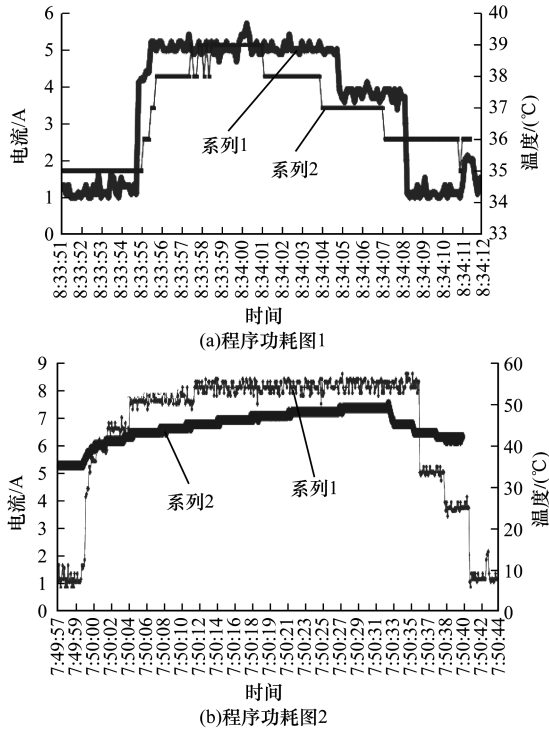


图 12 延迟密集型程序功耗示意图

5.3 同一程序在不同温度下的功耗

图 13 所示为同一程序在不同温度下功耗测量情况^[19]。实验结果表明,同一程序在不同的温度下所测得的功耗数据是不一样的,初始温度更高时,测得的功耗数据会更大些,但功耗的变化情况不明显。

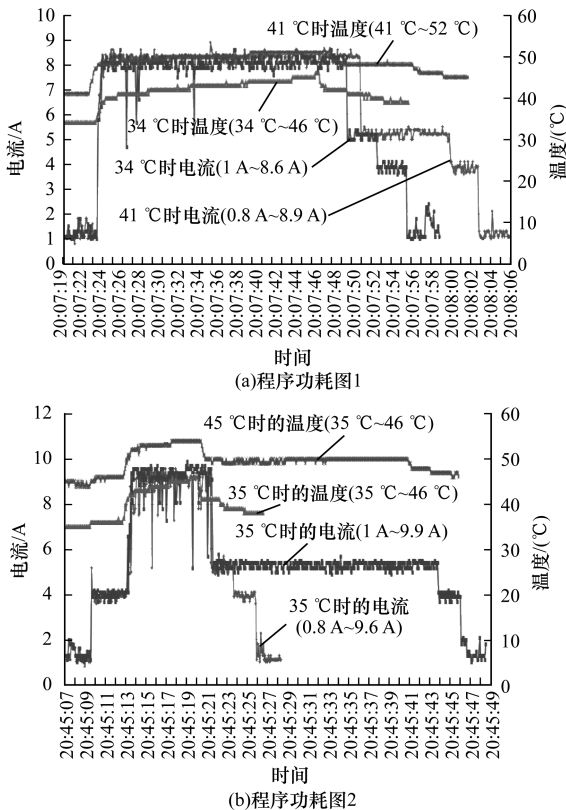


图 13 不同温度下功耗的曲线示意图

图 14 为系统的软硬件实物图,其中,左面为本地监控界面,包括串口号选择、采样周期、实时数据、实时数据曲线图,右边为硬件测量系统部署在 GPU 集群上面的实物图。图 15 为远程监控界面截图,包括采样周期、端口号、无线有线发射切换、各个节点实时数据曲线图。

