国家高性能计算环境资源准入和分级标准体系

白皮书

2018 年 6 月

目录

国家高性能计算环境聚合了国内优秀的高性能计算资源，面向用户提供高效、便捷的高性能计算服务。目前，国家高性能计算环境由十多家计算中心的计算资源整合而成，资源种类繁多、异构性强、地域分布广。为加强环境建设、提高服务质量，本白皮书提出一套可以量化网络服务水平和集群计算服务水平的规范，为环境资源准入和服务化模式运行提供基础支撑，支持和引导用户合理使用资源，形成全局统筹的资源布局。

本白皮书包括三部分：资源服务水平的评价、资源评价计算方法研究、资源分级和准入标准。

1、资源服务水平的评价

环境资源的软硬件条件是高性能计算环境对外提供服务的基础，针对高性能计算环境中的计算、存储和网络资源，本部分制定一整套指标体系，用于评测环境中现有的和今后要加入到环境中的各类资源，其目的在于评估资源的服务能力，为环境建设提供指导，为用户使用环境资源提供参考。

1.1 指标体系的设计原则

高性能计算环境中的主要资源是高性能计算集群。评价指标体系围绕高性能计算集群的性能、可用性、服务质量等方面进行设计，以表征高性能计算资源各方面特性及其相互的联系。指标体系包含的各个指标将构成具有内在结构的有机整体。设计原则如下：

(1) 系统性原则。能够全面描述环境中计算资源各方面的性质，并且各指标之间要有一定的逻辑关系，它们不但能从不同的侧面反映出计算资源的性能、服务等主要特征，而且还要反映计算资源各子系统之间的内在联系。各指标之间相互独立，又彼此联系，共同构成一个有机统一体。指标体系的构建具有层次性，自上而下，从宏观到微观层层深入，形成一个不可分割的评价体系。

(2) 典型性原则。面向资源准入评价的根本目标，务必确保评价指标具有一定的典型代表性，尽可能准确反映出资源及其子系统的综合特征，便于数据计算和保证评测结果的可靠性。评价指标体系的设置、权重在各指标问的分配及评价

标准的划分都应该与高性能计算环境的根本特征相适应。

(3) 科学性原则。指标体系的设计及评价指标的选择必须以科学性为原则，能客观真实地反映高性能计算环境和环境资源的特点和状况，能客观全面反映出各指标之间的真实关系。各评价指标应该具有典型代表性，不能过多过细，使指标过于繁琐，相互重叠；又不能过少过简，避免指标信息遗漏，出现错误、不真实现象，并且数据易获且计算方法简明易懂。

(4) 可比、可操作、可量化原则。指标选择上要特别注意在总体范围内的一致性。指标体系的构建是为评价资源服务的，指标选取的计算量度和计算方法必须一致统一，各指标尽量简单明了、微观性强、便于收集，各指标应该要具有很强的现实可操作性和可比性。而且，选择指标时也要考虑能否进行定量处理，以便于进行数学计算和分析。

(5) 动态性原则。应用新需求也会导致环境和用户关注的资源特性不断变换，技术发展会导致新资源呈现出新特性。因此，指标的选择要充分考虑到动态变化的特点，能根据时代和应用的需要不断完善、补充、优化指标体系，能够在系统性原则的基本要求下对新旧资源进行全面科学的评测 。

(6) 综合性原则。系统、科学地评价环境资源是最终目标，也是综合评价的重点。在各级指标体系的选择上，应全面考虑影响资源性能、可用性、可靠性、服务质量的诸多因素，并进行综合分析和评价。

1.2 资源评价的指标

依据评价指标的设计原则和评测高性能计算环境资源的要求，资源评价指标体系将由性能指标、可用性指标、可靠性指标、安全性指标、需求管理指标、技术支持和服务响应指标等组成，构成一套完整的体系，不仅反映高性能计算集群软硬件的性能、可用性、可靠性，也反映出环境资源服务质量的好坏和应用水平的高低，为高性能计算环境的建设和发展提供依据。

1.2.1 性能指标

性能指标用于评估高性能计算环境中高性能计算集群系统的计算性能。高性能计算集群系统（以下简称“系统”或“集群系统”）是高性能计算环境（以下

简称“环境”）中重要的计算资源，是环境计算能力的提供者。集群系统由计算子系统、存储子系统、网络子系统等组成，系统性能是各子系统协同工作的综合表现。

根据指标的来源和性质，进一步将性能指标分为基础性能指标和应用性能指标。决定系统性能的主要有计算（CPU+加速卡）、存储、网络及应用四个方面。因此，资源评价性能指标主要从这五个方面进行定义。

基础性能指标包括了计算性能测试（又分为 CPU 性能测试、加速卡性能测试）、内存性能测试、外存储或并行存储性能测试、网络性能测试等主要方面。系统的综合性能是各方面子性能的综合。一些测试指标，如实测峰值、应用程序测试速率等，是系统内 CPU、内存、网络甚至存储多部件或子系统协同工作的综合体现。

应用性能指标描述系统或节点通过执行基准测试程序或典型应用程序，经实测而得到的性能参数。测试对象为系统整体，通过应用性能测试可以评估系统对特定类型的应用的支持情况。

1.2.1.1 基础性能指标

基础性能指标体系包含四大项，分为计算、存储、网络等几个方面。表 1.1 针对系统进行相关指标的定义。这里，系统指一台完整的高性能计算集群系统，简称“系统”或“集群系统”。一般情况下，系统整体性能是系统中所有组成单元协同工作的结果：节点之间通过内部互联网络连接，运行于系统内的程序的进程之间按照控制逻辑通过网络进行通信交互，进程间有一定的同步或数据交换操作，同时进程需要存取存储在并行网络存储里面的文件，I/O 操作对应用程序的性能带来一定的影响。因此，基础性能测试是在上述复杂执行环境下，测试系统计算、内存、存储及网络的表现，以此评估其性能。

基础性能测试是对系统基本性能的评测。节点是集群系统的基本组成单元，是集群系统性能的基础；而系统是节点的集合，系统整体性能由其各组成子系统协同工作具体体现。因此将测试对象定为集群系统。

系统基础性能测试特指在复杂应用场景下，评测整个系统的计算子系统、并行存储子系统、网络子系统的协同工作表现，体现系统的整体性能。系统的基础

性能评测指标由表 1.1 给出。具体包括，计算性能评测：评估系统中所有节点的 CPU、加速卡提供的聚合计算力；内存性能评测：评估系统在多节点协同工作模式下的内存性能表现；并行存储性能评测：评估系统在多任务负载情形下并行存储子系统的性能；网络性能测试：评估系统在多任务工作模式下的网络通信性能。

表 1.1 基础性能指标

序号	指标名称					单位
计算性能测试指标						
1	CPU	整数计算性能		整数计算速率		
2		浮点计算性能	单精度	矩阵乘法速率		GFlops
3			双精度	稠密线性方程组求解速率		GFlops
4				稠密线性方程组求解计算效率 E_l^c		
5				矩阵乘法速率	EP 模式	GFlops
6				一维 DFT 速率	EP 模式	GFlops
7					GP 模式	GFlops
8				稀疏线性方程组求解速率		GFlops
9				稀疏线性方程组求解计算效率 E_g^c		
10	加速卡	浮点计算性能	单精度	矩阵乘法速率		GFlops
11			双精度	稠密线性方程组求解速率		GFlops
12				稠密线性方程组求解计算效率 E_l^s		
13				矩阵乘法速率	EP 模式	GFlops
14				一维 DFT 速率	EP 模式	GFlops
15					GP 模式	GFlops
16				稀疏线性方程组求解速率		GFlops
17				稀疏线性方程组求解计算效率 E_g^s		
18	CPU+加速卡混合	浮点计算性能	单精度	矩阵乘法速率		GFlops
19			双精度	稠密线性方程组求解速率		GFlops
20				稠密线性方程组求解计算效率 E_l^m		
21				矩阵乘法速率	EP 模式	GFlops
22				一维 DFT 速率	EP 模式	GFlops
23					GP 模式	GFlops
24				稀疏线性方程组求解速率		GFlops
25				稀疏线性方程组求解计算效率 E_g^m		
内存性能测试指标						
26	可持续内存带宽			复制模式		GB/s
27				加法模式		
28				数乘模式		
29				综合模式		

30	内存随机访问速率		EP 模式	GB/s
31			全局模式	GB/s
存储性能测试指标				
32	IOPS		随机读	ops/s
33			随机写	
34			顺序读	
35			顺序写	
36			随机存取	
37	存储带宽			MB/s
38	响应时间		最小响应时间	ms
39			平均响应时间	ms
40	元数据处理	文件操作	创建	the number of files/s
41			删除	the number of files/s
42			查询	the number of files/s
43		目录操作	创建	the number of files/s
44			删除	the number of files/s
45			查询	the number of files/s
46		树形目录	创建	the number of files/s
47			删除	the number of files/s
48			查询	the number of files/s
网络性能测试指标				
49	内部网络	PingPong 测试	延时	最小延时	ms
50				最大延时	ms
51				平均延时	ms
52			带宽	最小带宽	MB/s
53				最大带宽	MB/s
54				平均带宽	MB/s
55		环网测试	平均延时	自然序环	ms
56				随机序环	ms
57			平均带宽	自然序环	MB/s
58				随机序环	MB/s
59				最大延时	Ms
60				平均延时	ms
61		mpiGraph 测试	带宽	最小带宽	MB/s
62				最大带宽	MB/s
63		综合性能	总有效带宽		MB/s
64			每个处理器的平均有效带宽		MB/s
65		多处理器对间大数据传输速率			GB/s

66	外部网络	电信速率*	MB/s
67		联通速率*	MB/s
68		移动速率*	MB/s
69		教育网速率*	MB/s
70		科技网*	MB/s
71		其他网络*	MB/s

说明：

[1] 计算性能测试：计算性能测试用于度量集群系统的计算力。

节点的计算力来自于 CPU 和加速卡，包括整数计算性能和浮点计算性能。其中，整数运算主要针对 CPU，测量在 CPU 上运行整型数据运算基准程序的计算速率；浮点运算又分为单精度运算和双精度运算，并以双精度为主，测量在 CPU 或加速卡上运行浮点型数据运算基准程序的计算速率。节点计算性能评测分别对 CPU、加速卡、及 CPU+加速卡混合结构三种方式进行，体现了三种方式下节点能够提供的不同计算力。

系统的计算力主要来自于多节点的聚合计算能力，由协同工作的众多节点的 CPU 和加速卡提供。与节点类似，系统的计算性能也体现在对整数运算和浮点运算的支持上，其中整数运算主要测量 CPU 的整数协同计算能力，浮点运算可针对 CPU 或加速卡分别测量浮点数协同计算能力。浮点运算分为单精度运算和双精度运算，以双精度为主。为体现可提供的不同计算力，系统计算性能评测分别对 CPU、加速卡、及 CPU+加速卡混合结构三种方式进行。

国际上对 HPC 的基准测试多以 Linpack Benchmark（集群环境下采用 HPL Benchmark）为准，其测试值为集群双精度浮点计算最大峰值速度。但随着时代的发展，其局限性也日益突出。新的评测方法和评测标准被提出，如 SPEC、HPCG 等。这些测试的评测结果一定程度上能更准确地反映了高性能计算机系统的实际性能。因此，综合目前多种主流测试方法，制定多项测试内容和多个测试指标，从多个方面评价节点和系统的计算性能。

[2] 测试场景：根据测试对象的不同及测试程序运行在节点或集群系统上执行方式的不同，定义三种测试场景：

（1）Local, 此时只有一个节点或处理器工作，独立运行测试程序，简称“SP 模式”测试。

（2）Embarrassingly Parallel, 此时系统的所有节点工作，对应系统测试，特点是系统中的每一个处理器都运行测试程序的相关计算任务, 但是它们相互之间不进行显式通信。简称为“EP 模式”测试。

（3）Global, 此时系统的所有节点工作，与 EP 模式不同，不仅每一个系统中的处理器都要运行测试程序的相关计算任务, 而且它们相互之间还进行显式通信。简称为“GP 模式”测试。

EP 模式是一种简单的并行模式，进程间没有同步开销，但可能因为竞争通信或 IO 资源而产生少量开销，因此评估的是系统在多任务情形下无通信开销但存在资源竞争情况时的性能表现。GP 模式也是多任务并行，但执行计算任务的并行子进程间存在同步和通信开销，也可能存在资源竞争开销，因此更接近大多数实际应用的情形，GP 模式测试评估系统在复杂应用场景下的性能表现。

[3] 整数计算性能基准测试：在传统的高性能计算领域，单纯的整数运算并非主流应用。但随着 HPC 应用领域的拓展，HPC 系统的整数运算能力也受到关注，有其特定的应用场景，如离散时间处理、数据压缩、搜索等。因此整数计算性能测试将选择在这些领域有代表性的基准测试程序完成。整数计算性能测试主要针对 CPU 进行，即不管是节点性能测试还是系统性能测试，都只选择使用节点的 CPU 执行基准测试程序。

[4] 浮点计算性能基准测试：浮点计算性能测试分别针对 CPU、加速卡及 CPU+加速卡混合结构进行。当前存在多种基准测试程序，每种基准测试程序都有一定的特点，能从不同的侧面反应节点或系统的浮点计算性能。这里选择四种有代表性的测试方法进行评测：稠密线性方程组求解、矩阵乘法运算、离线傅里叶变换、稀疏线性方程组求解。

