

并行程序性能和功耗的测试分析工具研究

潘晓东, 孙晓乐, 郑文旭, 吴 丹
(国防科技大学 计算机学院, 湖南 长沙 410073)

摘 要:高性能并行计算在社会的各个方面发挥了重要作用,然而计算性能发展所带来的高功耗已经成为阻碍其进一步提高的重要因素。除了在硬件制程工艺等方向进行优化外,从软件角度来提高系统的能量有效性是当前高性能计算领域研究的热点问题。其中对并行程序的性能和功耗特征进行检测分析,可以有助于研究人员了解程序行为,找出程序性能瓶颈和功耗的热点区域,发现程序的能量有效性优化的方向。方便准确的并行程序性能和功耗的测试分析工具对于并行程序开发和能量有效性的优化有着极其重要的作用。该文对并行程序性能和功耗的测试工具进行了对比研究。首先对并行程序性能和功耗测试的原理进行阐述;其次,对当前国外超算中心常用的分析和测量软件进行介绍和对比;最后进行了总结。

关键词:高性能计算;性能分析工具;功耗检测;并行计算;能量有效性;超算中心

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2021)07-0069-06

doi:10.3969/j.issn.1673-629X.2021.07.012

A Survey of Performance and Power Measurement and Analysis Tools for Parallel Programs

PAN Xiao-dong, SUN Xiao-le, ZHENG Wen-xu, WU Dan
(School of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract: High-performance parallel computing has played a major role in all aspects of society, but the high power consumption brought about by the development of computing performance has become an important factor hindering its further improvement. In addition to optimization in the direction of hardware process technology, it is a hot issue in the field of high-performance computing to improve the energy efficiency of the system from the perspective of software. Among them, the measurement and analysis of the performance and power consumption characteristics of parallel programs can help researchers understand program behavior, find out program performance bottlenecks and power consumption hot spots, and find the direction of energy efficiency optimization of programs. Convenient and accurate parallel program performance and power measurement and analysis tools play an extremely influential role in parallel program development and energy efficiency optimization. We summarize the measurement and analysis tools of for performance and power measurement parallel program. Firstly, the principle of performance and power consumption measurement of parallel program is described. Secondly, the software commonly used in foreign supercomputing centers is introduced and compared. Finally, the summary is made.

Key word: high performance computing; performance analysis tools; power consumption measurement; parallel computing; energy efficiency; supercomputer center

0 引 言

高性能计算(high performance computing, HPC)现在已经成为促进军事进步、科技创新和经济发展的的重要手段,无论是科学计算还是新兴的人工智能都对HPC的性能提出了新的更高的要求。随着 Dennard 缩放效应的失效,处理器单核的频率到达极限,多核、众核并行以及异构并行成为 HPC 发展的趋势,同时随之

不断增长的能耗成为阻碍 HPC 发展的重要因素^[1], HPC 开始转向高效能并行方向。

HPC 能量有效性提高的策略大致可以分为系统和应用两方面。从系统角度就是调节与运行相关的系统硬件参数或者软件环境参数,例如静态的基础架构节能、电路的低功耗设计、运行时基于硬件的参数调优^[2];以及系统各部件的动态电源管理技术(DPM),

收稿日期:2020-05-20

修回日期:2020-09-21

基金项目:国家重点研发计划(2018YFB1003203)

作者简介:潘晓东(1989-),男,硕士研究生,研究方向为高性能计算、并行计算;吴 丹,硕士,高级工程师,研究方向为高性能计算、并行通信。

包括任务映射(mapping)、器件动态休眠、动态电压频率缩放(dynamic voltage and frequency scaling, DVFS)^[3]等。而从应用角度而言是在应用执行之前通过对并行程序本身进行优化来达到节能的目的。包括常用的软件代码优化、编译优化和运行库优化^[4]。

这两方面策略的制定,都需要对应用进行性能和功耗的检测分析,并作为基准对策略部署后的情况进行对比,以此来迭代修订策略,同时在 HPC 运行当中往往也需要对性能和功耗数据进行检测汇总。并行计算的性能检测分析与体系结构、并行算法、并行程序设计一同构成了并行计算研究的四大分支。