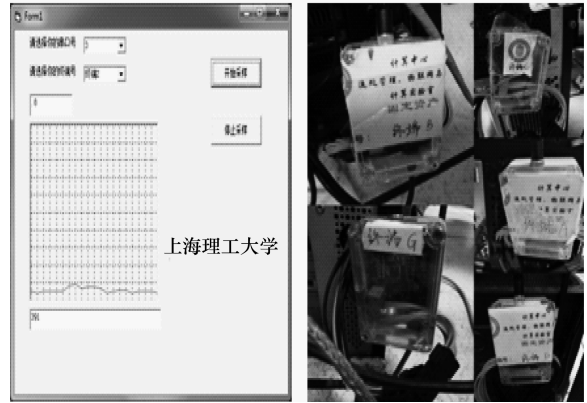


图 14 系统软硬件实物框图

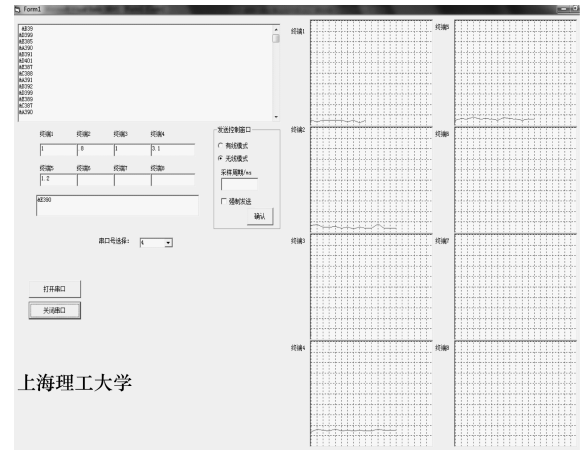


图 15 远程监控界面

6 结束语

本文给出一种 GPU 集群功耗传感器,并设计基于 ZigBee 技术的 GPU 集群功耗收集监控网络系统。该系统由终端传感器电流采集节点、协调器节点和本地及远程监控终端组成,传感器节点部署在 GPU 上,并且每个 GPU 部署一个终端采集节点,将采集到的 GPU 实时功耗数据通过 ZigBee 构建的无线传感器网络传输到协调器,经 USB 串口通信发送到监控终端进行实时监测分析。通过在远程控制终端界面设定无线或者有线模式和发送周期命令,采集终端会实时地按照指定的采集周期采集 GPU 的功耗数据,并发送到远程或者本地监控终端进行实时显示回放。实验结果表明,该系统精度高、稳定性好,同时具有较好的可扩展性。