（1）稠密线性方程组求解：通过并行求解 N 元一次稠密线性代数方程组来评价系统的浮点计算性能。

应选择不同规模（阶）的方程组分别进行测试，以较全面地反应系统的计算性能和扩展性。

（2）矩阵乘法运算：通过实矩阵并行乘运算来评价系统的浮点计算性能。

实矩阵乘计算公式为：

$$C = \alpha AB + \beta C$$

其中，$A, B, C \in R^{m \times n}, \alpha, \beta \in R^{n}$。

矩阵乘法运算具有时间局部性和空间局部性都非常高的特点，适合 SP 模式和 EP 模式测试，测试主要反应单处理器浮点运算的最大速度。

（3）一维离散傅里叶变换：通过并行求解一维 DFT 来评价系统的浮点计算能力。FFT 的计算公式如下：

$$y_k = \sum_{j=1}^{m} x_j e^{2\pi i jk/m}, 1 \le k \le m$$

其中，$x_j, y_k \in C^m, 1 \le j, k \le m$

傅里叶变换的计算过程中，每一轮计算都要利用上一轮计算的数据进行重新组合，故数据重用多，在存储访问上有时间局部性高而空间局部性低的特点。所以一维 DFT 集中体现具有此类特征的应用在节点和集群系统中的性能表现。

（4）稀疏线性方程组求解：相比较稠密线性代数方程组求解，采用共轭梯度法的稀疏线性方程组求解在评价节点或系统的浮点计算性能上的应用逐渐受到人们的关注。它具有如下优点：共轭梯度超级计算基准测试旨在对现实世界应用程序的数据访问模式进行建模，测试超级计算机系统的内存子系统和内部互连的限制对其计算性能的影响。因为内部绑定 I/O，所以相比较 LinPack 类测试，共轭梯度法测试通常只能达到计算机峰值速率的一小部分，但被认为是更接近于实际应用的测试指标。

稀疏线性方程组求解程序在利用差分方程进行迭代计算的过程中，使用到了主要的通信模式，如全局和邻近进程数据的通讯，和主要的计算模式，如向量更新、点乘，稀疏矩阵向量乘和局部三角求解器。因此稀疏线性方程组求解作为一个基准测试程序，足够简单，但在通讯和计算模式上实现了对大部分模式的覆盖，具有代表性，在现代高性能计算机基准测试中被广泛采用。

以上基准测试程序可根据系统的软硬件特点，在计算和通信模式上进行必要的优化。优化后的程序在性能上应该有较大提升，对比分析优化前后基准运行速率优化运行速率的变化情况，为集群系统上部署实际应用提供参考。

上述测试原则上可以对 CPU、加速卡、CPU+加速卡混合结构分别进行，可以分别得到仅 CPU 参与计算时的计算速率、仅 GPU 参与计算时的计算速率、及

CPU+加速卡混合共同参与计算时的计算速率。不同的测试结果可以为应用选择不同的计算单元提供参考。

[5] 计算效率：由于受各种因素的影响，系统（节点）的实测性能往往比其峰值（理论）性能低得多。针对计算性能的评估，定义两个效率指标，以反应系统（节点）计算性能的实际发挥水平。

E_l= 稠密线性方程组法基准测试速率/节点计算理论峰值速率

E_g= 稀疏线性方程组法基准测试速率/稠密线性方程组法基准测试速率

E_l是系统（节点）理想情况下最大实测计算速率（稠密线性方程组法基准测试速率）与理论峰值速度的比值，体现了系统（节点）理论峰值性能可转化为实测峰值性能的比例，代表系统（节点）的理想可用计算效率。

E_g是系统（节点）的稀疏线性方程组法基准测试速率与理想情况下最大实测计算速率的比值，体现了理想情况下的最大计算性能转化为贴近实际情况的计算性能的比例，代表系统（节点）的实际可用计算效率。

[6] 内存性能测试指标：内存性能指标主要用于评估节点内存的访问性能，包含两项基本指标：持续内存带宽和内存随机访问速率。

（1）持续内存带宽：内存的数据访问和传输速度决定了 CPU 和内存之间的数据交换速率，对系统的计算性能存在潜在的影响，而且随着现代计算机 CPU 性能的快速增长，CPU 与内存之间的通信瓶颈问题日益突出。通过评测持续内存带宽，反应系统中内存子系统的性能，可作为系统性能评测的有益补充。

持续内存带宽度量内存数据的传输速率。通过执行四个简单的核心向量运算进行测量，并评估四个核心向量运算的计算性能。这四个核心向量运算如表 1.2 所示。

表 1.2 核心向量运算操作

名称	操作	每次迭代	
		字节数	FLOPS
COPY	a(i) = b(i)	16	0
SCALE	a(i) = q*b(i)	16	1
SUM	a(i) = b(i) + c(i)	24	1
TRIAD	a(i) = b(i) + q*c(i)	24	2

其中，a、b、c 为双精度数组，q 为双精度浮点数。这些操作被定义成有代表性的长向量运算。数组的大小被定义为每个数组都大于要测试的机器的缓存，

并且利用结构化的代码，杜绝 cache 数据重用的可能。

持续内存带宽度测试将内存子系统的测量与 CPU 峰值性能测试解耦，独立体现了系统内存子系统的性能，对系统性能评价具有重要的意义。

（2）内存随机访问速率：随机内存访问的性能直接影响应用性能，应用中即使比例很小的随机内存访问（缓存缺失）都会严重影响该应用的整体性能。由于目前处理器通常牺牲随机访问性能来换取顺序访问的性能，因此以随机访问性能为度量来评判系统内存访问性能和对应用的影响就尤为重要。

内存随机访问性能测试测量在内存数据随机读写时，“随机访问一个内存单元、读取其中原来的内容、进行一定的修改后再写回该单元”这种随机更新操作单位时间内完成的次数，记为 GUPS 率 (Giga UPdates per Second)。

$$\text{GUPS} = \text{随机内存更新操作的次数} \ / \ 10^9$$

在上述随机更新操作中，数据从内存读到 cache 的时间局部性和空间局部性都非常低，因此能够屏蔽硬件上可能采用的技巧，有效地检验了系统的内存延迟特性，为相关应用提供重要参考。

[7] 存储性能测试指标：存储性能指标用于评估节点或系统外存储的性能。其中，节点的存储性能评测主要评估节点本地存储的性能，系统的存储性能评测主要评估网络共享并行存储的性能。

节点本地存储指除系统提供的网络共享并行存储之外的本地外存储设备，一般为硬盘或 SSD（如果没有本地存储，本项可以不评测）。在高性能计算集群系统中，节点本地存储一般不作为主存储使用，而是作为应用程序执行计算时的临时数据缓存空间。如果计算程序执行中涉及内、外存缓存数据的交换，节点本地存储的 IO 性能将极大地影响应用程序的执行效率。因此，准确评估节点外存储性能，对于指导高性能计算集群系统建设和优化应用性能具有重要意义。

网络共享并行存储是系统的主要存储。并行存储的性能极大影响集群系统的 IO 性能，对应用程序，尤其是具有大量 IO 操作的应用程序的执行速度和效率有关键性影响。准确评估系统并行存储的性能是高性能计算集群系统评测的一个重要方面。

存储性能测试主要测量外存储（节点本地存储或系统并行存储）的 IOPS、带宽、响应时间和元数据处理速度。

（1）IOPS

IOPS 指系统在单位时间内能处理的最大 IO 频度，一般指单位时间内能完成的随机小 IO 个数。

在保证系统环境配置基本相同的情况下，测量以下四项分项指标和一项综合指标：

（i）随机读 IOPS：在 100%随机读负载情况下，通过读取大量随机分布在存储器不同区域的文件，测量节点本地存储或系统并行存储随机读的 IOPS。

（ii）随机写 IOPS：在 100%随机写负载情况下，通过将大量文件写入存储器的不同区域，测量节点本地存储或系统并行存储随机写的 IOPS。

（iii）顺序读 IOPS：在 100%顺序读负载情况下，通过在存储器的连续区域读取大文件，测试节点本地存储或系统并行存储顺序读的 IOPS。

（vi）顺序写 IOPS：在 100%顺序写负载情况下，通过在存储器的连续区域写入大文件，测试节点本地存储或系统并行存储顺序写的 IOPS。

（v）随机读写 IOPS：在 100%随机负载情况下，通过在存储器的不同区域同时执行文件的随机读取和写入操作，测试节点本地存储或系统并行存储随机读写速率。

需要注意的是，IOPS 数值受数据块大小和工作负载等多种因素的影响，即使使用一个标准的系统测量 IOPS 可能也有较大的差异，测量时应充分考虑工作负载的影响。

（2）带宽

带宽（memory bandwidth）指单位时间里存取的信息量，用单位时间内读出/写入的位数或字节数衡量。带宽是度量存储设备数据传输速率的技术指标，决定了以存储设备为中心获取信息的传输速度。

理论带宽：设带宽用 B_m 表示，存储周期用 t_m 表示，每次读/写 n 个字节，则理论带宽 $B_m = n/t_m$。

带宽的基准测试应使用不同的工作负载进行评估。根据 SPC-2 标准，定义了三种不同负荷模型，用以衡量存储系统在连续大规模移动数据时的性能:

（i）大文件处理模型：该模型模拟同时读写多个大容量模型的应用场景，这些场景一般常用在科学计算和大规模金融计算领域中；

（ii）大数据量的数据库查询模型：该模型模拟数据之间的大量连接和全表扫描应用场景，这些场景一般常用在数据挖掘和常务智能领域；

（iii）VOD 模型：该模型主要模拟非线性编辑应用场景，会同时读取多个大的影音文件并写入存储系统中。

IOPS 和带宽分别代表了存储系统不同的性质。一般认为，读写大容量连续数据，适合以带宽为标准，应选择高带宽存储系统。而传输小块、不连续数据，则适合以 IOPS 为标准，选择高 IOPS 的存储系统。

（3）响应时间

响应时间等于一个 I/O 的完成时刻减去开始的时刻所得的时间段，即该 I/O 请求的完成时间。对被测系统，可以测量在轻量级负载（通常不超过 10%负载）情况下的存储响应时间（称为最小响应时间），也可测在重量级负载（90%以上负载）、大批量、多任务、高并发应用情况下的响应时间（体现存储系统在有大量并发任务请求、I/O 请求近于饱和的情况下的性能）。

平均响应时间=被测的所有 IO 的响应时间之和/所有被测 IO 的数量。

对于应用程序而言，I/O 请求量过大，存储系统会无法及时响应，导致响应时间变长，甚至导致系统崩溃。

（4）元数据处理

在实际应用中，集群系统的并行存储可能管理着数以亿计的文件，面临着比数据操作更多的元数据操作。元数据的存取性能成为整个并行文件系统性能的关键，评测元数据处理性能成为评测文件系统性能的一个关键指标。

通过模拟对文件、目录及层次结构的目录的 open/stat/close 操作，测量系统的元数据处理能力。

系统的并行文件系统管理着海量的文件和目录，其元数据处理性能尤为关键，是元数据性能测试的主要对象。对于单节点的本地文件系统的元数据处理的性能评测可以参照进行，一般不作为硬性指标。

（5）网络性能测试指标

分为内部网络和外部网络两部分。

内部网络评测

内部网络评测主要测量内部网通信延时和带宽以及处理器对间的数据传输

速率。

内部网通信延时和带宽分两种网络拓扑评测：端到端和环形网络。端到端的测试在多对处理器之间进行：每对处理器之间，处理器 a 向处理器 b 发送一定量消息，然后处理器 b 将消息返回给处理器 a。在尽可能多的处理器对之间进行测量，从而获取各个连接之间的带宽和延时，其中最小带宽和最大延时是网络连接最薄弱的部分。

环形网络是将多个处理器连成环(可以按照进程编号组成自然顺序的环，也可以按照随机的方式进行连接)，环中每个处理器与其左右邻居之间发送和接收数据，测量环内通信的平均延时和带宽。

系统总有效带宽是系统内节点通信的聚合带宽，每个处理器的平均有效带宽是总有效带宽相对于参与通信的节点数的均值。

多处理器对间大数据传输速率：测量多对处理器之间同时进行两两通信，传输大量数据的速率和带宽，作为评测存储访问模型特点是空间局部性高而时间局部性低的一类应用的参考。

外部网络评测

以当前超算中心为基点，测试连接至其他地区超算中心的网络速率，为应用程序远程使用环境资源提供网络性能方面的参考。以超算中心为单位测试当前中心连接其他地区超算中心测试点的网络速率。分别测试通过不同服务提供商的链路连接至其他地区超算中心测试点的上行、下行速率。测试点指在各超算中心设置的测试连接点，应具有典型的网络连接属性。

1.2.1.2 应用性能指标

应用性能测试是通过在集群系统上执行一组选自某些专业领域的基准测试程序或典型应用程序，进一步测试系统执行相关应用的性能。通过应用性能测试可以评估系统对特定类型的应用的支持情况。

根据应用的专业领域，将应用性能测试进一步划分为高性能计算应用测试、深度学习应用测试及大数据分析应用测试。高性能计算应用测试通过执行 12 个选自不同专业领域的小应用程序（见表 1.2），评估系统整体的高性能计算应用性能。高性能计算应用测试结果应记录使用的应用程序、节点规模、执行时间、相