长期以来国内高性能计算的基础并行环境落后于计算机系统的发展^[5],软件的性能分析工具(performance analysis tools, PAT)也是如此。国内出现过一些并行分析工具,如中科院的 ParaVT、清华大学的 VIMP、曙光上的 Para Vision 以及面向云计算资源监控的并行科技的 Paramon,尽管使用分析工具对于理解 HPC 应用中的程序行为、性能瓶颈和优化潜力很有帮助,但由于使用太复杂或者是售价和推广的原因,这些工具并没有流行起来。通过调查,发现许多实验室在进行性能和功耗检测时,最常用的方法仍是手工插桩(使用时间戳)输出日志,再进行统计后处理的方法,这种方法虽然简单直观,但是工作量大,并且得到的信息量过少,缺少对软件运行的整体认识,后续处理也相对困难,给并行程序的能耗和性能调优带来了不便。

随着 HPC 系统复杂性的提高,以及大规模应用软件开发的需要,手工检测和分析的方法已经不能达到要求,因此该文介绍了性能和功耗检测的原理和相关工具,并对 4 类跟踪和分析工具进行了对比分析,以便在后续工作中使用。

1 性能和功耗检测原理

针对不同平台和不同编程模型,性能和功耗检测也有所区别。并行程序的编程模型分为消息传递和共享主存两种模型,消息传递接口(message passing interface, MPI)和 OpenMP(open multi-processing)已经成为两种模型的代表,而异构计算引入了在加速器上使用的和 OpenMP 类似的 CUDA(compute unified device architecture)、OpenCL(open computing language)以及 OpenACC(open accelerators)。现在的并行性能和功耗检测多集中于采用 MPI、OpenMP 和 CUDA, OpenCL 和 OpenACC 在国内超算集群中应用较多,但目前缺乏成体系的检测工具。

1.1 性能检测

计算机的理论峰值性能是所有处理器性能的总

和,而实际性能受限于系统和软件,使得程序执行过程中集群有很多处理器处于松弛状态(即空闲或阻塞),这样性能无法达到最优,松弛状态的功耗也就浪费了。如果不能掌握程序的计算性能、通信状况、处理器状态、内存状态等性能数据,对程序的优化调整就无从下手。HPC 程序测试中通常采用动态测试的办法,在中小规模核心数目上进行实际运行并收集性能数据,经过整理得出分析结果汇报给使用者。

性能检测一般分为确定数据采集种类、获取和记录数据、数据分析表达几个步骤。性能检测方式有追踪(tracing)和轮廓(profileing)两种方式^[6]。前者采用插桩技术(instrumentation),对应用程序进行修改,在程序中插入附加指令,详细记录程序执行过程中发生的事件,从而进行事件跟踪(event-based tracing),通过日志文件记录,之后可以用时间轴的方式来呈现结果,帮助用户了解程序执行过程中的详细特征。后者的典型形式是对路径调用(call-path)信息或者硬件计数器进行统计,即在并行程序执行过程中,定期对程序当前执行的指令和对执行过的函数堆栈进行回溯,得到子程序、基本块和语句的执行时间和执行次数等信息。

性能检测的步骤如图 1 所示。

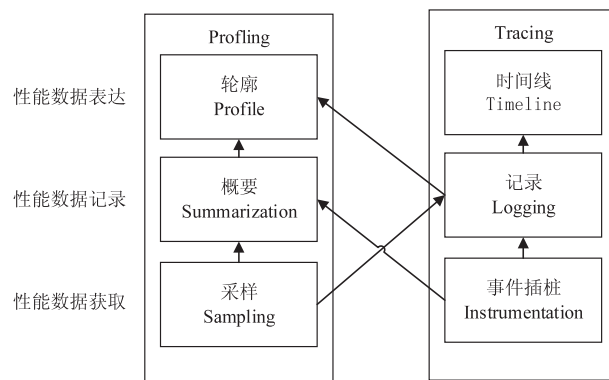


图 1 性能检测的三个步骤

1.1.1 数据选择

并行程序一般用运行时间、加速比和并行效率来衡量性能。因此通常会选择 MPI 通信时间、通信量、程序各阶段运行时间,以及各硬件性能计数器数据作为需要采集的数据。

1.1.2 数据获取和记录

数据获取主要有两种方法,其中插桩技术也被称为软件打点技术,目的是通过在程序中插入指令来获取程序的状态,可以分为三种:(1)源代码插桩,在程序源代码或库的源代码中直接插入性能分析函数,这种最为直观,对于实验中使用的测试程序的源代码可以根据语法和语义信息,准确进行标记。由于手工插桩较为繁琐,因此也可以采用自动化分析插桩工具如