参 考 文 献

- [1] 王卓薇,程良伦,肖红.一种基于GPU的高精度体系结构级功耗模型[J].计算机科学,2016,43(11):30-34.
- [2] 王桂彬.基于硬件性能计数器的GPU功耗预测模型[J].计算机工程与科学,2012,34(3):46-50.
- [3] 江修波.ZigBee技术及其应用[J].低压电器,2005,55(31):27-30.
- [4] 王吉军,程华.通用图形处理器功耗估算模型[J].计算机应用,2017,43(2):92-95.
- [5] DAS S K, SUDHAKARAN G, ASHOK V. Learning based performance and power efficient cluster resource manager for CPU-GPU cluster[C]//Proceedings of the 4th International Conference of Emerging Applications of Information Technology. Washington D. C., USA: IEEE Press, 2014: 161-166.
- [6] 王卓薇,程良伦,肖红.一种基于GPU的高精度体系结构级功耗模型[J].计算机科学,2016,43(11):30-34.
- [7] 李杰,陈庆奎.基于蓝牙4.0的GPU集群功耗测量系统设计[J].电子测量与仪器学报,2014,28(3):314-320.
- [8] VIVIAN D, MANNEBAVK P, PINEL F. Comparing the performance and power usage of GPU and ARM clusters for map-reduce[C]//Proceedings of International Conference on Cloud and Green Computing. Washington D. C., USA: IEEE Press, 2013: 199-200.
- [9] 黄昏.主板和显卡是如何给CPU/GPU供电的[J].个人电脑,2017(1):84-87.
- [10] 王海峰,陈庆奎.图形处理器通用计算关键技术研究综述[J].计算机学报,2013,36(4):757-772.
- [11] ZHANG C, HUANG K, CUI X. Programming-level power measurement for GPU clusters[C]//Proceedings of 2011 IEEE/ACM International Conference on Green Computing and Communications. Washington D. C., USA: IEEE Press, 2011: 182-187.
- [12] 董萃,葛万成,陈康力.CUDA并行计算的应用研究[J].信息技术,2010,17(4):11-15.
- [13] SHOWERMAN M, ENOS J, STEFFEN C. EcoG: a power-efficient GPU cluster architecture for scientific computing[J]. Computing in Science and Engineering, 2011, 13(2): 83-87.
- [14] SILVA L, RODRIGUES G, MEDEIROS D. CUDA-based parallelization of power iteration clustering for large datasets[J]. IEEE Access, 2017(99): 1.
- [15] 王昱皓,钟贻兵,时圣利.高可靠霍尔电流传感器的研究和应用[J].新型工业化,2015,11(5):8-11.
- [16] 畅秋.ZigBee发展方兴未艾[J].电子设计应用,2006(10):128.
- [17] 陈鹏.MSP430系列芯片简要介绍[J].电子技术应用,1998,21(7):57-63.
- [18] 杨柳.基于GPU的并行集群系统的发展[J].中国外资,2013(22):269.
- [19] CAO Y, WANG H. Task scheduling scheme for preventing temperature hotspot on GPU heterogeneous cluster [C]//Proceedings of 2017 International Conference on Green Informatics. Washington D. C., USA: IEEE Press, 2017: 117-121.
- [4] RAYCHAUDHURI D, NAGARAJA K, VENKATARAMANI A. Mobilityfirst: a robust and trustworthy mobility-centric architecture for the future internet [J]. ACM SIGMOBILE Mobile Computing and Communications Review, 2012, 16(3): 2-13.
- [5] XIA W, WEN Y, FOH C H, et al. A survey on software-defined networking[J]. IEEE Communications Surveys and Tutorials, 2015, 17(1): 27-51.
- [6] 谢凡,李勇,柳嘉强,等.基于SDN-NFV的移动核心网联动机制研究[J].计算机工程,2017,43(2):144-149.
- [7] WU J, NI H, ZENG X, et al. A storage approach for OpenFlow switch based on protocol oblivious forwarding[C]//Proceedings of 2016 IEEE Symposium on Computers and Communication. Washington D. C., USA: IEEE Press, 2016.
- [8] KALGHOUM A, GAMMAR S M. Towards new information centric networking strategy based on software defined networking[C]//Proceedings of 2017 IEEE Wireless Communications and Networking Conference. Washington D. C., USA: IEEE Press, 2017: 1-6.
- [9] 田静.海云协同,走向未来(序言)[J].网络新媒体技术,2014,3(1):1.
- [10] 卓煜,尤佳莉,王劲林,等.海服务中面向在线视频服务的测量与推荐系统[J].计算机工程,2018,44(4):28-34,40.
- [11] SOLIS I, MOSKO M, SCOTT G, et al. CCNx 1.0 tutorial: theory and practice[C]//Proceedings of the 2nd International Conference on Information-Centric Networking. New York, USA: ACM Press, 2015: 1-2.
- [12] LAOUTARIS N, CHE H, STAVRAKAKIS I. The LCD interconnection of LRU caches and its analysis [J]. Performance Evaluation, 2006, 63(7): 609-634.
- [13] LI Z, SIMON G. Time-shifted TV in content centric networks: the case for cooperative in-network caching[C]//Proceedings of 2011 IEEE International Conference on Communications. Washington D. C., USA: IEEE Press, 2011: 1-6.
- [14] ZHU X, QI W, WANG J, et al. A popularity-based collaborative caching algorithm for content-centric networking [J]. Journal of Communications, 2016, 11(10): 886-895.
- [15] SONG H. Protocol-oblivious forwarding: unleash the power of SDN through a future-proof forwarding plane [C]//Proceedings of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking. New York, USA: ACM Press, 2013: 127-132.

编辑 索书志

(上接第64页)

编辑 司森森