对加速比及并行效率等参数。其中，节点规模指的是本次执行所使用的节点个数，记为 N，单位：个；执行时间为本次执行的用时，记为 T，单位：ms；相对加速比为相比较单节点（或参照系统），所获得的加速比，记为 S；

$$S = T_0 / T$$

这里，T_0 是在单节点（或参照系统）上执行同样的程序，求解相同规模的问题所花费的时间，单位：ms

并行效率指的是相对加速比与节点规模的比值，记为 E，即：

$$E = S / N$$

表 1.3 高性能计算应用测试程序

应用程序	描述
AMG	稀疏矩阵的数学求解
UMT	建立在多个核分布式存储，多节点并行计算机系统上，执行三维非结构化空间网格上的时间依赖性，能量依赖性，离散坐标和非线性辐射问题的解决方案。为了实现极大的可扩展性，应用程序利用节点之间的消息传递和在节点内角度的线程算法进行空间分解。
GTC	用于燃烧等离子体中湍流运输的陀螺动力学粒子模拟。它是一个完全自我一致的 3D 粒子单元代码（Particle-in-cell code, PIC），具有非光谱泊松解算器和跟随磁场线（围绕环面的扭转）的网格。 它解决了实际空间中的陀螺平均 Vlasov 方程; Vlasov 方程描述了在自洽电磁场的影响下粒子系统的演化。未知量为通量 f（t，x，v），它是时间 t，位置 x 和速度 v 的函数，代表相位空间中的粒子（电子，离子，...）的分布函数。
GTC-P	通过使用粒子单元算法求解 Vlassov-Poisson 方程来模拟离子通过托卡马克的运动。在每个 PIC(particle-in-cell)时间步长期间，粒子的电荷分布被内插到网格上，泊松方程在网格上求解，电场从网格内插到粒子，并且根据电场更新粒子的相空间坐标。
MiniFE	MiniFE 中心的迭代解算器具备大型模拟器建模流体和结构动力学常见的性能特点，执行完整的有限元生成、装配和解析。物理域是由六面体单元模拟的三维盒子。该盒子离散为结构化网格，但被视为非结构化网格。使用递归对分法 (RCB) 分解域以支持并行执行。MiniFE 的主要计算内核是未经过预处理的共

	轭梯度迭代解算器。
MiniGhost	miniGhost 是一种有限差分微型应用程序，它用于模拟不同模板通过均匀三维域。测试模拟了在 Dirichlet 边界条件下通过同质域的热扩散。它目前不能解决可以检查正确性的具体问题。但是它可以在有限的意义上检查正确性的模式下运行：将域初始化为 0，将热源应用于全局域中间的单个单元，将网格值相加和初始值进行比较用于验证。
SNAP	作为代理应用程序来建模现代离散坐标中性粒子传输应用程序的性能。SNAP 被认为是 Sweep3D 的更新，旨在用于混合计算架构。它是由洛斯阿拉莫斯国家实验室代码 PARTISN 建模的。
MILC	代码表示由用于研究量子色力学(Quantum Chromodynamics)的 MIMD 晶格计算(MILC)协作的一组代码，属于亚原子物理学强相互作用的理论，通过并联机器进行四维格子规格理论的仿真。属于物理研究和模拟场景下的基准。
MiniDFT	用于建模材料的平面波密度泛函理论(Density Functional Theory)的模拟应用程序。MiniDFT 使用 LDA 或 PBE 交换相关函数计算 Kohn-Sham 方程的自相一致解。对于自相一致的场循环的每次迭代，构建 Fock 矩阵，然后对角化。为了构建 Fock 矩阵，使用快速傅立叶变换将平面波基(其中最容易计算的动能)转换为实际空间(其中电位被评估)和返回的轨道。
Meraculous	一种大规模并行基因组组装基准，构造并遍历存在于冗余短序列输入数据集中的长度为 k(k-mers)的所有重叠子串的 de Bruijn 图。通过遍历 de Bruijn 图，并发现所有(可能断开的)线性子图，Meraculous 能够构建基因组数据的高质量连续序列。
PENNANT	一款用于高级架构研究的应用程序。它具有用于操纵包含任意多边形的二维非结构化有限元网格的数据结构。PENNANT 使用几何域分解支持 MPI 并行性，对使用 MPI 调用的处理器上实现的点数据进行采集和散射操作，还支持使用 OpenMP 或 CUDA 的线程并行。
MiniPIC	解决具有反射壁的任意域中的静电场中的离散 Boltzman 方程。MiniPIC 基准测试使用非结构化的基于 hex 或 tet 的网格以及用于粒子网格的静态分区。粒子被跟踪到每个单元格交叉区，打包后并使用 MPI 传递到相邻的处理器。主要代码库使用了 Trilinos 数学库中的 Tpetra 对象进行矩阵/向量操作。

深度学习应用测试是通过在集群的节点上执行深度学习领域的基准测试程序，进一步测试系统的深度学习性能。通过深度学习应用测试可以评估系统节点对深度学习应用的支持情况。

深度学习基准测试结果记录使用的基准测试程序及执行时间等参数。其中，执行时间指的是完成指定基准测试程序所消耗的时间，记为 T，单位：ms。

大数据分析应用测试是通过在系统上执行相关研究领域的基准测试程序，进一步评估系统在进行相关研究领域的海量数据处理和分析时的性能。通过大数据分析应用测试可以了解系统对特定研究领域的大数据分析应用的支持情况。

基准测试程序可以在不同的研究领域、不同的工作负载实现及不同的数据集上执行。测试时应选择相应的研究领域、工作负载实现及数据集等参数，以评估系统数据查询工作负载及复杂分析工作负载的性能。其中，数据查询工作负载测试主要评价系统的文件格式压缩效率及数据查询性能；复杂分析工作负载测试则主要评价系统在执行复杂分析（如协方差、奇异值分解及 QR 分解等）方面的性能。

1.2.2 可用性指标

超算中心资源的可用性主要体现在是否配备有丰富的系统平台、工具环境与应用软件，是否能够满足应用的各种需求，是否能够为应用提供良好的使用体验。从两个方面衡量超算中心资源的可用性：软件资源配置情况和资源服务情况。

1.2.2.1 软件资源配置

软件资源配置反映超算中心系统平台、工具环境与应用软件的多样性，资源配置越丰富越能为更多的应用提供支持。同时，超算中心的某些特定软件资源可为一些特殊应用提供支持。

1) 操作系统的类型及版本支持，包括操作系统类型、版本及备注等信息。

2) 编程语言与环境

编译器：内容包括类型、版本及备注等信息。

并行库：内容包括类型、版本及备注等信息。

加速器开发包：内容包括类型、版本及备注等信息。

商业/开源软件

（i）商业软件

对宏观、建模与数据处理、计算流体力学、多体动力学、耐久性与疲劳、结构优化、结构分析、安全碰撞、跌落、热分析、超大规模精细电磁计算平台、计算结构力学分析、钣金成形模拟分析、多体动力学分析、计算流体力学分析、CFD 前后处理软件、CAE 前后处理软件、多目标优化分析、过程集成和优化分析、计算流体力学前处理等商业软件的支持情况。

（ii）开源软件

对数值天气预报与研究、海洋、大气环境、宇宙、地球物理、地球系统、天文、宏观、纳观、介观、微观、生物医学、生物化学、细胞生物学、生物信息学、分子生物学、数据处理和统计分析、科学计算、工程计算、图形图像、计算材料学、药物设计、生物信息、计算生物学、材料科学、计算化学、天体物理、流体力学等 28 个分类的开源软件的支持情况。

1.2.2.2 资源服务

（1）资源接入时间/运行率

内容包括中心名称、设备名称、接入时间及运行率等。

资源接入时间指超算中心的集群系统本统计周期内接入环境的总时间，单位：天。

运行率 = 接入天数 / 统计周期总天数

（2）资源利用率

内容包括中心名称、设备名称及利用率等。

资源利用率指超算中心的集群系统本统计周期内的利用率。

（3）平均排队时间

内容包括中心名称、设备名称及平均排队时间等。

平均排队时间指超算中心的集群系统本统计周期内提交的所有作业的平均排队时间，单位：s。

1.2.3 可靠性指标

内容包括超算中心名称、设备名称、故障频次、故障率、重大故障次数、最长连续运行天数及平均连续运行天数等。其中故障频次和故障率分为节点故障、存储故障、网络故障、接入故障、调度系统故障及其他故障等六个小类。

说明：

[1] 如果超算中心有多台设备，复制设备项目，逐台填报。

[2] 故障类型可根据情况增减。

[3] 故障频次指本统计周期内每种故障发生的次数。故障率是各种类型的故障发生次数在总故障次数里的占比。

[4] 重大故障指造成宕机、30%以上节点不可用的故障，统计本统计周期内重大故障发生的次数。

[5] 最长连续运行天数是本统计周期内设备不停机连续运行的最长时间。

[6] 平均连续运行天数是本统计周期内设备平均每次连续运行的时间。

平均连续运行天数 = 总运行天数 / 开机次数

1.2.4 安全性指标

安全性指标以超算中心为单位填报。

1、基础设施的安全保障

内容包括中心名称、数据灾备、电力保障、基础环境设施及相应的情况说明等。其中，数据灾备需要统计有无灾备（有/无）及灾备方式（本地灾备/异地灾备）等；电力保障需要统计是否双路市电（是/否）、是否有自备电源（是/否）及 UPS 延时时间（h）等；基础环境设施需要统计空调（很好/好/较好/一般/差/很差）及供水（很好/好/较好/一般/差/很差）等。

2、网络和信息安全

内容包括中心名称、网络安全、信息安全及相应的情况说明等。其中，网络安全需要统计是否有网络防火墙（有/无）、是否配有 VPN（是/否）及是否能够 Internet 直连（是/否）等；信息安全需要统计是否通过（ISO27001）安全认证（是/否）、是否具有保密资质（有/无）、是否有内部安全制度（有/无）、

是否设有专职信息安全人员（有/无）及专职信息安全人员的数量等。

1.2.5 需求管理指标

需求管理指标主要用于评价超算中心的管理水平。

1、需求计划管理

（1）年度经营计划

提供佐证材料

（2）市场调研计划

提供佐证材料

（3）软件更新计划

提供佐证材料

（4）设备升级改造计划

提供佐证材料

2、需求响应

（1） 市场需求调研

次数/报告列表

（2）上年度经营完成情况

提供佐证材料

（3）软件更新列表

内容包括序号、软件名称、版本及用途等。

（4）硬件更新列表：

内容包括序号、设备名称、型号、数量及用途等。

1.2.6 技术支持与服务响应指标

技术支持与服务响应指标用于评价超算中心技术服务的能力、水平和质量。

1、 服务能力

（1）技术支持服务台制度

（2）技术支持人员数量

（3）技术支持方式

2、 资料更新

内容包括序号、资料名称、版本及用途等。

3、 技术培训

内容包括序号、培训内容、培训地点、人数及合计人数等。

4、 故障响应

内容包括故障类型、承诺时间及响应达标率等。其中，故障类型分为节点故障、存储故障、网络故障、接入故障、调度系统故障及软件故障等。

5、 服务评价

（1）投诉人次

（2）满意度

2、资源评价计算方法

本部分对资源评价中使用方法的技术原理进行说明。

2.1 基础性能评价

2.1.1 理论浮点峰值

理论浮点峰值是指计算机系统理论上每秒钟能完成浮点计算最大次数，它主要由计算核的主频决定。

（1）对于纯 CPU 架构，理论浮点峰值为

R_{Peak}^{C} = CPU 主频 × CPU 每个时钟周期执行浮点运算次数 × 系统 CPU 总核数，

其中 CPU 主频单位为 GHz；每个时钟周期执行浮点运算的次数是由处理器中浮点运算单元的个数及每个浮点运算单元在每个时钟周期能处理几条浮点运算来决定的。以 IBM POWER4 为例，每个 POWER4 的处理器有两个浮点运算单元，每个浮点运算单元在一个时钟周期内可以同时处理一个加法和一个乘法的操作，所以如果处理器的主频为 1.7GHz（POWER4+），那么该处理器的峰值速度为$1.7\text{Ghz} \times 2 \times 2 = 6.8\text{Gflops}$。

（2）加速器的理论浮点峰值为

R_{Peak}^{S} = 加速器频率 × 加速器每个时钟周期执行浮点运算次数 × 系统加速器总核数，

其中加速器频率为加速器加速频率，即最大运行频率，单位为 GHz；每个时钟周期执行浮点运算次数这里指的是加速器每个时钟周期能完成的单精度浮点运算次数。

需要注意的是，不是所有加速器都能进行双精度浮点运算，因此在计算加速器的双精度理论浮点峰值时，总核数为能进行双精度浮点运算的核心数。以 NVIDIA Tesla K80 为例，Telsa K80 采用双核心 GPU 设计，拥有 CUDA 核心数量为 2496 个，加速频率为 875Mhz，那么该加速器的理论单精度浮点峰值为$0.875\text{Ghz} \times 2 \times 2496 \times 2 = 8736\text{Gflops}$，即 8.7Tflops；其中只有$1/3$的核具有双精度浮点单元，则该加速器的理论双精度浮点峰值为$8.7\text{Tflops}/3 = 2.9\text{Tflops}$。

对于拥有异构并行的计算系统，其理论浮点峰值为

$$R_{Peak}^{H} = R_{Peak}^{C} + R_{Peak}^{S}。$$

2.1.2 整数计算测试

CPU 性能可以理解为数据运算性能，分为整数运算和浮点数运算。根据整数运算特性，一般编译、压缩、排序、视频压缩转换和 XML 处理等功能都是依赖 CPU 整数运算性能。此处采用 SPEC2017 的 SPECrate2017 Integer 测试计算系统的整数性能。测试基中包含 10 个测试包。测试包如表 2.1 所示：

表 2.1 整数运算性能测试包

测试程序名	源码语言	应用领域	标准测试时间
perlbench	C	Peal 语言解释器	1591
gcc	C	GNU C 编译器	1415
mcf	C	路径规划	1615
omnetpp	C++	离散事件模拟-计算机网路	1311
xalancbmk	C++	XML 文档/XSL 表到 HTML 文档的转换	1055
x264	C	视频压缩	1751
deepsjeng	C++	人工智能：国际象棋（α - β 树搜索）	1145
leela	C++	人工智能：围棋（蒙特卡洛树搜索 ）	1655
exchange2	Fortran	人工智能：九宫格（递归解法产生器）	2619
xz	C	数据压缩	1076