Program Database Toolkit (PDT) 来减少工作量;(2) 二进制插桩,通过对静态或者运行中的可执行文件进行二进制代码插入,不需要重新编译程序,但是插桩的开销较大,主要工具有 Pin 和 Dyninst;(3) 库替换 (instrumentation-library interposition),使用带有测试接口的 .so 动态库调用替换 I/O、MPI、CUDA、OpenCL 的 API 以及内存分配/释放例程,通过拦截应用程序对共享库的函数调用,来获得调用数据。

而另一种是采样,是一种统计方法,按照一定的周期程序的性能数据进行采样和记录,据此来分析应用程序部分代码的性能特征,以代表整个应用程序特征。采样不需要对应用程序做修改,操作简单,开销小,但是不能获得应用的完整性能视图,并且很难确定合适的采样频率。

在获取性能数据的过程中也会根据检测目的来进行选择,或者是同时使用两种采集方式。在采集的同时会将采集到的数据记录到特定格式的文件中去,以便分析工具进行处理。记录的形式一般分为记录的类型、记录的时间戳、性能计数器的读数等。

1.1.3 数据表达

数据表达也就是将采集到的数据通过报告或图表的方式展示给用户。根据数据分析整理的时机可以分为在线分析和离线分析,在线分析就是在程序运行时实时分析程序的运行状态和性能特征,将分析结果实时展现给用户;离线分析是在程序执行期间收集性能数据并进行存储,在程序运行完毕后,基于程序运行时的全部性能数据进行分析 and 展示。由于当前高性能并行程序的规模大,运行时间较长,需要对多节点多方面的性能数据进行综合分析,在线分析无法满足要求,因此离线分析是实际中常用的研究方法。

1.2 功耗检测

由于高功耗对高性能计算发展的制约,以及带来的高昂运行和维护成本,对 HPC 以及程序进行功耗测量,以掌握其工作情况也变得相当重要。早期的功耗研究使用基于模拟器的测量分析,可以在硬件设计阶段用来平衡性能和功耗^[7],但由于精度低和速度慢,实际难以推广使用。实际使用的功耗检测方法一般分为两类^[8]。

(1) 基于设备直接测量结果的硬件测量法。是指用各种外接或者集成的传感仪器设备来测量硬件设备的电流和电压,接着使用测量值来计算被测对象的功耗值,如使用智能配电单元 (intelligent power distribution units, iPDU)、传感器电阻、Plug-wise 智能插头、Watts up pro 功耗计、IPMI (intelligent platform management interface)、数字万用表等^[9]。这种方式精度很高,但是缺点也很明显,如基于硬件的测量接口各

异,数据只能离线记录到运行机器之外的设备上,不能够对部件内部的更小器件进行功耗分析和测量,成本较高。

(2) 基于性能事件或者寄存器读数的软件方法。操作系统提供的系统事件可以反映软硬件状态,并且现代主流的处理器中都集成了硬件性能监控计数器 (performance counter, PMC),可以用来监控硬件相关活动事件的发生次数,更直观反映其硬件使用情况^[10]。因此结合系统事件和 PMC 事件的读数来建立模型,可以做到实时性和非侵入性,现在使用的较多^[11]。常见的采用软件方法的功耗检测工具有 PowerTOP、Likwid-Power meter、PAPI。

其中 PAPI (performance application programming interface)^[12]使用更为广泛,PAPI 是田纳西大学开发的,可以在多种平台上对硬件性能计数器进行访问的标准接口,可以监测和采集由计数器记录的处理器事件信息。对程序运行中带来的功耗,更多关注处理器、加速器及内存的动态功耗。PAPI 既提供了程序调用接口,也提供了一些组件的 PAPI-C 接口对上述部件进行功耗检测,如 NVML (Nvidia management library)、RAPL (running average power level)、libmsr 的接口组件。可以在外部工具中调用 PAPI 中的这些组件接口来获取功耗值,因此为性能和功耗的协同检测提供了可能,例如 TAU、Extrae 就已经支持在追踪性能数据的时候调用 PAPI 来提供性能和功耗数据。

