高性能计算系统性能评测关键问题探讨①

Study on performance evaluation of HPC system

肖文名　李永生　陈晓宇　(广东省气象信息中心 广州 510080)
宗　翔　(国家气象信息中心 北京 100081)

摘　要： 本文介绍了 HPC(高性能计算)系统评测理论和评测方法。同时介绍了我们在引进 HPC 系统时设计的评测方案以及在招投标过程中所综合考虑的各类技术要求,评测结果表明我们设计的方案能够很好的满足气象业务需求。同时对评测结果进行了理论分析,分析结论对相关业务需求的 HPC 系统建设具有一定的实际意义。

关键词： HPC 系统　性能评测　加速比

HPC 经过几十年的发展,目前主流的 HPC 是可扩展的并行机,包括共享存储的对称多处理机(SMP)和分布存储的大规模并行机(MPP)与工作站机群(COW)。面对众多的 HPC 系统,如何评测和选择适合气象数值天气预报业务应用的 HPC 系统是我们要必须解决的问题,为此必须首先了解 HPC 系统的性能指标和 HPC 系统的评测理论和方法。本文在探讨 HPC 系统的性能指标体系和 HPC 系统的评测理论和方法的基础上,介绍了我们用于数值天气预报使用的 HPC 系统建设的实践经验,并对评测结果进行了理论分析。理论分析和实际运行效果表明我们设计的评测方案符合气象数值天气预报业务的实际需求。

1　HPC 系统性能评测指标

在评测 HPC 系统性能时,用户根据不同的需求经常提出不同的评测标准,以下描述的 6 种性能指标是用户经常提到的:执行时间、速度、吞吐率、利用率、成本有效性以及性能成本比。对于许多实际应用,有的用户可能明确要求达到某一处理速度,但有的用户对速度没有明确的要求,而明确要求更成本有效的运行其应用程序,因此根据特定需求合理选择评测指标是客观评价 HPC 系统性能的关键。

1.1　工作负载和速度指标

与工作负载和速度相关的指标是执行时间、执行的指令数以及执行的浮点操作数。对于运行在具体计算机系统上的给定程序,工作负载的衡量指标是执行该程序所需的总时间对任意具体程序工作负载的衡量指标是指令数,另外对于数值计算占主要地位的工程计算和信号处理应用来讲,负载的衡量指标是浮点操作数。表 1 总结了这 3 个性能指标。

表 1　工作负载和速度指标

工作负载类型	工作负载单位	速度单位
执行时间	秒(s)、CPU 时钟	应用/每秒
指令数	百万条指令或十亿条指令	MIPS 或 BIPS
浮点操作数	Flop,Mflop、Gflop、Tflop	Mflop/s

1.2　并行性能指标

加速比是并行处理的主要评测指标[1],并行系统加速比是指给定程序在单处理机上的执行时间与在多个同样处理器组成的并行系统上的执行时间之比。加速比可以分为问题受限(固定问题规模)扩展加速比,时间受限(固定时间)扩展加速比,内存受限(固定内存容量)扩展加速比。

固定问题规模是指在计算量相同(即相同问题)的前提下,通过增加处理器数来减少计算时间。此时

① 基金项目:国家科技基础条件平台建设项目(2005DKA64000)专题项目(专题任务书编号:KJB－NWG－2007－08)

的加速比满足 Amdahl 定律：

$$s=\frac{w_s+w_p}{w_s+\frac{w_p}{n}} \quad (1)$$

其中公式(1)中 w_s,w_p 分别表示问题规模 w 的串行分量(问题中不能并行化的那一部分)和并行分量，n 表示处理器数量；取 $f=\frac{w_s}{w}$ 表示问题中串行分量所占的比重，则加速比 $S=n/(1+f(n-1))$。这表明：串行比例 f 增加，加速比 S 就减小，当 $n--->\infty$ 时，$S=1/f$，即通过增加处理器个数不能提高加速比，此时，串行分量成了程序的瓶颈。此结论在历史上曾对并行系统的发展起到了悲观的作用。