程序可以测试出每个测试包完成 n 次的执行时间T_{SUT}^{i}，单位为 s，则每个测试的计算速率为：

$$R_i = \frac{n}{T_{SUT}^{i}}, 1 \le i \le 10$$

其中计算速率R_i的单位为 jobs/s。

执行时间与参考执行时间T_{Ref}^{i}相比较能得出每个测试的计算效率，计算公式为：

$$Ratio_i = \frac{nT_{Ref}^{i}}{T_{SUT}^{i}}, 1 \le i \le 10。$$

将所有计算效率求几何平均值，可得出综合整数运算效率，计算公式为：

$$Ratio_A = \left(\prod_{i=1}^{10} Ratio_i\right)^{1/10}$$

可见，计算效率越高代表单位时间完成的运算次数更多，性能越好。

2.1.3 稠密线性方程计算速率

稠密线性方程计算速率通过对高性能计算机采用高斯消元法求解 N 元一次稠密线性（dense linear）方程组来测试计算机的浮点性能，单位 GFlops。

稠密线性方程计算速率测试方法也叫高度并行计算基准测试，它对数组大小 N 没有限制，求解问题的规模可以改变，除基本算法（计算量）不可改变外，可以采用其它任何优化方法。用户在不修改任意测试程序的基础上，可以调节问题规模大小、使用 CPU 数目、使用各种优化方法等来执行该测试程序，以获取最佳的性能。

(1) 计算方法

稠密线性方程计算速率通过求解一个 N 元一次稠密线性方程组来测试计算机的浮点性能，如下式所示：

$$Ax = b$$

其中，$A = (a_{ij})_{N\times N}$且为非奇异矩阵，$b = (b_1, b_2, \dots, b_N)^T$，$x = (x_1, x_2, \dots, x_N)^T$，A 与 b 均为已知，而 x 是待求的 N 维列向量。

求解问题规模为 N 时，浮点运算次数为$\frac{2N^2}{3} + \frac{3N^2}{2}$。因此，只要给出问题规模 N，测得系统计算时间$T_{DL}$，单位为 s，则测试结果为：

$$R_{DL} = \frac{\frac{2N^2}{3} + \frac{3N^2}{2}}{T_{DL}} \times 10^{-9}$$

计算机的理论峰值为R_{PEAK}，则执行效率为：

$$\eta = \frac{R_{DL}}{R_{PEAK}}$$

（2）测试方法

a）环境准备

在测试稠密线性方程计算速率之前，系统中必须已经安装了编译器、并行环境 MPI 以及基本线性代数子方程(BLAS)或矢量图形信号处理库(VSIPL)两者之一。

编译器必须支持 C 语言和 Fortran77 语言。并行环境 MPI 一般采用 MPICH，当然也可以是其它版本的 MPI，如 LAM－MPI。HPL 运行需要 BLAS 库或者 VSIPL 库，且库的性能对最终测得的 Linpack 性能有密切的关系。常用的 BLAS 库有 GOTO、Atlas、ACML、ESSL、MKL 等。

b）优化方法

HPL 的性能调优涉及的面很多，是一项复杂而永无止境的工作。HPL 性能主要受三个因素的影响，分别是硬件因素、软件因素和 HPL 执行参数。

硬件因素主要包括 cache 大小和存储系统结构、访存速度、处理器性能、计算机系统的结构以及互联网络的性能等，这些因素都会影响机器的 HPL 性能。

软件因素主要指的是 MPI 和 BLAS 对 HPL 性能的影响。MPI 常用的有 LAMMPI、MPICH 和 OpenMPI，不同的 MPI 在运行方面有一定差别。对于 MPICH 来说主要有两种运行方法。

BLAS 库中的 ESSL、MKL 和 ACML 分别由 IBM、Intel 和 AMD 开发，并且对各自的处理器支持比较好。选择哪种 BLAS，不仅要参考计算机硬件类型，还要通过实验分析。

HPL 执行参数很多，在文件 HPL.dat 中进行设置，其中对 HPL 性能影响比较大的是处理器网格的排列方式$P \times Q$、矩阵规模 N 和矩阵分块大小 NB。

2.1.4 矩阵乘法计算速率

此测试方法是通过执行矩阵乘法操作，测试浮点运算性能，可以分别测试整数、单精度和双精度计算速率。其通过矩阵 A、B 和矩阵 C 的旧值根据以下公式常规计算矩阵 C 的新值：

$$C \leftarrow \alpha AB + \beta C;$$

其中，$A, B, C \in R^{n\times n}$；$\alpha, \beta \in R^{n}$。

上述算法的计算规模为$2n^3$，而结果的正确性通过计算残差进行验证：

$$\left\|C-\hat{C}\right\|/(\varepsilon n\|C\|_F)。$$

矩阵乘法测试方法与稠密线性方程测试方法的共同之处在于主要的计算都是浮点数加法和浮点数乘法，调用 BLAS 库运算，尽可能把实际运算速度向处理器的峰值速度逼近；不同的是稠密线性方程测试的是整个系统全局的浮点数运算性能，考虑到了通信速度的影响，因此多个处理器内部互连的性能对结果影响很大，而矩阵乘法仅测试出单个处理器的浮点运算最大速度。

由于矩阵乘法计算不涉及通讯，影响测试速率的因素只有矩阵规模 n 以及 BLAS 库的设置。

2.1.5 DFT 计算速率

离散傅立叶变换（Discrete Fourier Transform，DFT）是数字信号处理常用的变换方法。N 点 DFT 的计算公式为：

$$X_k=\sum_{j=0}^{N-1}x_j\omega_N^{jk}；$$

其中，$\omega_N^{jk}=e^{-2\pi i/N}$称为旋转因子，$0\le k\le N$，$i=\sqrt{-1}$，$x_j, X_k\in C^N$。旋转因子具有对称性、周期性和可约性。

由于 DFT 计算量大，对内存需求也大，因而难以实施实时处理。通过快速傅立叶变换（FFT）算法，把长序列 DFT 变成短序列的 DFT，从而减少运算量。以基 2 的 FFT 算法为例，其基本思想是将 N 点序列 x_j 对半分开，利用旋转因子的性质将 N 点 DETy_k，$k=0,1,\ldots,N-1$，分解成两个 N/2 点的 DFT，依次可继续向下分解。下图给出了 N=8 的 FFT 分解计算流程图：

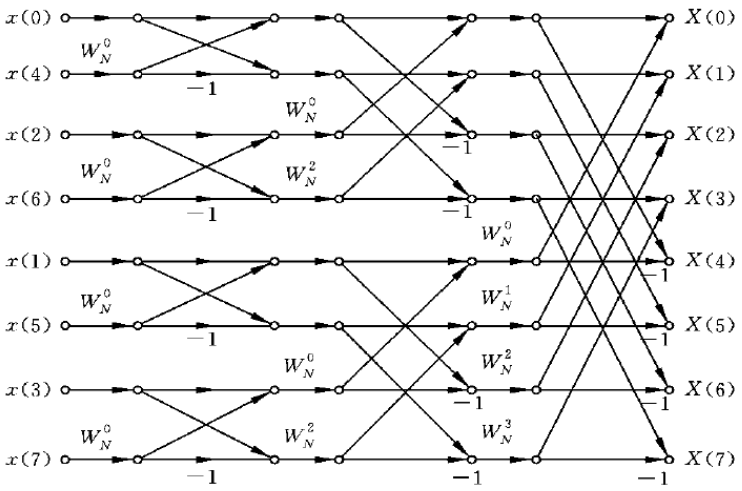


图 2.2 基为 2 的一维 FFT 计算过程示意图

同理可扩展到基为 3 或者 5 的 FFT 算法。而 FFT 算法的计算量为 $5(\mathrm{N}-1)\log_2(\mathrm{N}-1)$，可得出 DFT 计算速率为：

$$\mathrm{R_{DFT}}=\frac{5(\mathrm{N}-1)\log_2(\mathrm{N}-1)}{\mathrm{T_{DFT}}}\times 10^{-9}$$

其中$\mathrm{T_{DFT}}$为计算一次 N 点 DFT 的总时间，单位为 s；得出 DFT 计算速率$\mathrm{R_{DFT}}$单位为 TFlops。

由上述计算过程可以看出 ，每一轮的计算都要利用上一轮计算的数据重新组合，所以 FFT 算法中在时间上的数据重用很多，虽然也是测试浮点运算速度，但和稠密线性方程测试不同的是： 其存储访问特点为时间局部性高而空间局部性低。

(1) 单 CPU 的 DFT 计算速率

选择一个处理器测试双精度、一维、复数 DFT 的浮点执行性能，基可以选择 2、3 或者 5。在基准测试中可以调节序列的规模 N，使其大小为节点内存的一半。在优化测试中可以使用 FFTW 库，选择库的不同选项可以缩短计算时间。

(2) 单加速器的 DFT 计算速率

选择一个加速器测试单精度、一维、复数 DFT 的浮点执行性能，基可以选择 2、3 或者 5，能使用 CUDA 加速。在基准测试中可以调节序列的规模 N，使其大小为节点内存的一半。在优化测试中可以使用 FFTW 库，选择库的不同选项可以缩短计算时间。

(3) EP 模式 CPU 的 DFT 计算速率

系统中所有处理器同时测试双精度、一维、复数 DFT 的浮点执行性能，相互间不进行通信，基可以选择 2、3 或者 5。在基准测试中可以调节序列的规模 N，使其大小为节点内存的一半。在优化测试中可以使用 FFTW 库，选择库的不同选项可以缩短计算时间。

(4) GP 模式 CPU 的 DFT 计算速率

系统中所有处理器共同测试双精度、一维、复数 DFT 的浮点执行性能，基可以选择 2、3 或者 5。若系统共有 P 个处理器，序列规模 N>P，通过上面 FFT 基本原理的分析，可知输入的序列具有良好的数据无关性，可将数据分为 P 块，每个处理器负责一块数据，当下一轮计算需要时横向或者纵向将本轮结果发到相应的处理器。

在基础测试中可以调节序列的规模 N，使其大小接近整个系统所有内存的一半。在优化测试中处理可以使用 FFTW 库的不同选项，还可以对 MPI 编译器进行优化。

(5) EP 模式加速器的 DFT 计算速率

系统中所有加速器同时测试单精度、一维、复数 DFT 的浮点执行性能，相互间不进行通信，基可以选择 2、3 或者 5。在基准测试中可以调节序列的规模 N，使其大小为节点内存的一半。在优化测试中可以使用 FFTW 库，选择库的不同选项可以缩短计算时间。

(6) GP 模式加速器的 DFT 计算速率

系统中所有加速器共同测试单精度、一维、复数 DFT 的浮点执行性能，基可以选择 2、3 或者 5。若系统共有 P 个加速器，序列规模 N>P，通过上面 FFT 基本原理的分析，可知输入的序列具有良好的数据无关性，可将数据分为 P 块，每个加速器负责一块数据，当下一轮计算需要时横向或者纵向将本轮结果发到到相应的加速器。

在基础测试中可以调节序列的规模 N，使其大小接近整个系统所有内存的一半。在优化测试中处理可以使用 FFTW 库的不同选项，还可以对 MPI 编译器进行优化，以及使用 CUDA 加速。

(7) 异构并行的 DFT 计算速率

对于采用 CPU-GPU 异构计算系统，可以测试单精度、一维、复数 DFT 的浮点执行性能。测试可以选择 1MB、8MB、96MB、256MB 和 512MB，支持 OpenCL 和 CUDA。测试可以得出包含 PCI-E 总线传输时间的计算速率和只包含 Kernel 执行时间的计算速率。假设 Kernel 执行时间为T_K，PCI-E 总线传输时间为T_P，则包含 PCI-E 总线传输时间的计算速率为：

$$R_{DFT}^{a} = \frac{5(N-1)\log_2(N-1)}{T_K + T_P} \times 10^{-9}$$

而只包含 Kernel 执行时间的计算速率为：

$$R_{DFT}^{K} = \frac{5(N-1)\log_2(N-1)}{T_K} \times 10^{-9}$$

单位均为 TFlops。异构并行 DFT 计算测试的结果内容如下：

- 核心数
- 计算规模（MB）
- R_{DFT}^{CPU}（Tflops）
- R_{DFT}^{GPU}（Tflops）
- R_{DFT}^{K}（Tflops）
- R_{DFT}^{a}（Tflops）

2.1.6 稀疏矩阵计算速率

与前面计算速率关注线性方程的计算性能不同，稀疏矩阵测试方法求解一个三维离散微分方程模型问题，测试使用共轭梯度法迭代计算稀疏线性系统的计算速率 。

稠密线性方程测试方法衡量的是线性方程计算的速度和效率，测试中的数据主要是跨区段的访问，可以通过合理地组织计算来掩盖数据访问的延迟。而求解偏微分方程的稀疏矩阵要求超算系统的运算性能、内存容量、带宽以及互连性能之间取得平衡。简单来说，求解稠密线性方程更考验超算的处理器理论性能，而求解稀疏矩阵更看重实际性能，对内存系统、网络延迟要求也更高，更容易反映出有限元法和流体分析等超算实际应用的性能。

(1) 计算原理

求解偏微分稀疏矩阵中使用带有局部对称高斯塞德尔预条件子的预处理共轭梯度法（Preconception Conjugate Gradient，PCG）。相关代码采用 C++语言开发，并可以使用 MPI 和 OpenMP 编译。代码包含以下功能模块：

① 问题设置：生成对称正定矩阵 A。稀疏矩阵 A 使用按行压缩的存储模式，同时生成向量 b 和设置 x 的初始值。问题的规模可以通过一套固定的程序进行确定，以保证合理的使用硬件系统资源。测试能够使用不同格式存储的矩阵，测试程序不对构建初始矩阵的时间进行测量和记录。但是，使用原始的数据结构进行的矩阵向量乘的开销将被记录在计算时间内。尽管矩阵是对称的，形状有可能是规则的，或者近似规则，矩阵将进行非结构化的存储，并且保持一份包含矩阵所有值的拷贝。测试者不能利用矩阵的规则性，比如对稀疏的对称对角矩阵，使用特殊的结构来减少矩阵存储所需的空间。

