面向评估的高产出率计算指标量化分析*)

杨沙洲　杜云飞　杨学军

(国防科技大学计算机学院　长沙 410073)

摘　要　为了解决当前传统的高性能计算片面强调性能而忽视系统各方面的平衡性所导致的问题，研究人员提出了"高产出率计算"的概念，并为此展开了高产出率计算评估体系的研究。本文从当前高产出率计算评估体系研究现状出发，从高产出率计算对传统"性价比"评估概念的扩展入手，提出了以"指标量化"和"统一量纲"为中心的评估高产出率计算效果的方案，并针对计算能力、鲁棒性、能耗和易用性四个要素给出了一种量化的评估办法。

关键词　高产出率计算，评估指标，量化，量纲

Quantified Analysis of High Productivity Computing Metrics Oriented to Evaluation

YANG Sha-Zhou　DU Yun-Fei　YANG Xue-Jun

(Institute of Computer, National University of Defense Technology, Changsha 410073)

Abstract　In traditional high performance computing area, the system's performance gets the greatest focus. The balance among all aspects of the whole system is neglected. The idea for building computing system causes problems. An approach for evaluation of high productivity computing (alias HPC) based on quantified metrics and unified dimensions is provided, starting with the status quo of the research on evaluation schema of HPC, and the extension to the evaluation concept of the traditional 'performance-cost ratio' by HPC. A quantified evaluation method is proposed on 4 main factors of HPC: computing abilities, robustness, energy consumption and the 'easy-to-use' abilities.

Keywords　Hing produetivity computing, Evaluation metrics, Quantification, Dimension

1　引言

传统的高端计算业界，计算机(或称计算系统)是处于市场的优势地位的，开发者只需要关心计算性能的提高，就能够获得用户的认同。但随着计算机的应用迅速地扩展到社会生产的各个领域，用户对计算系统能力的要求也越来越具体，越来越苛刻，这是IT业发展成熟的必然趋势。用户需求的细化，反过来促进计算机界对计算系统性能评估指标的反思，从而逐渐产生了"高产出率计算"的概念[1]，追求性能、可编程性、可移植性、鲁棒性和系统生成维护成本的平衡。该概念就是希望准确反映用户对计算能力的精确定义，从而指导业界研发朝着这一方向努力。

为了让"高产出率计算"概念走入实用，需要一系列具备良好可操作性的评估手段，"高性能"、"高可编程能力"、"高可移植能力"和"高稳定可靠性"四个概念性评估指标高度概括了高产出率计算的特点，但在评估的可操作性上仍有明显的不足。很多研究者都对此进行了和进行着深入的研究。Kepner等人对产出率评估的流程进行了分析，围绕执行时间和开发时间两个最重要的指标提出了一种产出率评估框架[2]。Sterling提出了一个概念化的公式 $\Psi_L=\frac{R_L}{C_L\times T_L}$ [3]，其中 Ψ_L 表示计算系统的产出率，R_L 指系统产出的加权和，C_L 表示为获得 R_L 而引入的耗费(包括系统的部署、运行及软件编制等开销)，T_L 则指的是系统获得上述产出所花费的时间。Kennedy等人分析了软件领域中开发成本与执行效率的折衷问题[4]，而Faulk等人则从软件工程角度来看待产出率的实质，对"产出"和"耗费"的量化定义进行了讨论。

本文对高端计算领域中的高产出率计算和传统高性能计算的评估进行了对比分析，从"性价比"入手探讨了未来高端计算环境的评估体系建设问题。

2　评估指标的量化

一个具备现实可操作的评估体系首先应是一个量化的指标体系，即评估者能够根据系统可获取的数据资料，在一定的算法框架内求出系统"产出率"的可比值。正如传统高性能计算通常所采用的"性价比"评估体系，机器执行的操作数和执行这些操作所花费的时间的比值，即可被接受为系统计算能力的评价指标之一。而可比的计算能力与购买计算系统的开销的比值则成为更高层次的"性价比"计算方式：

$$Ratio=\frac{Performance}{Cost}$$

其中Performance在量化过程中通常以系统的峰值速度来衡量，例如通常用MFLOPS描述某大型机系统的速度，表示该系统每秒能执行多少条浮点指令。很多系统分析师也用特定程序(通常是一组Benchmark)的程序执行时间作为评价计算系统性能的标准。这里的程序执行时间通常定义为程序启动至程序运行完成的墙钟时间。