2 性能和功耗分析工具

性能和功耗的检测数据采集之后,往往需要某种方式来呈现给用户,以便用户从大量数据中找出规律,发现程序性能的瓶颈和能量优化的机会。这样就需要一些工具来对数据进行处理,工具应当具有可视化、易操作等特性。由于当前没有比较完善的能耗可视化工具,通常是调用 PAPI 或者第三方硬件仪器的接口来采集能耗信息,并且在性能可视化工具中进行展示,因此这里选用了几类常见的支持 PAPI 的性能可视化工具。

2.1 基于 Score-P 检测的工具

由于传统的性能测试工具采用了各自不同的测量系统和输出格式,给使用带来不便。但是各种测量工具在 MPI 函数插桩、事件记录和数据记录格式上往往具有相同的特征,这给统一测量工具提供了可能。针对这个问题,SIILC (scalable infrastructure for the automated performance analysis of parallel codes) 项目开发了 Score-P^[13] 测量系统,采用了开源数据记录格式 CUBE4 和 OTF2 (open trace format V2)^[14]。Score-P 支持 MPI 框架,支持 CUDA、OpenACC 和 OpenCL,

也支持使用 PAPI 来获得硬件性能技术器和功耗信息。并得到了 Vampir、Scalasca、TAU 和 Intel Trace Analyzer 等分析软件的支持(见图 2)。

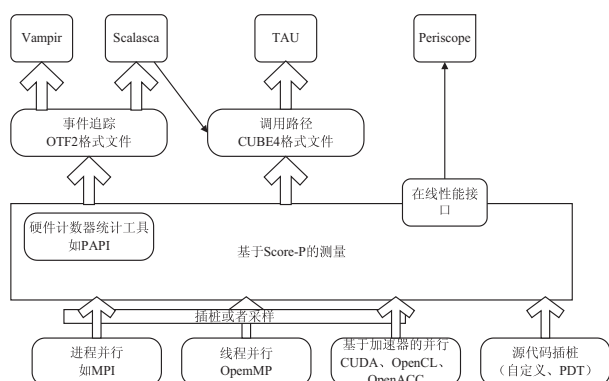


图 2 基于 Score-P 检测工具

Score-P 需要对用户程序在编译和链接时进行插桩,需要在相应命令前添加 scorep 前缀来替换成 Score-P 的插桩命令,如 mpicc 就需要更改为 scorep mpicc。通过插桩后生成的可执行程序通过 mpirun 或者直接执行完成后,就会生成 profile 文件和 trace 文件,以供下列几种工具分析。Score-P 支持 C/C++ 和 Fortran,但是安装和参数设置较为繁琐,并且不能对未使用 Score-P 编译的二进制可执行程序进行检测。

2.1.1 Vampir

Vampir (visual and analysis of MPI resource) 是由 Pallas GmnH 公司开发的商业工具,主要用来分析并行程序执行过程中的 MPI 调用,以便了解通信模式、发现程序热区和性能瓶颈,可以提供事件和通信时间的详细时间视图、程序执行分析、通信操作的统计分析等功能,广泛应用于许多超算中心。Vampir 提供了时间线视图, MPI 通信矩阵,性能计数器视图和 MPI 信息详情。

Vampir 的安装和使用是在基于 Score-P 的可视化分析工具中最为方便的,并且提供了上述丰富的分析功能,可以直观看出程序的 MPI 详细通信过程。缺点是只针对 MPI,没有其余函数调用的分析。

2.1.2 Scalasca

Scalasca (scalable performance analysis of large scale applications) 是由德国于利希研究中心以及模拟科学学院开发的开源工具,通过 Scout 中预定义的低效应用程序行为模式,利用自动分析工具识别 MPI 中的不平衡到达 (unbalanced process arrival, UPA) 等问题。Scalasca 更多像是一个列表式的分析工具,分析出结果之后,输出给 CUBE-GUI 显示分析结果,并可以通过输出 Cube4 格式到 Vampir 或 TAU 中来进行可视化,Scalasca 和 Cube 不支持时间线视图。

2.2 Intel Parallel Studio

Intel 公司的 Parallel Studio 也是普遍使用的商业

并行程序性能检测和分析工具套件,ITAC (Intel trace analyzer and collector) 是 MPI 通信的分析工具,Vtune Amplifier 是其中的多线程性能分析组件。