② 预条件子设置：测试会为局部对称高斯塞德尔预条件子设置相应的数据结构。在代码中将使用按行压缩存储的格式存储分别存储上三角和下三角矩阵。对于这些矩阵，测试者可以自由的进行转换，这一部分的开销不计算在测试时间中，但是需要进行记录。这些使用原始格式存储的矩阵的一次对称高斯塞德尔消去法的开销将被归一化。

③ 验证和检验：测试代码能估算先决条件、后置条件和不变量，使用谱方法近似估计误差，使用 PCG 算法和 SPD 矩阵的一些特性对结果进行验证辅助检测迭代过程中的异常情况。

④ 迭代：算法中会迭代 m 步，重复执行 n 次，每次将使用同样的初始条件。m 和 n 需要取足够大的值来保证整个系统运行较长的时间。在 m 步迭代结束之后，残差$|x - \hat{x}|$可以用于验证精确性，其中$\hat{x}$是问题的数值解。在每执行完 k 次完整的 m 步迭代后，缓存将被刷新，同时计算时间的平均值将被记录下来。

⑤ 后处理和输出测试结果

(2) 测试结果

测试结果内容如下：

- 计算结果的检验与验证标准；
- 计算的时间与结果；

- 计算所使用的资源：节点数、存储大小、使用的处理器和加速处理器个数，精度、编译器版本、优化等级和编译命令、计算速率、功耗、缓存、加载和存储信息等；
- 检查点和故障重启信息（选择开启）。

2.1.7 可持续内存带宽

通过简单的向量运算测量内存的持续带宽，操作数组 a，b，c 必需符合要求：数组所占的空间必须超过与系统内存最接近的那一级 cache 的容量总和的 4 倍。例如，对于有两级 cache 的处理器，如果 L2 cache 为 512kB，那么数组长度至少为 256K 个元素（64 位双精度数 ,即 8Bytes）。因此，可持续内存带宽测试采用远远超过 cache 容量的数据集，并且利用结构化的代码，从而杜绝 cache 重用的可能。在测试中，数组的分配是从堆中动态分配大小，这样能够根据内存的大小适当的改变。

测试内容中，复制测试没有算术运算时的数据传输速率，后面三项逐渐加入了一些简单的加法和乘法操作。通过测试结果可以得到高性能计算机系统的内存持续带宽性能，公式分别如下：

$$\text{点乘：} V = \frac{2m}{T}$$

$$\text{加法：} V = \frac{2m}{T}$$

$$\text{综合：} V = \frac{3m}{T}$$

其中，T 为操作所用时间，单位为 s；m 为数组大小，单位为 GByte。

这项测试还反映了系统对大量 cache 不命中的情况的处理能力，以及系统的存储平衡性 (memory balance)。这里将平衡性定义为：

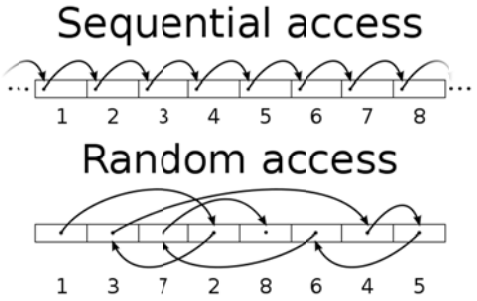
$$\text{balance} = \frac{\text{每周期浮点运算的操作次数}}{\text{每周期持续存储访问操作次数}}$$

在这种定义中，平衡性可以解释为在平均内存访问时间内能够执行的浮点操作次数，balance 越小表示平衡性越好。扩展性好指随着处理器数量增加 balance 变化较小。

可持续内存带宽测试将内存性能与系统的峰值运算性能分离开来，其存储访问的特点是连续访问很长的地址空间，因此空间局部性高，时间局部性低。

2.1.8 每秒读写操作的次数

每秒读写（I/O）操作的次数，即 I/O per second(IOPS)。机械硬盘的连续读写性很好，但随机读写性能很差。这是因为磁头移动至正确的磁道上需要时间，随机读写时，磁头不停的移动，时间都花在了磁头寻道上，所以性能不高。如下图：



在存储小文件(图片)、OLTP 数据库应用时，随机读写性能（IOPS）是最重要指标，衡量随机访问的性能。测试给出了 100%顺序读，100%顺序写，100%随机读，100%随机写，随机存取 5 种操作的 IOPS。

测试的时候请注意，设置的测试文件的总体大小一定要大过内存，建议为内存的两倍大小，否则计算系统会将读写的内容进行缓存，使得数值非常不准确。

2.1.9 存储带宽

存储带宽，或者吞吐量，指的是单位时间按内最大的 IO 流量。往往是采用大的 IO 块、大的带宽获得的最大流量。

IOPS 和带宽既相互独立又相互关联。一般来说，当涉及更多的频繁读写时（OLTP)，更多的考虑 IOPS 与响应时间；而一些大量的顺序文件访问，例如数据仓库应用（OLAP)，流媒体，更多的考虑带宽指标。

测试由 3 个不同负荷模型构成，主要衡量存储系统在连续大规模移动数据时的性能。这 3 种负荷模型包括：

（1） 大文件处理模型。该模型模拟同时读写多个大容量模型的应用场景，这些场景一般常用在科学计算和大规模金融计算领域中。

（2） 大数据量的数据库查询模型。该模型模拟数据之间的大量连接（join）和全表扫描应用场景，这些场景一般常用在数据挖掘和常务智能领域。

（3） VOD 模型。该模型主要模拟非线性编辑应用场景，会同时读取多个大的影音文件并写入存储系统中。

可以看出存储带宽测试盖了目前大量连续 I/O 的所有典型测试场景，因此能够很好地测量出存储系统的带宽。这 3 种模式的测试方法如下：

（1）大文件处理模型：测试会对单个文件大小分别为 1024KB 和 256KB 两种测试文件分别进行只写、读写和只读三种操作，共 6 个小测试，并分别计算带宽（MB/s），将总文件大小除以总用时得出综合存储带宽（MB/s）。6 个小测试需要依次连续执行，其顺序为：

只读，单个文件大小为 1024KB；

只读，单个文件大小为 256KB；

读写，单个文件大小为 1024KB；

读写，单个文件大小为 256KB；

只读，单个文件大小为 1024KB；

只读，单个文件大小为 256KB。

（2）大数据量的数据库查询模型：测试会对单个文件大小分别为 1024KB 和 64KB 两种测试文件分别在同时发送的 IO 请求数为 1 和 4 的情况下进行数据传输，共形成 4 个小测试，并分别计算带宽（MB/s），求取 4 个小测试测出带宽的平均值得出综合存储带宽（MB/s）。4 个小测试分别为：

同时发送的 IO 请求数为 4，单个文件大小为 1024KB；

同时发送的 IO 请求数为 1，单个文件大小为 1024KB；

同时发送的 IO 请求数为 4，单个文件大小为 64KB；

同时发送的 IO 请求数为 1，单个文件大小为 64KB。

（3）VOD 模型：测试存储在传输一定数量的大影音文件的带宽（MB/s）。

在大文件处理模型和大数据量的数据库查询模型中，每个小测试可以设置一次操作最大的文件数，每个小测试中会依次连续操作最大文件数的 100%、50%、25%、12.5%以及单个文件。

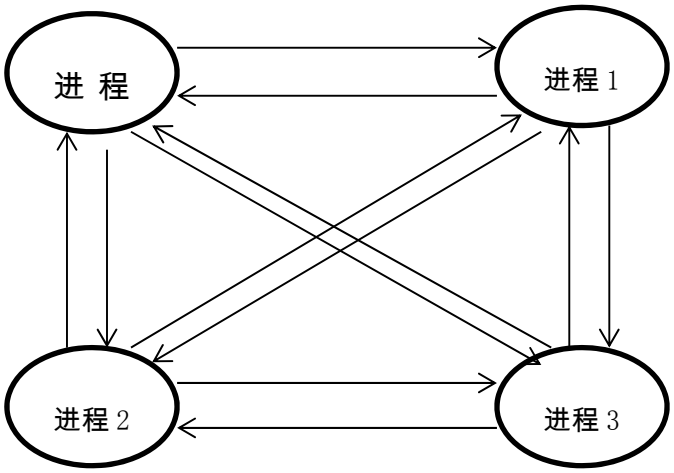
综合存储带宽为上述三个测试所得带宽的平均值。

2.1.10 内部网络

2.1.10.1 ping-pong 延时与带宽

ping-pong 模式是一种点对点的通信模式，指的是一对进程之间的通信模式，通过调整发送接收信息规模大小，获得对应的传输带宽和延时情况，再通过测试许多不同组对进程之间的网络延时和带宽，便可取得相应的最大值、最小值与平均值信息。在 ping-pong 进程对之间进行通信测试时，其他的进程处于接收阻塞等待状态。

假设测试系统中进程数 n_{pes}=4，那么任一个进程将分别与其他 3 个进程进行消息个数为 n 的“乒乓”通信。进程之间相互通信的关系如下图所示：



进程间通过调用 MPI_Sendrecv()函数，实现消息的发送和接收。首先记下任意两个进程间重复 n_{rpt} 次（设 n_{rpt}=100）“乒乓”通信所用的时间，最后求出每一进程分别与其他 3 个进程进行单程通信所需的时间 $t_{n(i)}$ (i=1, 2, 3)。

对同一个消息大小 n，根据每个进程得到的时间，分别求出所有进程之间通信的最小时间 t_{min}、平均时间 t_{ave}、最大时间 t_{max} 和最小带宽 b_{min}、平均带宽 b_{ave}、最大带宽 b_{max}。通过发送不同大小的消息进程测试，即 n 取不同的值，可以得到上述性能参数不同的测试值，n 的取值规律为$n = 2^m (m = 0,1,2 \cdots 15)$。

2.1.10.2 环延时与带宽

采用环网结构时，把所有进程串联形成一个环状，系统中所有进程被分配到某一个环中，每一个进程并行地向其左侧进程发送信息，同时从它右侧的进程接

收信息。按照进程在环中的排序方式分成随机序环和自然序环两种，如果按照进程的 rank 排序就属于自然序，否则就是随机序。在具体测试时，我们基于一维循环拓扑构建 6 种环模式：

在第一种环模式中，除了最后一个环规模可能是 2 或 3 外，其他所有环规模都是 2；例如如果 MPI_COMM_WORLD 有 7 进程, 则 rank0 和 rank1 的进程形成第一个环, rank2 和 rank3 形成第二个环, 而 rank4、rank5 和 rank6 形成第三个环；

在第二种环模式中，所有环大小规模都是 4，最后一个环大小可能是 1 * 3, 1 * 5, 或 2 * 5。如果所有进程总数量小于等于 7，则然后所有进程形成一个环；

在第三种环模式中，除最后一个环大小规模可能是 3*7, ... 1*7, 1*9, ... 4*9 外，其他环的规模都是 8；

在第四种环模式中，标准环规模大小为 min(max(16, size/4), size)；

在第五种环模式中，标准环规模大小为 min(max(32, size/2), size)；

在第六种环模式中，所有进程构成一个环；

自然序环是指在每一个环中，所有的进程都是根据其在拓扑结构中的 rank 值进行顺序排序的，而随机序环则是环中进程是按照 rank 值随机排序的。

为了保证自然序环和随机序环拥有同等权重，在计算时分两步分别计算其平均值的。

对于 sizeof(int) < 64 的操作系统，$L_{\max}$的值要小于等于 128MB，即 $L_{\max} = \min(128MB, \text{单进程平均内存量}/128)$。

2.1.10.3 mpiGraph 带宽

通常，每个节点（或每个互连链路）运行一个 MPI 任务。对于由 N 个 MPI 任务组成的作业，N 个任务在逻辑上排列在从等级 0 开始计数的环中。等级数向右递增，并且会在结束时回退到等级 0。随后执行共 N-1 个步骤。在每个步骤中，每个 MPI 任务向右向任务 D 发送消息，同时接收来自任务 D 的左向消息。D 的值从 1 开始并运行到 N-1，所以当 N-1 个步骤结束时，每个任务都向除自己之外的相邻任务发送了消息，并接收了来自相邻任务的消息。在运行结束时，两个 N×N 的带宽矩阵将被写入标准输出——一个为发送带宽矩阵，另一个为接收带

宽矩阵。

然后名为 crunch_mpiGraph 的脚本将被执行来将此输出生成为报告。报告包括一对表示不同任务配对之间带宽值的位图图像。此图像中的像素根据相对带宽值进行着色。最大带宽值被设置为纯白色（值 255），其他值则根据其与最大带宽值的百分比在纯白色与黑色（值 0）间进行缩放。可以通过将鼠标光标悬停在图像上来放大和检查图像的细节特征。至此，可以通过视觉上直观地观察到每个节点（或每个互连链路）间的互连性能。

2.1.11 系统总有效带宽

通过设置不同规模的消息大小，采用自然序环和随机序环等多种通信模式，来模拟实际应用中传输不同规模消息时分布式计算系统的通信网络累计有效带宽。

1. $b_{eff} =$

$$logavg\left(logavg_{ring\ patterns}\left(sum_L\left(max_{mthd}\left(max_{rep}\left(b(ring\ pat.,L,mthd,rep)\right)\right)\right)/21\right),logavg_{random\ patterns}\left(sum_L\left(max_{mthd}\left(max_{rep}\left(b(\text{random } pat.,L,mthd,rep)\right)\right)\right)/21\right)\right)$$

其中：

$b(pat.,L,mthd,rep)$

$= L * pat.$模式消息总数

$*$循环长度/每个进程循环长度次数中执行通信模式最大时间

每次重复测量 3 次（即 rep=1..3），使用所有重复测试中最大带宽值。

每种模式使用 3 种方法编程，使用所有方法中最大带宽值。

测量针对不同大小规模的消息，消息长度 L 的值有：

L = 1B, 2B, 4B, ... 2kB, 4kB, 4kB*(a**1), 4kB*(a**2), ... 4kB*(a**8)，其中 4kB*(a**8)=$L_{\max}$=每处理器平均内存/128。