而Cost通常指的是软硬件系统的购买花费。"性价比"指标体现的是"用最少的钱获得最高性能的计算机"的思想。

显然，这样的量化方法是工业时代的做法，把计算系统当

*)863软件专项-服务器操作系统(项目号 2002AA1Z2101)。杨沙洲　博士生，研究方向：操作系统；杜云飞　博士生，研究方向操作系统；杨学军　教授、博士生导师，研究方向计算机软件、并行技术。

作普通的商品来买卖，评估所针对的是系统本身，而不是针对终端用户或所要解决的终极问题。而实际上，计算系统作为一种复杂的“工具”，仅仅是解决方案的最基本的一个部分，它为最终用户所带来的经济效益并不能简单地用计算系统本身的“性能”来反映，而是与该计算系统的“使用过程”有密切的关系。一个“合理”的系统的评估方法应该反映的是用最少的投资最大限度满足最终用户的需求。在经济学上，这种“投入产出比”就是“产出率(Productivity)。此时，对一个计算系统的评价，将从用户需求的各个角度出发，性能以及性价比的衡量被扩展为对计算系统的产出率的衡量，包括计算能力、系统可靠性和稳定性、能耗控制能力以及用户易用程度四个方面。所有这四个方面都牵涉到计算的全过程，而不仅仅是软硬件系统的交易过程，而这正是“高产出率计算”能力评估的复杂之处。

将产出率定义为 Capacity(计算能力)、Robustness(鲁棒性，即可靠稳定运转能力)、Energy(能耗)和 easy-to-Use(易用性)四元组，即

$\Psi=\langle C,R,E,U\rangle$

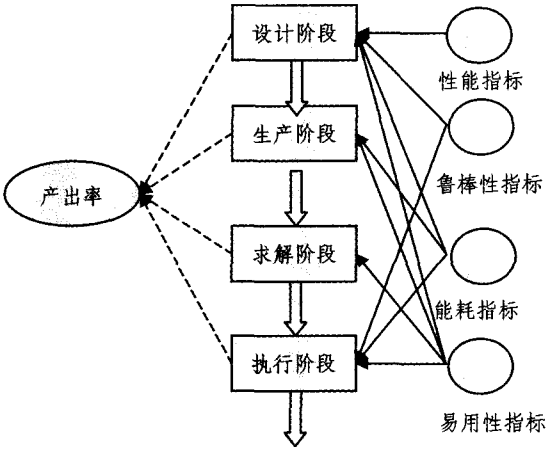

图 1　高产出率评估指标对系统的影响

3　量化实例

如上所述，评估产出率的关键之一就是评估指标的量化，而指标量化则有多种途径。本文将“性能”、“编程能力”、“稳定运行能力”、“移植能力”和维护成本等概念化指标用“性能”、“鲁棒性”、“能耗”和“易用性”四个可量化的指标体系表示，并给出了其中一种量化途径，以证明这一量化方案的可行性。

3.1　计算能力

计算能力 C 直接反应系统的性能因素，量化时可定义为如下二元组：

$C=\langle C_v,C_t\rangle$

其中：C_v 指峰值速度，定义为

$$C_v=\mathrm{MAX}\left(\frac{\Delta w}{\Delta t}\right)$$

表示单位时间内最大的执行能力，其中 Δt 表示某个单位时间段，Δw 表示 Δt 时间段内的工作量。该数值越大说明系统性能越好。

C_t 指的是单位工作量的平均执行时间，定义为

$$C_t=\frac{\sum_{i=1}^{k}(T^i_{end\_time}-T^i_{start\_time})}{\sum_{i=1}^{k}w_i}$$

其中 $T^i_{end\_time}$ 和 $T^i_{start\_time}$ 分别表示第 i 个工作的结束时间和启动时间，C_t 的分子部分表示所有任务所花时间之和，分母部分则是所有任务的操作数之和。该数值越小，说明系统性能越好。

在并行系统中，因为任务的并行作用，T 在绝对时间上有重叠，因此通常用系统所有任务最早的启动时间与最迟的完成时间之差来表示任务执行时间，即

$$C_t=\frac{T_{end\_time}-T_{start\_time}}{\sum_{i=1}^{k}w_i}$$

在完全并行、所有任务完全重叠执行的情况下，有

$$T_{end\_time}-T_{start\_time}=\frac{\sum_{i=1}^{k}(T^i_{end\_time}-T^i_{start\_time})}{n}$$