固定时间问题是指在完成程序所需时间固定的前提下，通过增加处理器数来求解更大规模的问题。此时计算负载是可以改变的，我们在增加处理器的同时增加问题规模。这时加速比满足 Gustafson 定律：

$$s=\frac{w_s+nw_p}{w_s+n\frac{w_p}{n}} \quad (2)$$

其中公式(2)中问题规模在并行后变为了 w_s+nw_p，即增加了可并行部分的问题规模。取 $f=\frac{w_s}{w}$，则 $s=(1-f)n+f$，它表明：加速比与处理器个数成线性关系，它意味着随着处理器数的增加，加速比几乎与处理器个数成比例的线性增加，串行分量比例 f 不再是程序的瓶颈，这对并行系统的发展是个非常乐观的结论。

固定内存容量是指在内存空间受限的前提下求解尽可能大的问题。其基本思想是假若有足够的存储容量(通过增加处理器数)，并且规模可扩展的问题满足 Gustafson 定律规定的时间要求，那么就有可能进一步增大问题规模求得更好更精确的解。此时加速比满足 Xian－HeSun 和 Lionel Ni 定律：

$$s=\frac{(f+(1-f)G(n))}{(f+(1-f)\frac{G(n)}{n})} \quad (3)$$

式(3)中 $f=\frac{w_s}{w}$ 表示问题中串行分量所占的比重。它是 Amdahl 定律和 Gustafson 定律的更一般表示形式，其中 G(n) 反映存储容量增加 n 倍是工作负载的增加量。当 G(n)=1 时，就是 Amdahl 定律，当 G(n)=n 时，就是 Gustafson 定律。当 G(n)>n 时，它相应于计算负载比存储要求增加得快，此时 Sun 和 Ni 加速均比 Amdahl 加速和 Gustafson 加速为快。

2　HPC 系统性能评测方法

为了比较不同计算机的性能，现在有许多测试方法和基准测试程序。可以分为微观测试和宏观测试两大类，宏观测试把被测试的系统做一个整体看待，通过比较不同系统上运行同一组程序得到的性能数据来评价系统的相对性能；微观测试则用来评价系统在某一方面的特定表现[2]，如处理器速度、内存速度、I/O 速度、操作系统、网络性能等。目前国际上流行的基准测试软件包大约有 100 多种。因此在系统性能评测过程中准确确定评测目标以及评测方法是非常关键的。

2.1　性能评测目标

高性能计算机业务系统建设应该与具体的业务需求相联系，合理确定系统的评测目标，这是系统评测的关键点之一。一般来讲系统的评测目标包括如下几类：

(1) 系统的配置规模：目标机所需实际资源(CPU、内存、通信网络以及外存储器等)，以寻求系统配置的合理性。

(2) 系统的总体性能：一般通过比较不同系统上运行同一组程序得到的性能数据来评价系统的相对性能。

(3) 系统的单项性能：MPI 通信带宽/延迟、内存带宽、数据库性能、并行性能等。

(4) 系统的可管理性：通常认为包括可用性和可维护性。比如 Checkpoint/Restart 功能、多作业同时运行的吞吐能力等。

(5) 系统的可扩展性：有几种机器可实现目标系统，并展示出好的系统性能。这样可以满足不断发展的业务需求。

当然随着技术的不断发展应用的不断深入，更多的性能评测目标将会不断的被提出。

2.2　性能评测方法

系统性能测试方法主要包括基准测试和实际工作负载驱动测试两类。

基准测试是指利用业界开发的多种基准测试程序测试 HPC 统的性能指标，这些测试指标通常都是对计算机某一方面进行测试，如 CPU 的速度、存储器速度、

I/O 速度、操作系统性能等。而不能全面地说明系统的整体性能。目前常见的基准测试程序具体包括[3]：

(1) Linpack Benchmark：Linpack 采用矩阵乘算法来测试系统实际浮点峰值计算能力，Top500 就是按此速度排名。Linpack 测试的主要指标 Rmax(持续最大速度)、Rpeak(系统的峰值速度)、Nmax(达到 Rmax 时的问题规模)、N1/2(达到一半 Rmax 时的问题规模)。