ITAC 主要关注基于 MPI 的通信,该部分提供基于事件的追溯机制,提供计算能力统计、线程跟踪功能,对于资深的开发人员还可以通过插入二进制指令对 MPI 函数调用以及 MPI 通信信息的参数进行动态调试。ITAC 支持底层 MPI 和硬件配置的调优,并且可以进行 MPI 正确性的检查,主要用于跨节点的并行程序通信性能分析。

Vtune 针对程序运行过程中产生的事件计数,可以进行多线程的代码分析、堆栈采样和硬件事件采样,然后利用二进制文件中的编译信息将样本统计数据对应到进程、线程、函数、反汇编代码和源代码中^[15]。Vtune 通过 Intel 处理器上的片上性能监控单元,可以搜索由于缓存失效和错误分支预测而导致的系统停滞,帮助用户识别性能瓶颈。Vtune 主要用于各节点内性能分析。

Intel Parallel Studio 在并行程序应用中也较为广泛,但是缺乏对各类非 MIC 加速器和非 Intel x86 处理器的支持,并且由于是商业软件,售价高昂,不利于在大规模集群上进行使用。

2.3 Nvidia 性能分析工具

在进行基于 CUDA 的并行程序开发时,往往会使用 Nvidia 公司的开发工具和性能分析工具,如源代码的可视化 CUDA 开发环境 Nsight、可执行文件的性能可视化分析工具 Visual Profiler (NVVP)。工具对 Nvidia 的 API 支持最为全面,主要用来精确定位 CPU 和 GPU 之间的负载不平衡或 GPU 上的性能瓶颈,以优化程序的配置。但是对于多节点的大规模并行并不适用,通常需要和 MPI 分析工具配合使用。可以查看插桩开销、GPU 函数、主从设备交换等时间线,并可以通过 Nvidia Tools Extension 插桩来查看 CPU 上具体函数的时间线视图,同时提供 GPU 的功耗和能耗的输出。优点是和本公司的计算卡结合紧密,在 GPU 异构计算上结合开发、调试和性能分析于一体,使用方便。缺点是只支持本公司的计算卡,并且对于主设备如 CPU 上运行的非 OpenACC 程序,就只能使用工具进行手动插桩。

2.4 BSC Tools

BSC Tools 是一个由西班牙巴塞罗那超级计算中心开发的开源工具集,主要由 Paraver (可视化分析器)、Extrae (测量工具) 组成,还有一些额外的集群性能分析工具。Extrae 使用不同的插入机制 (库替换、手动插桩、dyninst 二进制插桩) 将探针注入到目标应用程序中,以便收集性能指标。Extrae 还使用 PAPI 接口

来收集有关硬件性能的信息,允许在并行调用发生时捕获此类信息,而且还可以在用户例程的入口和出口点捕获这些信息。Paraver 是一个非常强大的开源可视化分析工具,该工具主要是支持用户自定义模式和全局概览。Paraver 通过时间线和概要统计来分析数据记录中的运行时间、函数调用、MPI 以及硬件使用情况、PAPI 事件、功耗等信息,并且通过自定义的 cfg 配置文件,可以灵活地对输出形式进行配置,可以将某几项指标综合输出展示^[16],如将指令计数和周期计数进行处理,可以输出 IPC(instructions per cycle)数据。所得到的每种事件的数据都可以导出为 csv 格式由其他程序进行处理。

BSC Tools 作为一款开源工具,在欧洲的巴塞罗那超算中心的多个集群长期开发使用,具有很多优点。(1)安装简单,配置灵活。安装通过一条指令就可以配置好依赖环境,并且 Paraver 有 win 和 mac 系统下的安装包,可以将跟踪数据导出后处理,也可以直接在 Linux 下使用官方已经编译好的压缩包,直接解压使

用。Extrae 通过在 xml 文件中简单修改参数就可以进行配置,其集成的库劫持和二进制插桩的功能可以在没有源文件的情况下,对目标应用进行性能和功耗的跟踪分析,在集群实际使用中更为灵活。并且 Paraver 中可以自由定制多种 cfg 文件,对采集的数据进行定制化输出;(2)支持多种平台,除了 x86 外,BSC Tools 支持含 arm 和 power,并且由于其对 OpenCL 的支持,理论上对 CPU+xeon phi、CPU+FPGA、CPU+DSP 都可以进行跟踪分析。但是由于配置过于灵活,BSC Tools 并没有提供详细的使用引导,其中 Paraver 的用户说明长期未更新,已经失效。