其中 n 为系统运算节点数，因此

$\dfrac{\sum_{i=1}^{k}(T^i_{end\_time}-T^i_{start\_time})}{n\sum_{i=1}^{k}w_i}$ 是并行系统平均执行时间 C_t 的最大值。

本质上 C_v 衡量系统的最大计算能力，而 C_t 所衡量的是系统平均的计算能力。如果系统保持峰值速度运行，则有

$C_v\cdot C_t=1$

通常情况下，并行系统的计算速度很少能保持在峰值，因此常将二者综合起来反映相对问题求解需求的系统价值。

3.2　鲁棒性

鲁棒性 R 既反映系统的能力，同时也反映系统的维护成本。在计量时同样采用复合概念，将其定义为如下二元组：

$R=\langle R_t,R_f\rangle$

此处，R_t 指系统平均稳定运行时间比：

$$R_t=\frac{\sum_{i=1}^{k}(T^i_{dnum\_time}-T^i_{restart\_time})}{t}$$

其中 $T^i_{restart\_time}$ 和 $T^i_{dnum\_time}$ 分别表示系统第 i 次失效时重启的时间和重启后首次失效的时间，k 表示计量时间段 t 内连续稳定运行的次数。R_t 越大，表示系统越稳定，无故障的理想情况下，$k=1$，R_t 得到最大值，值为 1。

R_f 是系统故障率：

$$R_f=\frac{k_t}{t}$$

其中 K_t 为 t 时间段内的失效次数。R_f 越大，说明系统稳定性越差。无故障的理想情况下，$R_f=0$。

R_t 和 R_f 本质上都是统计值，在无限长的时间段内做统计时才能准确反映系统稳定性和可靠性。两个指标虽然都可反映系统失效的影响，但却是两个不同的方面。在实践上，通常以历史采样值来代替无限长时间段的理论值。

3.3　能耗

能耗 E 与以上两个因素不同，它并不仅仅是计算系统的特性，更多表现为环境对计算系统提出的要求，同时反映系统的运行成本。量化时采用如下二元组：

$E=\langle E_p,E_w\rangle$

E_p 是峰值能耗：

$$E_p=\mathrm{MAX}\left(\frac{dE}{dt}\right)$$

用以衡量计算过程中单位时间内的最高能耗。通常用户会对该值设定上限，从而对系统提出了要求。该数值越小，说明系统能耗越低。

E_w 为系统平均能效，以单位能量所完成的工作量来衡量能量效率，即

$$E_w=\frac{\sum_{i=1}^{k}w_i}{\sum_{i=1}^{k}E_i}$$

此处 E_i 和 w_i 分别表示作业 i 的能耗和操作数。该数值越大，说明系统能效越高。

如果系统保持峰值功率运行，则有

$$E_p \cdot E_w \cdot t=W$$

W 为总的工作量。

假定系统也能保持峰值速度运行的话，则有

$$E_p \cdot E_w=\frac{W}{t}=C_v$$

3.4 易用性

用户的易用性是用户能够感知的系统特性之一，体现的是系统对环境的适应能力。但由于衡量易用性时人的感觉因素较多，因此度量上有一定的难度。这里也用三元组来定义：

$$U=\langle U_p, U_s, U_r\rangle$$

U_p 表示系统的编程能力，是对系统编程模型、编程工具表达能力、编程工具编码效率等因素的综合评价。本文将其针对某一特定的问题 i 简化为一个可量化的二元组：

$$U_p^i=\langle\frac{T^i_{problem\_to\_solution}}{W^i},\frac{T^i_{execute}}{W^i}\rangle$$

第一项表示获得对问题 i 的求解方案的过程的时间与 i 的问题规模的比值，以此来衡量编程工具的表达能力和编程模型的易用性，数值越小说明系统易用性越好。第二项表示求解方案的执行时间与问题规模的比值，以此来衡量编程工具的编码效率，数值越小易用性越好。

U_s 表示系统的可扩展能力，即系统对问题、环境的适应能力。它也包括两个方面：同一个问题，规模变化时系统的应对灵活性；系统对不同问题的针对性优化能力。用以下二元组来量化：

$$U_s^i=\langle\frac{\Delta T^i_{execute}}{\Delta W^i},\frac{T^i_{problem\_to\_solution}+T^i_{execute}}{T^{base}_{problem\_to\_solution}+T^{base}_{execute}}\rangle$$