循环长度是动态减少的，以实现每次循环执行之间在 2.5~5 毫秒之间，且最小值为 1。

所有消息规格的平均带宽通过$(sum_L(\cdots)/21)$来计算。

使用一组环型模式和随机模式。

针对环型模式和随机模式的平均值求对数。

最后有效带宽是这两个值的对数平均值。

2.1.12 多处理器对间大数据传输速率

假设任意矩阵 A、B

$$\mathrm{A},\mathrm{B} \in R^{n\times n}$$

通过

$$\mathrm{A} \leftarrow A^T + B$$

并行矩阵转置来计算成对处理器之间同时交互大量信息的传输情况，有助于测算系统内部互联总的通信能力。数据传输速率 V（Gbyte/s）的计算公式为：

$$\mathrm{V} = \mathrm{T}/n^2$$

即用矩阵转置所用时间除以矩阵元素个数，验证结果的公式为：

$$\left\|A - \tilde{A}\right\|/(\varepsilon\, n)$$

2.2 应用性能评价

2.2.1 高性能计算应用测试

以上的测试偏向于纯粹的工具性能测试，更具有普适性，而高性能计算应用测试则偏向于高性能计算领域的基准测试，测试更具有针对性。测试包含 12 个小测试程序，各个程序都可以选择大、中、小三种规模运行。下面将对各个测试程序进行简单介绍，然后介绍评价指标的计算方法。

2.2.1.1 测试程序

① GTC-P

GTC-P(Gyrokinetic Toroidal Code) 通过使用粒子单元算法求解 Vlassov-Poisson 方程来模拟离子通过托卡马克的运动。在每个 PIC(particle-in-cell) 时间步长期间，粒子的电荷分布被内插到网格上，泊松方程在网格上求解，电场从网格内插到粒子，并且根据电场更新粒子的相空间坐标。

② Meraculous

Meraculous 是一种大规模并行基因组组装基准，构造并遍历存在于冗余短序列输入数据集中的长度为 k(k-mers)的所有重叠子串的 de Bruijn 图。通过遍历

de Bruijn 图，并发现所有(可能断开的)线性子图，Meraculous 能够构建基因组数据的高质量连续序列。

③ MILC

MILC 基准代码表示由用于研究量子色力学(Quantum Chromodynamics)的 MIMD 晶格计算(MILC)协作的一组代码，属于亚原子物理学强相互作用的理论，通过并联机器进行四维格子规格理论的仿真。属于物理研究和模拟场景下的基准。

④ MiniDFT

MiniDFT 是用于建模材料的平面波密度泛函理论(Density Functional Theory)的模拟应用程序。MiniDFT 使用 LDA 或 PBE 交换相关函数计算 Kohn-Sham 方程的自相一致解。对于自相一致的场循环的每次迭代，构建 Fock 矩阵，然后对角化。为了构建 Fock 矩阵，使用快速傅立叶变换将平面波基(其中最容易计算的动能)转换为实际空间(其中电位被评估)和返回的轨道。

⑤ MiniPIC

MiniPIC 是解决具有反射壁的任意域中的静电场中的离散 Boltzman 方程。MiniPIC 基准测试使用非结构化的基于 hex 或 tet 的网格以及用于粒子网格的静态分区。粒子被跟踪到每个单元格交叉区，打包后并使用 MPI 传递到相邻的处理器。主要代码库使用了 Trilinos 数学库中的 Tpetra 对象进行矩阵/向量操作。

⑥ PENNANT

PENNANT 是一款用于高级架构研究的应用程序。它具有用于操纵包含任意多边形的二维非结构化有限元网格的数据结构。PENNANT 使用几何域分解支持 MPI 并行性，对使用 MPI 调用的处理器上实现的点数据进行采集和散射操作，还支持使用 OpenMP 或 CUDA 的线程并行。

⑦ SNAP

SNAP 作为代理应用程序来建模现代离散坐标中性粒子传输应用程序的性能。SNAP 被认为是 Sweep3D 的更新，旨在用于混合计算架构。它是由洛斯阿拉莫斯国家实验室代码 PARTISN 建模的。

⑧ UMT

UMT 是建立在多个核分布式存储，多节点并行计算机系统上，执行三维非结构化空间网格上的时间依赖性，能量依赖性，离散坐标和非线性辐射问题的解

决方案。为了实现极大的可扩展性，应用程序利用节点之间的消息传递和在节点内角度的线程算法进行空间分解。

⑨ MiniFE

MiniFE 是一个有限元小应用程序，它包含了几个代表隐式有限元应用程序。它在线性 8 节点六边形元素的砖形问题域上从稳态传导方程构成稀疏线性系统，然后使用简单的非预处理共轭梯度算法求解线性系统。

⑩ MiniGhost

miniGhost 是一种有限差分微型应用程序，它用于模拟不同模板通过均匀三维域。测试模拟了在 Dirichlet 边界条件下通过同质域的热扩散。它目前不能解决可以检查正确性的具体问题。但是它可以在有限的意义上检查正确性的模式下运行：将域初始化为 0，将热源应用于全局域中间的单个单元，将网格值相加和初始值进行比较用于验证。

 AMG

AMG 是由非结构化网格问题引起的多重网格线性系统的并行求解器。

 GTC

GTC 用于燃烧等离子体中湍流运输的陀螺动力学粒子模拟。它是一个完全自我一致的 3D 粒子单元代码（Particle-in-cell code, PIC），具有非光谱泊松解算器和跟随磁场线（围绕环面的扭转）的网格。 它解决了实际空间中的陀螺平均 Vlasov 方程; Vlasov 方程描述了在自洽电磁场的影响下粒子系统的演化。未知量为通量 f（t，x，v），它是时间 t，位置 x 和速度 v 的函数，代表相位空间中的粒子（电子，离子，...）的分布函数。

2.1.1.2 评价指标

① 加速比S_P

$$S_P = \frac{T_S}{T_P}$$

其中，T_S为最快的串行算法在单台处理器上的执行时间；T_P为并行算法在 p 台处理器上的执行时间。

S_P定义为解同一个问题的单机计算时间与并行机所需总时间之比，通常用来度量算法的并行性对计算时间的减少程度，或者说度量算法用多个处理器时使计

算速度提高的加速倍数。假设不考虑通信等开销，在最理想的情况下，若$T_S = t$，则$T_P = t/p$，有

$$S_P = \frac{T_S}{T_P} = \frac{t}{t/p} = p$$

即理想的加速比为 p，称为线性加速比。

由于并行算法总有一些额外的通信开销（如同步等待等），即有$T_P \geq t/p$，所以$S_P \leq p$。当处理器数 p 增加时，加速比S_P可提高，但 p 增加会使成本增加，也可能使处理器的利用率降低（如计算时间减少、通讯时间增多），因此S_P还不能全面评判一个并行算法的优劣。

② 并行效率E_P

$$E_P = \frac{S_P}{p}$$

由$S_P \leq p$可知，$0 < E_P \leq 1$。

并行效率E_P可以度量并行算法对处理器的利用率，并且可以补偿S_P评价指标的不足，E_P，S_P是一对互补指标。此外还有一种同时考虑E_P、S_P的综合指标：

$$F_P = \frac{S_P}{pT_P}$$

将E_P代入上式有

$$F_P = \frac{E_P}{T_P} = \frac{E_P}{T_s/S_P} = \frac{E_P S_P}{T_s}$$

即F_P与并行效率E_P和加速比S_P成正比关系。因此高效的并行算法应使F_P达到极大。

③ 应用测试综合计算速率 SSP

$$SSP = N\left(\prod_{i=1}^{12} \frac{R_i}{n_i}\right)^{1/12}$$

其中 N 为系统的总核数；R_i为第 i 个应用测试的计算速率，单位为 Tflops；n_i为第 i 个测试使用的核数。这里规定每个测试都必须使用大规模算例，测试所用的核数可以根据计算规模进行调节，计算 SSP 时先计算出各测试单核的计算速率，求出几何平均值，再乘以机器所有核数，SSP 代表着机器对各个应用所能达到平均计算速率，用以评价计算系统的综合计算能力。

2.2.2 深度学习应用测试

2.2.2.1 测试程序

深度学习应用测试主要是通过在系统上执行深度学习领域的基准测试程序，如百度的 DeepBench，以综合评估系统的深度学习应用性能。

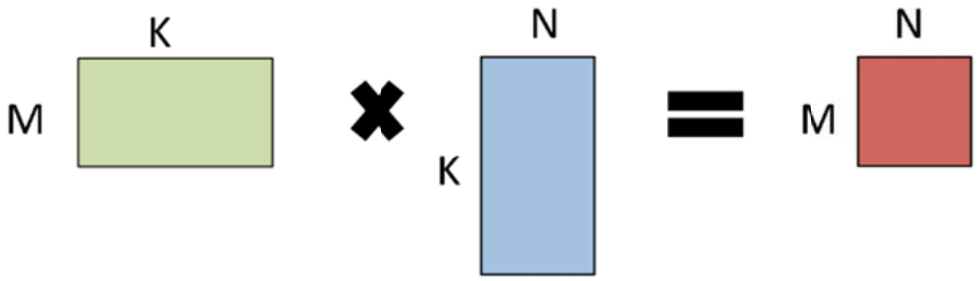
DeepBench 使用常用的神经网络库，如 cuDNN、MKL 等，测量深度学习基础运算在不同硬件上的表现。它对深度学习框架和应用性的深度学习模型不起作用，也无法用于测量训练整个模型所需的时间。由于针对不同任务而建立的模型的性能特征彼此之间差别很大，因此 DeepBench 测试的是深度学习模型中的基础运算。测试这些运算有助于了解深度学习模型的训练和推断的瓶颈。

DeepBench 包括一系列深度学习模型中的基础操作（稠密矩阵乘法、卷积和通信）以及一些递归层类型。它按深度学习的过程可以分为训练基准测试和针对推断基准测试两类。

① 稠密矩阵乘法

稠密矩阵乘法现存于几乎所有的深度神经网络中。它们被用于实现完全连接的层和 vanilla RNN，并且还是其他类型的递归层的构建模块。有时，它们也被用来快速实现不存在自定义代码的新型图层类型。

当执行 GEMM = A * B = C 时，A 和 B 两者之一或者两者皆可被转置。描述矩阵问题的常用术语为三元组（M, N, K），它描述了所涉及矩阵的大小，并使用标志“op”来表示矩阵是否被转置。下图描述了三元组（M, N, K）如何对应相乘矩阵的大小。



两个矩阵都被转置的变量不会被用于神经网络。其他三种变量则会被神经网络使用，但是它们不需要被实现为具有这些转置描述符的 SGEMM 调用。有时，

执行就地转置，然后进行适当的乘法和逆转置，运算可能会更快。常数系数 α 和 β 都应该是 1.0，这样就不会有操作被省略掉。

② 卷积计算

卷积运算构成深度神经网络中绝大多数的图像和视频操作，构成深度神经网络中的重要组成部分，如语音和自然语言建模。因此，它是从性能角度来看最重要的一层。

卷积具有 4 或 5 维的输入和输出，从而产生了这些维度的大量可能的排序。DeepBench 基准测试的第一版只关注 NCHW 格式的性能，即数据以图像、特征图、行和列的形式被呈现。

对于不同大小的过滤器和图像来说，有许多最优的计算卷积的技术，包括：直接，基于矩阵乘法，基于 FFT 和基于 Winograd 的方法。在 DeepBench 的第一个版本并不关心不同方法的准确性，因为一般认为 32 位浮点对于每一个方法而言都足够准确。

③ 递归层

递归层通常由上述计算的一些组合以及更简单的计算组成，例如一元计算和二元计算。这些计算不是非常计算密集的，并且通常只占整个运行时间的一小部分。然而，GEMM 和卷积计算在迭代层中相对较小，因此这些较小的计算的成本可能变得很大。如果与开始计算相关的固定开销较高，则尤为如此。对于递归矩阵也可以使用其他的存储格式，因为转换成新存储格式的成本可以在递归运算的很多步骤中被分摊。如果这样做的话，转换到自定义格式和从自定义格式转换回来的时间应该被包括在总执行时间中。

这些因素在时间步长内和时间步长序列内带来了许多优化可能性，使得测量操作的原始性能不一定代表整个递归层的性能。在这个基准测试中，只关注单一迭代层，尽管如果考虑它们的堆叠，还会有更多的优化机会。

输入的计算不应该被包括在迭代层运算的执行时间内，因为它在计算中可以作为一个大乘数被实际的迭代计算所消耗。因此在下式中：$h_t = g(Wx_t + Uh_{t-1})$ 对于所有 t 计算 Wx_t 的时间不应该被包括在迭代层运算的时间内。

后向计算应该计算关于权重的更新，而不是输入。所有迭代计算都是为了计算权重更新，因此计算关于输入的更新也会掩盖了需要测量的内容。

使用三种类型的递归单元：vanilla RNN、LSTM 和 GRU。ReLU 是非线性的 vanilla RNN。LSTM 的内部非线性应该是标准计算——门单元的 sigmoid 函数和 tanh 激活函数。LSTM 不应该有窥视孔连接。GRU 的内部应该是用于复位和更新门的 sigmoid 函数。输出门非线性应该是一个 ReLU。

④ All-Reduce

现在的神经网络通常使用多个 GPU 甚至拥有多 GPU 的多个系统来进行训练。有两种主要的技术类型来实现这样的操作：同步和异步。同步技术依赖于保持模型的所有实例上的参数同步，通常通过在进行优化步骤之前确保模型的所有实例具有相同的梯度副本来实现。通常用于执行此操作的消息传递接口（MPI）原语被称为 All-Reduce。基于等级的数量、数据的大小和网络拓扑结构，实现 All-Reduce 的方法有很多。异步方法是相当多样的，因此在这个版本的基准测试中不会测试这些方法。