2.5 对比分析

通过对上述工具的功能和特点进行对比,见表 1,可以看出对于并行程序,作为硬件厂商自产的性能分析工具,Intel 和 Nvidia 的工具套件对各自的产品针对性更强,功能和可操作性也更胜一筹,但是也由于其闭源,不可能移植到其他平台上,也就给自主的硬件平台或者强调自主可控的应用场景带来了困难。

表 1 几种性能功耗测量工具的对比

指标	基于 Score-P 分析工具	Intel Parallel Studio	Nvidia 工具	BSC Tools
性能检测	Score-P (代码插桩和采样)	Intel MPI 的 - trace 调用 (代码插桩、MPI 采样)	CUPTI (代码插桩和采样)	Paraver (代码插桩)
功耗检测	PAPI、Usage	无	NVML	PAPI
追踪视图 (trace)	除 Scalasca 不支持时间线视图,Score-P 提供了在线实时追踪的接口,没有提供专门的 CUDA 视图	MPI 函数的可视化信息和消息详情,支持时间线视图,不提供在线追踪	支持时间线视图,不支持在线追踪	查看所有测试信息视图,并提供了在线实时追踪功能
轮廓视图 (profile)	可以追踪完整调用堆栈的函数	Vtune 可以提供函数视图	可以显示 GPU 和 CPU 的完整调用堆栈的函数	可以自定义任何测量结果的视图
测量开销	和设定有关 5% ~ 40%	2%	5%	通常 5%
易用性	Score-P 安装使用困难, Vampir 使用简单, TAU 和 Scalasca 使用复杂	易于使用	易于使用	Extrae 和 Paraver 安装容易,进阶使用十分困难
功能	Score-P 功能强大,可视化工具各有特长,主要针对 MPI 分析	针对 Intel MPI 的调整优化给出的建议具体详实,对于多线程优化, Vtune 提供了完整的优化建议,不显示功耗数据	功能强大, GPU 相关的 API 提供最为全面,可以生成 GPU 相关参数调整报告,指导性强,可以提供 GPU 的功耗视图	提供多种 cfg 配置模版,简单分析实用性强,并且通过定制可实现多种分析功能,不提供分析报告,可以显示功耗视图
是否开源	Score-P、Scalasca、TAU 开源 (BSD)、Vampir(收费)	商业软件	商业软件	开源 (LGPL)
支持编程框架	OpenACC、OpenCL、CUDA、OpenMP、MPI	OpenMP、MPI	CUDA、OpenMP、OpenACC	OpenCL、CUDA、MPI、OpenMP
已支持平台	x86、Power CPU	Intel CPU	Nvidia GPU+x86 CPU	x86 CPU、Nvidia GPU、Xeon Phi、ARM

Score-P 对编程框架的支持最为广泛,提供了对 CUDA、OpenACC 和 OpenCL 的支持。但是其安装和配置较为复杂,并且需要对源程序进行修改,不能对已

经生成的可执行文件进行性能和功耗的分析。在基于 Score-P 的分析工具中, Vampir 使用最为方便,但是只有商业授权才可以使用完整功能。TAU 的功能最强,

提供了丰富的说明文档,但是安装使用较为复杂。Scalasca 提供了中文说明文档,但是本身缺乏时间线视图的输出。

BSC Tools 的使用很灵活。Extrae 可以支持 arm 平台,并且也提供对 python 程序的插桩跟踪。官方提供了各种编程框架的跟踪样例,可以直接拷贝到需要测试的场景,经过简单修改即可使用。在 Paraver 中预置了大量的 cfg 配置文件,在进行基本测试上非常容易上手。但是如果进阶使用,自己定制跟踪和分析时,配置较为繁琐,也缺乏说明文件。

综上所述,几种工具各有优缺,在实际应用中,可以相互结合使用,对于 Intel 和 Nvidia 平台无疑是官方软件的支持最为强大,可以给出最全面的分析报告。对于定制化集群平台,灵活配置 Score-P 和 BSC Tools 可以为优化工作带来很多便利。由于许多实验室集群中并没有专用的成套工具,因此使用开源工具也是一个很好的办法。BSC Tools、Score-P 都能在 Github 上找到完整的源代码,可以对不同的硬件平台进行分析和移植,也可以作为基础进行二次开发,使其更适用于特定应用场景。