第一项表示对于同一问题 i，在规模变化时执行时间上的变化，例如同样是数据统计，百兆级的数据量下执行时间为 $t1$，千兆级数据量下执行时间为 $t2$，则 ΔW^i 表示从百兆变为千兆的问题规模上的变化，而 $\Delta T^i_{execute}$ 则可以用 $t2-t1$（或相反）来表示。显然，该数值越小，说明系统的可扩展性越好。

第二项表示问题 i 的求解时间与基准问题 *base* 的求解时间之比。本文用某一个标准测试组的获得求解方案的时间和方案执行时间作为任何应用问题的比较基点，来说明系统对于处理新问题的灵活性。该数值越小，说明系统灵活性越高、对新问题的适应能力越强。

U_r 表示系统在问题求解过程中的用户开销，包括对用户空间、时间、人力、物力的占用程度，以此来衡量系统给用户带来的运行成本和维护成本。此项是与用户类型密切相关的，量化方法也和用户类型、问题类型密切相关。

4 产出率指标的统一量纲

就像传统“性价比”指标中“性能”和“价格”优化目标相矛盾一样，高产出率评价指标的各个因素之间也存在制衡。对系统的调优，实质上是针对具体的应用寻求各个因素之间所获得的综合值的最大化，即产出率的最大化。而从以上分析可知，各个产出率指标的量纲是不同的，峰值速度是“操作数/单位时间”，平均执行时间是“单位时间”，故障率是“次数/单位时间”，峰值能耗是“单位能量/单位时间”，平均能效是“操作数/单位能量”，问题求解时间是“单位时间/单位工作量”，而用户开销的量纲既包括时间，也包括空间，甚至还包括费用。

为了从理论上对产出率影响指标的均衡进行分析，有必要对它们的量纲进行统一，从而方便构造理论评估模型和最优化算法计算。在上述所有量纲中，“时间”是使用比较频繁的指标。同时，其他各个指标在固定某一较为恒定的参量之后，可以相应转化为“时间单位”。例如，能量 E 可以用系统较为固定的输入功率 P 作为媒介转化为时间（$T=E/P$），操作数则可以借助相对恒定的主频转化为时间。

另一种重要的参考量纲则是“金钱”，所有指标均可以转化为投入和产出，在预设的价格体系内，包括操作数、运行时间、能耗、故障损失、研发时间等各种参数都可以转化为人工、物质、管理等成本及“产品”的产值。

小结 DARPA 的高产出率计算概念已经提出有 4 年，针对高产出率系统的研究已经在 IBM、SUN、Cray 等三家公司及多家合作大学及国家实验室中展开了广泛而深入的研究。但对于如何评价一个计算系统的“产出能力”，则直到目前仍然没有获得权威的认识。SUN 公司建立了一套面向高性能计算的测试组 PBB(Purpose-based Benchmarks)[6]，并拟将其用于评估未来高产出率计算系统的能力。但就目前而言，PBB 仍然未能概括高产出率计算的全部涵义。

本文从“性价比”的分析入手，指出“量化”及此基础上的“量纲统一”是进行产出率评估的关键因素，并提出了一组量化和量纲统一的方案。通过分析获知，这种建立评估体系的思路非常符合高产出率计算跨越计算流程各个阶段、涵盖多个计算相关的领域的特点，比较适合于建立操作性较强的高产出率评估体系。

参 考 文 献

1 孟丹，张志宏，陈明宇. 高生产率计算系统. 计算机研究与发展，2005，42(4)：563～569

2 Kepner J. HPC Productivity: An Overarching View. International Journal of High Performance Computing Applications, 2004, 18(4)

3 Sterling T. Productivity Metrics and Models for High Performance Computing. International Journal of High Performance Computing Applications, 2004, 18(4): 433～440

4 Kennedy K, Koelbel C, Schreiber R. Defining and Measuring the Productivity of Programming Languages. International Journal of High Performance Computing Applications, 2004, 18(4): 441～448

5 Faulk S, Gustafson J, Johnson P, et al. Measuring HPC Productivity. International Journal of High Performance Computing Applications, 2004, 18(4): 459～473

6 Gustafson J. Purpose-based Benchmarks. International Journal of High Performance Computing Applications, 2004, 18(4): 475～487