(2) SPEC 基准程序序列：其中 CPU2000 测试的是单 CPU 性能及作业吞吐能力；SPEC OMP2001 的结果说明了共享内存的计算机使用共享内存并行模式(OpenMP)的并行效率和加速比。

(3) STREAM 主要对系统的数据访问能力(带宽和延迟)进行定量的评价。

(4) NPB：NPB 是由 NAS 开发的并行基准测试程序，NPB 由 5 个核心(EP、MG、CG、FT 和 IS)和 3 个模式应用(LU、SP 和 BT)程序组构成。是目前使用最广泛的并行基准测试程序。

(5) STAP：时 - 空自适应处理(Space - Time Adaptive Processing)基准程序组是一套实时雷达信号处理基准程序。

(6) PMB 是由 Pallas 研究开发的一套综合的 MPI 基准测试程序集，它主要测试，在 MPI 层上点对点消息传递、全局数据传送、单边通讯等性能。包含 PingPong、PingPing、Sendrecv、Exchange、Allreduce、Reduce、Reduce_scatter、Allgather、Allgatherv、Alltoall、Bcast 和 Barrier 12 个测试程序。另外还包括 lmbench、TPC、Iometer 等。

实际工作负载驱动测试是指采用用户自己的应用程序进行实际的测试[4]，或找到相似应用的商业软件的标准测试结果。这样的评价会更有针对性，更具实用价值。尤其是使用用户业务程序测试对用户最合适，因为用户业务程序测试结果很好的系统，用标准测试程序结果也许并不理想。

3 评测方案设计与性能分析

通过前面的介绍，我们知道不同的并行计算机系统之间差别很大，从处理器的类型到内存的分布方式，从处理器互联方式到操作系统，从编程模式到物理体积等等，各个方面都极不相同，价格也相差很大，我们应该如何设计测试方案，通过测试来选择最适合自己业务的 HPC 系统呢？我们在建设 HPC 系统过程中主要采用实际业务程序测试和基准测试相结合的方法。

3.1 测试方案

我们在 HPC 系统引进过程中，先后详细了解了 IBM 等多个厂家的高性能计算机产品，同时邀请各厂家到我局详细介绍各自产品的性能。我们在充分调研的基础上设计了测试方案。在方案设计的过程中，我们在实际业务程序测试的基础上综合考虑其他基准测试。测试的原则是：我们设计测试方案，提供实际业务软件和基准测试软件，厂家根据我们的测试要求自行确定计算机系统的软硬件配置和进行相关测试，要求 40 天内完成测试，按要求提供测试分析，并且要求厂家提供测试结果报告作为承诺依据，以保证测试的可重复性，中标后用户自己测试确认。该测试方案由 6 项测试，他们是：

TEST1：30 分钟内完成业务程序计算测试所需机器规规模，并且提供该规模下的具体价格。

TEST2：600 万投资规模下，完成该业务程序计算测试所需时间和机器规模。

TEST3：1000 万投资规模下，完成该业务程序计算测试所需时间和机器规模。

TEST4：Checkpoint/Restart 功能测试。TEST5：加速比测试。

TEST6：PMB 测试。

具体测试要求如下：

(1) 必须是 unix 或 Linux 环境和 64 位精度的机器。

(2) 所有测试相关脚本应为批作业的方式运行，不能用交互方式。

(3) 要求返回详细的电子输出结果、测试报告表和分析报告书。

我们设计的方案特别要求完成 Checkpoint/Restart 功能测试。该功能能够保证一个作业运行被中断后可以从离中断点最近的 checkpoint 处重新启动；或者运行业务程序的机器发生故障时，可以在另一台满足运行要求(内存，处理器数量)的机器上从离故障点最近的 checkpoint 处重新启动。

3.2 评测结果分析

多家高性能计算公司(涉及商业保密原因，分析中

不具体指明具体公司名称)参与了这次测试,并且提交了符合要求的测试报告和分析报告。我们在对测试方法正确性分 析、测试结果真实性分析的基础上,综合各方面因素对系统进行了综合评估,同时根据方案进行了实际验证测试,从实际验证测试的结果分析中总结了以下几点:

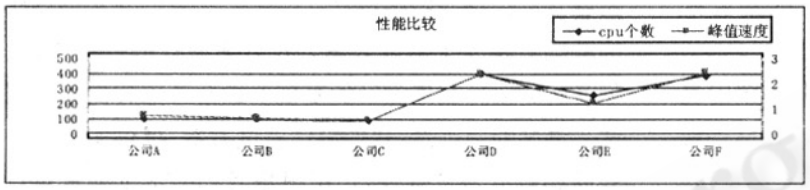

图 1　CPU 个数与峰值速度的关系

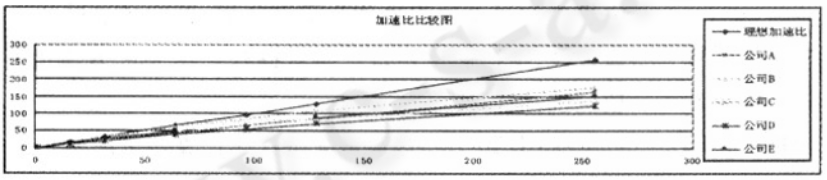

图 2　加速比比较图

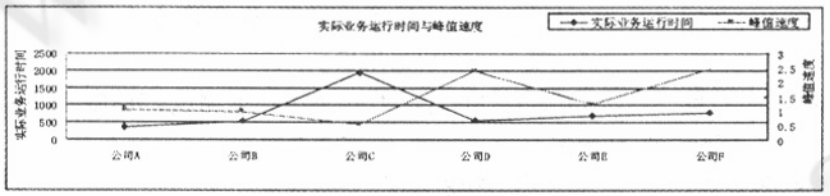

图 3　实际业务运行时间和峰值速度比较图

(1) 目前多数高性能计算机大多采用 Cluster 体系结构,Cluster 体系结构目前已经处于主流地位。同时在系统性能高低、规模大小上,各大高性能计算机厂商一般都做到随应用的要求而变化。

(2) 在运算速度指标方面:图 1 表明无论是 Linpack 指标还是具体应用测试的墙钟时间指标,各高性能计算机产品基本都可以通过扩展硬件规模达到要求。这也从实践上进一步证明作为高性能计算机排行榜的 Linpack 指标只反映了高性能计算机的一个侧面,只能用作一个参考。

(3)在高性能计算机并行性能方面,各高性能计算机产品之间的差距存在一定的差距,因此加速比性能指标仍然是性能评价的重要指标。如图 2 所示。

(4) 本次测试中,一些产品不能很好的的完成 Checkpoint/Restart 功能测试。另外部分 HPC 产品还暴露出作业调度性能不理想、软件成熟度差等缺点。因此随着高性能计算机的不断发展,系统的好用性、可靠性、交互性和易管理性逐渐成为衡量并行计算机的重要指标。

(5)在实际的测试过程中我们发现尽管有的厂家可以通过增加硬件规模达到很高的理论峰值速度,但是很高的理论峰值速度在我们的实际业务程序运行过程中并没有得到很好的体现,如图 3 所示公司 A 的 HPC 系统运行我们提供的实际业务程序所需的墙钟时间是最少的。但是它的理论值速度确不是最高的。相反公司 D 的理论峰值速度很高,但是运行实际业务程序所需的时间确很长。

4　结束语

实践经验告诉我们,高性能计算的性能指标用实际业务程序测试是最关键、最有说服力的,同时尽快建立符合气象数值天气预报业务要求的高性能计算机评测体系也至关重要。

参考文献

1　王丽,实时集群系统设计与性能分析[J],计算机工程与应用,2007,43(18):120~123.

2　殷新春、谢立等,一种选择最佳处理器个数的策略[J],计算机应用,2003,4(23):14~15.

3　洪文董、田浩,TFLOPS 级 HPC 应用性能测试方案设计[J],计算机工程与应用,2004,34:57~67.

4　Poly Wonopoulos C D, Banerjee U。Processor allocation for horizontal and vertical parallelism and related speedup bounds. IEEE Trans on Comp,1987,36(4):410~420.