为了评估 All-Reduce，DeepBench 使用以下库和基准：

- NVIDIA 的 NCCL
- 俄亥俄州立大学（OSU）基准测试
- 百度的 Allreduce
- 英特尔的 MLSL

NCCL 库可以在没有 MPI（单节点）和有 MPI（多节点）的情况下构建。DeepBench 在实验中为单个节点提供了两个 NCCL 版本。对于多节点实验，DeepBench 只使用带有 MPI 的 NCCL、来自 OSU 的基准测试程序以及百度的 Allreduce 实现。DeepBench 报告从每个配置的所有实现中获得的最短延迟。

Intel(R) Machine Learning Scaling Library (Intel(R) MLSL)是一个提供深度学习所用通信模式的有效实现的库。为了评估 All-Reduce 的性能，DeepBench 使用了 OSU 的 All-Reduce 基准测试程序。

训练基准测试：

训练基准测试包括对上述所有操作的支持。

在训练深度学习模型时，大多数研究人员通常对所有计算内核使用单精度浮点数。学术研究已经证明，针对在有限数据集上训练的几种不同模型，降低精度的训练是有效的。根据经验，16 位半精度浮点数足以可靠地训练大型数据集上

的大型深度学习模型。使用半精度数进行训练可以是硬件供应商更好地利用可用的计算能力。此外，参数需要占用整个模型的总存储量的一半。

训练精度，指定训练的最低精度要求。乘法和加法的最小精度被分别设置为16 位和 32 位。所有结果将包括用于基准测试的精度。

推断基准测试：基准测试推断是一个非常具有挑战性的问题。深度学习有许多应用程序，每个应用程序都有其独特的性能特征和要求。选择收到高用户流量的基准测试应用程序。DeepBench 还包括了来自深度学习模型的内核，这些内核被用于多个不同的应用程序。

对于推断内核，DeepBench 包括与训练基准测试相同的一组操作，即矩阵乘法、卷积和迭代运算。内核与训练部分的有一些区别。

- 部署平台

现实世界中的大型应用，如图像搜索、语言翻译和语音识别通常被部署在数据中心的服务器上。客户端通过互联网发送请求，在远程服务器上进行深度学习模型的处理。远程服务器通常是由许多处理器组成的强大的机器。内存和计算能力足以支持非常大的深度学习模型。在服务器上部署模型的缺点是延迟取决于客户端和服务器之间的网络带宽。它还要求用户连接到互联网。为了解决这些问题，几个模型被部署在终端设备上。设备上的部署使深度学习模型具有更低的延迟，并且无论互联网连接如何始终可用。但是，为了适应移动和嵌入式设备的功耗和内存限制，这些模型需要更小一些。

- 推断批处理大小

为了满足用户请求的延迟要求，大多数互联网应用程序在请求到达数据中心时分别进行处理。这种简单的应用程序使用单个线程处理每个请求。但是，因为两个原因，这样的做法是低效的。因为处理器需要加载网络权重，单独处理请求使得操作带宽被限制。这导致处理器难以利用芯片上的高速缓存。其次，可以利用来分类一个请求的并行量是有限的，使其难以利用 SIMD 或多核并行性。RNN 的部署尤其具有挑战性，因为通过样本评估 RNN 取决于矩阵向量乘法，这是受限于带宽的并且难以并行化。

为了克服这些问题，DeepBench 建立了一个被称为 Batch Dispatch 的批处理调度器，该调度器在执行前向传播之前将来自用户请求的数据流组合成批处理。

在这种情况下，需要在增加批处理大小从而提高效率和延迟增加之间进行权衡。为了组合批处理而缓冲的用户请求越多，用户等待结果的时间就越长。这为可执行的批处理数量设置了障碍。

对于数据中心部署的效率和延迟，批处理请求至多设置为 4 或 5 似乎效果不错。在设备上部署的情况下，批处理大小被限制为 1。

- 推断精度

深度神经网络使用单精度或半精度浮点数进行训练。推断的精度要求明显低于训练。与浮点模型相比，几种不同的模型可以使用 8 位表示进行推断而精度损失很小或没有损失。因此，对于推断内核，DeepBench 分别指定了 8 位和 32 位乘法和加法的最小精度。所有结果将包括用于基准测试的精度。

DeepBench 使用 Gemmlowp 库来使用 ARM 处理器的 8 位输入对矩阵乘法进行基准测试。ARM Compute 库中的卷积内核用于卷积基准测试。ARM Compute 库只支持单精度卷积。ARM Compute 库不支持 RNN。

对于服务器部署，DeepBench 针对 Nvidia 的 GPU 使用 cudNN 库和 cuBLAS 库。对于 Nvdia 的 GPU，RNN 内核只支持单精度。

- 稀疏计算

稀疏神经网络是指神经网络中的大部分权重为零的网络。这些零权重不能决定神经网络的预测。稀疏神经网络减少了内存和计算占用空间，使深度学习模型可以部署在移动设备上。RNN 的推断性能主要取决于硬件的内存带宽，因为大多数工作只是在每个时间步长读取参数。从密集计算转换为稀疏计算会带来惩罚，但是如果稀疏因子足够大，那么稀疏例程所需的数据量就会变小。

数据中心使用的更强大的服务器级处理器通常可以足够快地执行推断来服务一个用户，但在数据中心中，单位价格的性能非常重要。诸如稀疏性等技术可以更快地评估模型，使得每个 GPU 可以服务更多用户，从而提高单位价格的有效性能。

DeepBench 包括稀疏矩阵向量和稀疏矩阵乘法内核。研究表明，与稠密神经网络相比，90%到 95%稀疏度的神经网络可以获得相对较好的性能。然而，稀疏矩阵乘法的当前实现针对更高的稀疏性（大约 99%或更高）进行了优化。DeepBench 使用 Eigen 库来对 ARM 设备上的稀疏计算进行基准测试。针对 GPU

的基准测试则使用 Nvidia 的 cuSparse 库。

- 测量延迟

许多推断应用程序都有实时延迟要求。例如，语音接口需要语音识别模型来返回结果给用户而没有显而易见的延迟。DeepBench 内核可以作为测量单个操作的最佳延迟的起点。但是，测量完整的系统延迟并不在这个版本的 DeepBench 的范围之内，因为它只关注基本计算而不是完整的应用程序。例如，运行在移动设备上的完整应用程序在启动时可能需要修改系统的电源状态。在另一个例子中，一个完整的服务器应用程序可能有一个明显的延迟组成，这是由用户到服务器的网络连接决定的。

2.2.2.2 评价指标

执行时间 T 为执行全部基准测试程序所消耗的时间总和，单位：ms，即

$$T = \sum T_{Ops} = T_{GEMM} + T_{Convolution} + T_{Recurrent\text{-}Ops} + T_{All\text{-}Reduce}$$

其中，T_{Ops} 为相应计算或操作所消耗的时间，T_{GEMM} 为执行稠密矩阵乘法测试的用时，单位：ms；$T_{Convolution}$ 为执行卷积计算的用时，单位：ms；$T_{Recurrent\text{-}Ops}$ 为执行递归层测试的用时，单位：ms；$T_{All\text{-}Reduce}$ 为执行 All-Reduce 测试的用时，单位：ms。为避免测试运行过程中偶发性系统波动造成的误差，可分别多次执行基础运算测试并对测试结果进行综合取数，采用其中的最小值，即 $T_{Ops} = T_{min}$。每个单项基准测试均使用相同的内核参数设置。

执行时间 T 反映了系统在执行深度学习基础计算时的快慢程度，T 的值越小则该系统的深度学习应用性能越高。

2.2.3 大数据分析应用测试

2.2.3.1 测试程序

现代科学仪器所产生的数据已经达到了前所未有的规模。并且，科学大数据工作负载的数据格式和计算行为比互联网服务的要复杂得多。这两个因素对科学数据的管理和分析提出了严峻的挑战。

对于重点为科学研究领域提供服务的高性能计算环境资源而言，其大数据分

析应用性能已成为用户关注的焦点之一。常见的大数据基准测试程序要么专注于互联网领域（如 HiBench、BigDataBench、TPC-DS 等），要么只关注某一特殊领域（如 GeneBase）。因此，需要通过一个全面的，具有代表性的科学大数据基准测试程序来评估系统对科学大数据应用的支持情况。BigDataBench-S 是一个科学大数据基准测试套件，它由成熟的大数据基准测试程序 BigDataBench 延伸而来，增加了针对科学领域典型工作负载的支持。参考 BigDataBench-S，选取了几个重要的科学领域进行评测，涉及三个科学领域：高能物理学、天文学及基因组学。

- 高能物理学

在高能物理中，原始数据来源于大型强子对撞机（LHC）的许多传感器。每年的数据量可以达到 3PB。通过分析传感器采集到的事件数据，研究人员可以找到物理理论的实验证据或探索未知的物理现象。每个事件记录都包含与事件相关的多变量值，这些值以 ROOT 格式存储。典型的高能物理数据分析过程包含几个步骤：事件数据采集，实验相关事件过滤，特定实验事件分析和最终结果提取。LHC 的传感器负责事件收集，数据过滤过程需要将特定的事件与其他事件分开，以供日后分析。数据大小总是可以降低到单个机器在过滤后可以处理的程度。为了实现高效、准确的时间判别（即将某个具体研究所需要的事件与其他事件分开），需要在这个阶段进行大量基于规则的事件选择和分类操作，这是整个流程中最复杂的阶段。因此，BigDataBench-S 在构建高能物理基准时主要考虑这个阶段。

与特定物理实验相关的事件被称为实验信号事件（信号），其他不相关的事件被称为背景事件（背景）。在数据过滤过程中，研究人员首先给出简单的过滤条件，使用数据库查询操作来选择事件数据，然后使用复杂的分析算法来识别特定的信号事件。算法涉及分类和回归的典型操作，包括最大似然估计、函数判别分析、增强决策树及支持向量机等。

- 天文学

天文学的科学家们试图探索宇宙结构，并了解宇宙和不同天体如何演化。原始数据主要来自大型综合观测望远镜（LSST）等观测设备。收集的数据是不同时间段的图像。对于每一个时期，图像代表了这个时刻的宇宙观测。每个图像都以二维数组的形式存储。此外，一个完整的图像是由许多传感器的小图像组成，每个小图像也以二维数组的形式记录每个像素的信息。

在一个典型的天文图像分析场景中，原始图像首先被加工成大量的观测数据，然后通过分类和轨迹分析等多种分析操作进行处理。

- 基因组学

基因组学的目标是研究生命的基因组，分析功能并思考如何使用它们。基因组研究依赖于大量的基因组数据。每年收集的基因数据量大于 15PB。此外，随着基因收集技术成本的不断降低，基因序列数据量将以更快的速度增长。

DNA 微阵列可以在一次实验中同时测量数千个基因的表达数据，是当前基因组研究的主要技术之一。微阵列数据的收集依赖于含有大量互补的脱氧核糖核酸（cDNA）或寡核苷酸探针的基因芯片。该基因的表达数据通过荧光素标记技术得到，可以反映各基因的活性信息，为研究细胞的生理状态提供数据。

在实际的研究场景中，如疾病与基因的对应关系，生物学家从大量患者中收集了数千个基因样本，形成了以基因为列，以患者样本为行的大规模密集型浮点矩阵。生物学家可以对这个矩阵结构数据进行数据选择查询、统计、复杂分析和其他操作。

基于以上数据模型的分析和各个科学领域的典型应用构建。所选择的数据集包括高能物理的 ATLAS 数据集，使用 SS-DB 和 GenBase 数据生成器的天文和基因组模拟数据集。这三个科学领域的工作负载包含 17 个典型的操作。来自天文学的交叉操作是轨迹分析的常用操作。Sigma-Clipping 是天文学中常见的图像处理操作。不同于 GenBase 和其他现有的科学基准测试，每个工作负载都是数据操作查询或复杂的分析。这种设计便于不同大数据系统组件之间一一对应的比较。数据处理应用的研究领域、数据集及工作负载实现等细节如表 1.3 所示。

表 1.3 数据处理应用的数据集和工作负载

研究领域	数据集	工作负载	
高能物理学	ATLAS 数据集	数据操作查询	选择：根据过滤条件选择事件
		分类	SVM
			k-近邻
			LDA
		回归	增强决策树

			最大似然估计
天文学	SS-DB 生成器生成的模拟数据集	数据操作查询	选择：在给定的时间和空间范围内选择图像
			聚合：计算图像单元格的平均值以找到平均背景噪声
			联合：使用相同位置上的每个单元的平均值来联合每个图像单元以执行共添加图像处理
		复杂的分析	图像的交集
			Sigma-裁剪
基因组学	GenBase 生成器生成的模拟数据集	数据操作查询	选择：根据过滤条件选择基因
			聚合：计算基因的平均表达值
			联合：联合具有基因本体的基因
		复杂的分析	QR 分解
			奇异值分解
			协方差

2.2.3.2 评价指标

① 文件格式压缩效率

测试数据表所占用的存储空间，单位：GB。文件格式压缩效率反映了数据

格式的序列化和压缩效率。

② 数据查询性能

测试在相同的数据格式和相同的数据规模条件下系统执行选择查询、聚合查询及联合查询的性能，评价指标为相应查询操作的响应时间，单位：s。

③ 复杂分析性能

测试在相同数据格式和相同的数据规模条件下系统执行如协方差、奇异值分解及 QR 分解等复杂分析的性能，评价指标为相应复杂分析的执行时间，单位：s。

3、资源分级和准入标准

本部分对资源分级和准入标准进行说明。

3.1 资源分级标准

国家高性能计算环境资源的分级管理，不仅对资源的建设和发展有着重要的意义，还能够对使用资源的用户起到积极的引导作用，更有利于资源的全局统筹规划。为进一步加强资源管理，促进资源服务质量的持续提高，定期进行资源各项指标的评价工作，并制定资源分级标准。