3 结束语

由于高性能并行程序在性能和功耗上有更高的需求,因此要对其进行相关的检测分析。由于传统较多使用手动插桩输出的办法已经不太适宜当前的高性能并行软件调优工作,因此该文以并行程序的性能和功耗检测为主线,对目前国外常用的性能和功耗检测工具进行了对比介绍,讨论了其优缺点,以便后续使用,并希望能激发相关研究人员使用这类工具来进行软硬件优化的兴趣。

参考文献:

- [1] ABE Y, SASAKI H, KATO S, et al. Power and performance characterization and modeling of gpu-accelerated systems [C]//2014 IEEE 28th international parallel and distributed processing symposium. Phoenix, AZ, USA; IEEE, 2014: 113-122.
- [2] TRIVEDI P, TRIPATHI R P. Design & analysis of 16 bit RISC processor using low power pipelining [C]//International conference on computing, communication & automation. Greater Noida, India; IEEE, 2015: 1294-1297.
- [3] CALORE E, GABBANA A, SCHIFANO S F, et al. Evaluation of DVFS techniques on modern HPC processors and accelerators for energy-aware applications [J]. Concurrency and Computation: Practice and Experience, 2017, 29(12): 1-13.
- [4] CHEN J, TAN L, WU P, et al. GreenLA: green linear algebra software for GPU-accelerated heterogeneous computing [C]//SC16: Proceedings of the international conference for high performance computing, networking, storage and analysis. Salt Lake City, UT, USA; IEEE, 2016: 667-677.
- [5] 顾蓓蓓, 武虹, 迟学斌, 等. 国内外高性能计算应用发展概况分析 [J]. 科研信息化技术与应用, 2014(4): 82-91.
- [6] ILSCHKE T, SCHUCHART J, SCHÖNE R, et al. Combining instrumentation and sampling for trace-based application performance analysis [M]//Tools for high performance computing 2014. Stuttgart, Germany; Springer, 2015: 123-136.
- [7] 陈永然. 面向高性能计算的性能评价模型技术研究 [D]. 长沙: 国防科学技术大学, 2007.
- [8] ALMEIDA F, ARTEAGA J, BLANCO V, et al. Energy measurement tools for ultrascale computing: a survey [J]. Supercomputing Frontiers and Innovations, 2015, 2(2): 64-76.
- [9] GE R, FENG X, SONG S, et al. PowerPack: energy profiling and analysis of high-performance systems and applications [J]. IEEE Transactions on Parallel & Distributed Systems, 2010, 21(5): 658-671.
- [10] BELLOSA F. The benefits of event-driven energy accounting in power-sensitive systems [C]//Proceedings of the 9th workshop on ACM SIGOPS European workshop: beyond the PC: new challenges for the operating system. Kolding, Denmark; ACM, 2000: 37-42.
- [11] 刘辛, 沈立, 苏博, 等. 多核处理器的功耗估算模型 [J]. 软件学报, 2015, 26(7): 1840-1852.
- [12] DONGARRA J, LONDON K, MOORE S, et al. Using PAPI for hardware performance monitoring on Linux systems [C]//Conference on Linux clusters: the HPC Revolution. Urbana; Linux Clusters Institute, 2001: 25-27.
- [13] KNÜPFER A, RÖSSEL C, AN MEY D, et al. Score-p: a joint performance measurement run-time infrastructure for periscope, scalasca TAU, and vampir [M]//Tools for high performance computing 2011. ZIH, Dresden; Springer, 2012: 79-91.
- [14] ESCHWEILER D, WAGNER M, GEIMER M, et al. Open trace format 2: the next generation of scalable trace formats and support libraries [C]//International conference on parallel computing. Taipei, China; ACM, 2011: 481-490.
- [15] MAROWKA A. On performance analysis of a multithreaded application parallelized by different programming models using intel vtune [C]//International conference on parallel computing technologies. Berlin; Springer, 2011: 317-331.
- [16] MANTOVANI F, CALORE E. Performance and power analysis of HPC workloads on heterogeneous multi-node clusters [J]. Journal of Low Power Electronics and Applications, 2018, 8(2): 13-27.