由于建设思路和主要服务对象的不同，资源与资源之间可能存在着比较大的差异。有的资源侧重于性能，能够为用户提供较高的计算、存储、网络及应用性能，但在可用性、可靠性等其他指标方面则稍弱一些；有的资源侧重于高可用性或高可靠性，能够为用户提供稳定持续的服务，但其性能指标并不十分突出；有的资源则比较均衡，各项评价指标都处在很高的水平，综合能力很强。针对资源存在这样的特异性，可以分别根据各项评价指标来对资源进行分级，有利于保留资源本身独有的特点，并充分发挥资源在某些方面的优势。此外，还可以参考综合评价体系，全面考虑各项评价指标来对资源进行综合指标分级。

此外，为了保证资源分级评定工作的客观和公正，可以成立专门的资源分级评审小组，定期或不定期对资源进行分级考核，淘汰无法通过 III 级评定的资源。

3.1.1 按性能指标分级

如前文所述，性能指标分为基础性能指标和应用性能指标。基础性能指标包括了计算性能、存储性能及网络性能等三个方面，反映的是系统的硬件性能，是资源为用户提供服务的基础。通过一系列测试可以分别得出资源在计算、存储及网络方面的评价指标，分别将这些指标通过标准化计算转换为资源在计算、存储及网络等三个方面对应的评分。针对这三个评分进行算数平均，可以得出资源基础性能的综合评分，即

$$R_{Basic} = \frac{R_{Compute} + R_{Storage} + R_{Network}}{3}$$

其中，R_{Basic} 为资源的基础性能综合评分，$R_{Compute}$ 为资源的计算性能评分，

$R_{Storage}$ 为资源的存储性能评分，$R_{Network}$ 为资源的网络性能评分

根据这个综合评分来按资源的基础性能指标对其进行分级：

- I 级资源：基础性能综合评分达到 90 分及以上的资源
- II 级资源：基础性能综合评分达到 80 分~90 分（不含 90 分）的资源
- III 级资源：基础性能综合评分达到 70 分~80 分（不含 80 分）的资源

应用性能指标包括了高性能计算应用性能、深度学习应用性能及大数据分析应用性能等三个方面，反映的是资源的软件性能，是资源在实际生产中能力的体现。通过执行相关领域的基准测试程序，如高性能计算的一些小程序、深度学习基准测试程序及科学大数据分析基准测试程序等，可以分别得出系统执行相关应用领域的应用性能评价指标，分别将这些指标通过标准化计算转换为资源在高性能计算应用、深度学习应用及科学大数据分析应用等三个领域对应的评分。针对这三个评分进行算术平均，可以得出资源应用性能的综合评分，即

$$R_{Application} = \frac{R_{HPC} + R_{DL} + R_{BigData}}{3}$$

其中，$R_{Application}$ 为资源的应用性能综合评分，R_{HPC} 为资源的高性能计算应用性能评分，R_{DL} 为资源的深度学习应用性能评分，$R_{BigData}$ 为资源的科学大数据分析应用性能评分

根据这个综合评分来按资源的应用性能指标对其进行分级：

- I 级资源：应用性能综合评分达到 90 分及以上的资源
- II 级资源：应用性能综合评分达到 80 分~90 分（不含 90 分）的资源
- III 级资源：应用性能综合评分达到 70 分~80 分（不含 80 分）的资源

3.1.2 按可用性指标分级

可用性指标反映了资源投入使用的实际效能，站在用户的角度，能否使用资源完成工作，完成工作的效率及主观感受如何，是资源可用性最直观的体现。系统接入时间和系统运行率是评价资源可用性的重要依据。根据资源提交的系统接入时间可以计算得出资源的系统运行率，根据系统接入时间及系统运行率等信息，可以得出针对系统可用性的评价指标。将这个指标通过标准化计算转换为资源可用性的综合评分。

根据这个综合评分来按资源的可用性指标对其进行分级：

- I 级资源：可用性综合评分达到在 90 分及以上的资源
- II 级资源：可用性综合评分达到 80 分~90 分（不含 90 分）的资源
- III 级资源：可用性综合评分达到 70 分~80 分（不含 80 分）的资源

3.1.3 按可靠性指标分级

可靠性指标反映了资源向用户提供服务的可靠程度，体现了资源在规定的条件下和规定的时间内完成规定功能的能力。资源的故障频次、故障率、重大故障次数、最长连续运行天数及平均连续运行天数等是评价资源可靠性的重要依据。根据资源提交的上述信息，可以得出针对资源可靠性的评价指标。将这个指标通过标准化计算转换为资源可靠性的综合评分。

根据这个综合评分来按资源的可靠性指标对其进行分级：

- I 级资源：可靠性综合评分达到 90 分及以上的资源
- II 级资源：可靠性综合评分达到 80 分~90 分（不含 90 分）的资源
- III 类资源：可靠性综合评分达到 70 分~80 分（不含 80 分）的资源

3.1.4 按安全性指标分级

安全性指标反映了资源在软硬件方面的安全保障情况，分为物理基础设施的安全保障及网络和信息安全两个方面。其中，数据灾备、电力保障、基础环境设施及网络和信息安全等是评价资源安全性的重要依据。根据资源提交的是否有灾备、是否双路市电、是否有自备电源、是否有网络防火墙、是否有 VPN、是否能够 Internet 直连、是否通过（ISO27001）安全认证、是否具有保密资质、是否有内部安全制度、是否设有专职信息安全人员等信息，可以得出针对资源安全性的评价指标。将这个指标通过标准化计算转换为资源安全性的综合评分。

根据这个综合评分来按资源的安全性指标对其进行分级：

- I 类资源：安全性综合评分达到 90 分及以上的资源
- II 类资源：安全性综合评分达到 80 分~90 分（不含 90 分）的资源
- III 类资源：安全性综合评分达到 70 分~80 分（不含 80 分）的资源

3.1.5 按需求管理指标分级

需求管理指标反映了资源的管理水平，分为需求计划管理和需求响应两个方面。需求计划管理和需求响应是评价系统需求管理指标的重要依据。根据资源提供的年度经营计划、市场调研计划、软件更新计划、设备升级改造计划、上年度经营完成情况等佐证材料，以及实际记录的市场需求调研情况、软件更新列表及硬件更新列表等材料，可以得出针对资源需求管理的评价指标。将这个指标通过标准化计算转换为资源需求管理的综合评分。

根据这个综合评分来按资源的需求管理指标对其进行分级：

- I 级资源：需求管理综合评分达到 90 分及以上的资源
- II 级资源：需求管理综合评分达到 80 分~90 分（不含 90 分）的资源
- III 级资源：需求管理综合评分达到 70 分~80 分（不含 80 分）的资源

3.1.6 按技术支持与服务响应指标分级

技术支持与服务响应指标反映了资源提供技术服务的能力、水平和质量，分为服务能力、资料更新、技术培训、故障响应及服务评价等五个方面。技术支持服务制度、技术支持人员数量、技术支持方式、资料更新情况、技术培训情况、故障响应情况、用户投诉人次及用户满意度等情况是评价资源技术支持与服务响应指标的重要依据。根据资源提供的技术支持服务台制度及用户满意度等佐证材料，以及记录的技术支持人员数量、技术支持方式、技术支持资料、技术培训资料、故障响应情况及用户投诉人次等信息，可以得出资源的技术支持与服务响应的评价指标。将这个指标通过标准化计算转换为资源技术支持与服务响应的综合评分。

根据这个综合评分来按资源的技术支持与服务响应指标对其进行分级：

- I 级资源：技术支持与服务响应综合评分达到 90 分及以上的资源
- II 级资源：技术支持与服务响应综合评分达到 80 分~90 分（不含 90 分）的资源
- III 级资源：技术支持与服务响应综合评分达到 70 分~80 分（不含 80 分）的资源

3.1.7 按综合评价指标分级

综合评价指标反映了资源在上述各个方面的整体综合实力，是对资源进行分级的重要指标参考。参考综合评价体系，可以根据各项评价指标对于资源评价的重要程度，引入相应的影响因子。这些影响因子可以根据实际需要灵活地进行修改，以满足不同情况下的分级要求。将上述所有的指标的综合评分进行加权求和，可以得出资源整体的综合评分，即

$$\begin{aligned} R_{Overall} = {} & W_{Basic} \cdot R_{Basic} + W_{Application} \cdot R_{Application} + W_{Availability} \cdot R_{Availability} \\ & + W_{Reliability} \cdot R_{Reliability} + W_{Security} \cdot R_{Security} + W_{Management} \\ & \cdot R_{Management} + W_{Support} \cdot R_{Support} \end{aligned}$$

其中，$R_{Overall}$ 为资源整体的综合评分，W_{Basic} 为资源基础性能综合评分的影响因子，R_{Basic} 为资源基础性能的综合评分，$W_{Availability}$ 为资源可用性综合评分的影响因子，$R_{Availability}$ 为资源可用性的综合评分，$W_{Reliability}$ 为资源可靠性综合评分的影响因子，$R_{Reliability}$ 为资源可靠性的综合评分，$W_{Security}$ 为资源安全性综合评分的影响因子，$R_{Security}$ 为资源安全性的综合评分，$W_{Management}$ 为资源需求管理综合评分的影响因子，$R_{Management}$ 为资源需求管理的综合评分，$W_{Support}$ 为资源技术支持与服务响应综合评分的影响因子，$R_{Support}$ 为资源技术支持与服务响应的综合评分

根据这个综合评分来按资源的综合评价指标对其进行分级：

- I 级资源：评分达到 90 分及以上的资源
- II 级资源：评分达到 80 分~90 分（不含 90 分）的资源
- III 级资源：评分达到 70 分~80 分（不含 80 分）的资源

3.2 资源准入标准

近年来，我国逐渐拓宽了超级计算机的应用领域，从国家安全、核武器研制、气象预报、石油勘探这样的国家战略领域，拓展到互联网、大数据、人工智能、基因测序、影视制作、金融等领域，惠及各个不同的行业，越来越贴近国民经济生活。超级计算机不再只为少数的几个国家超级计算中心所独有，越来越多的科

研院所和企业也纷纷加入到自行研制组建超级计算机的行列中。国内高校、科研机构及企业共同合作研发和组建的超级计算机系统正逐步助力科研和相关领域产业的发展，形成了以国家超级计算中心为点带动多个领域全面发展的大好局面。

为进一步促进国家高性能计算环境资源的发展，吸引更多新鲜血液的加入，壮大高性能计算队伍，综合考虑前文所述的各项资源评价指标，制定本准入标准。

为了保证资源准入评审工作的客观和公正，可以成立专门的资源准入评审小组，定期或不定期地对待准入和已准入的资源进行准入考核，将达到准入标准的待准入资源加入到队伍中来，将达不到准入标准的已准入资源淘汰出去。只有通过这样的自我审核和自我淘汰机制，才能让国家高性能计算环境资源与时俱进，焕发出新时代的活力与朝气，更好地为用户提供稳定的高质量服务。

3.2.1 资源的性能

性能指标分为基础性能指标和应用性能指标。基础性能指标包括计算性能、存储性能及网络性能三个方面，资源的基础性能准入标准如下：

- 待准入的资源在以上三个方面的性能均不得低于已准入资源中的最低者
- 待准入的资源通过基础性能分级考评且被评定为 III 级或以上资源

应用性能指标分为高性能计算应用性能指标、深度学习应用性能指标及大数据分析应用性能指标等三个不同的应用领域。资源的应用性能准入标准如下：

- 待准入的资源在以上任意一个应用领域的性能综合评分不得低于已准入资源中的最低者
- 待准入的资源通过应用性能分级考评且被评定为 III 级或以上资源

3.2.2 资源的可用性

可用性指标包括软件资源配置和资源服务两个方面。待准入的资源提供的软件资源必须能够满足用户的最低基本需求，并且鼓励待准入的资源在有条件的情况下合理增加软件资源配置，提高自身软实力。资源的可用性准入标准如下：

- 待准入资源的系统接入时间不得低于指定的最低值。系统接入时间的最低值可以根据实际情况进行适当的调整

- 待准入的资源通过可用性分级考评且被评定为 III 级或以上资源

3.2.3 资源的可靠性

可靠性指标包括故障频次、故障率、重大故障次数、最长连续运行天数及平均连续运行天数等五个方面。综合考虑资源的总体故障率和重大故障次数，资源的可靠性准入标准如下：

- 待准入资源的总体故障率及重大故障次数均不高于国家高性能计算环境资源中的最高者
- 待准入的资源通过可靠性分级考评且被评为 III 级或以上资源

3.2.4 资源的安全性

安全性指标包括基础设施的安全保障及网络和信息安全两个方面。综合考虑待准入的资源在数据灾备、系统电力、基础环境设施、网络安全及信息安全等方面是否落实了保障工作。资源的安全性准入标准如下：

- 待准入的资源通过安全性分级考评且被评为 III 级或以上资源

3.2.5 资源的需求管理

需求管理指标包括需求计划管理及需求响应两个方面。综合考虑待准入的资源在需求管理上所做的相关工作，如是否有提交年度经营计划、市场调研计划、软件更新计划及设备升级改造计划等佐证材料，是否有积极开展市场需求调研、上年度经营完成情况调研及软硬件的更新工作等。资源的需求管理准入标准如下：

- 待准入的资源通过需求管理分级考评且被评为 III 级或以上资源

3.2.6 资源的技术支持与服务响应

技术支持与服务响应指标包括服务能力、资料更新、技术培训、故障响应及服务评价等五个方面。资源的技术支持与服务响应准入标准如下：

- 待准入的资源通过技术支持与服务响应分级考评且被评为 III 级或以上资源

3.2.7 资源的综合评价

综合评价是结合上述的所有评价指标得出的对资源总体的评价。资源的综合评价准入标准满足下面条件之一如下：

- 待准入的资源通过综合评价分级考评且被评为 III 级或以上资源；
- 在性能指标或应用性能指标上评级为 I 级的资源。