

# 集群操作系统的发展与展望

詹剑锋 王磊 涂碧波

## 引言

当前高性能计算机的体系结构主要有集群(Cluster<sup>1</sup>)、大规模并行处理系统(MPP<sup>2</sup>)、星群(Constellations)和向量机等几种类型。其中,集群已逐渐成为市场主流。在世界高性能计算机Top500排名中,无论是系统数目还是性能比重,集群所占份额都在逐年上升(如图1-a和图1-b所示)。在2008年6月份发布的世界高性能计算机

Top500中,集群体系结构占了总系统数目的80%(见表1)。

在过去的十几年中,无论是高端科学计算,还是网络信息服务,集群系统已经成为主流计算平台。近年来随着网络服务业(如谷歌、腾讯和百度)的兴起,低成本的商品化集群(Commodity Cluster)及其系统软件再次成为工业界和学术界关注的焦点<sup>[2][3][4]</sup>。在这种背景下,集群操作系统已经成为独立于硬件平台的重要基础软件。

传统的单一系统映像的集群操作系统透明地实现了资源管理和进程管理,并且为用户提供了交互界面。文献[5]认为单一系统映像的集群操作系统应实现以下主要功能:单一进程空间、单一I/O空间、单一作业管理系统、单一控制点和统一用户界面。单一系统映像的集群操作系统通过扩展节点操作系统内核来实现,其开发成本高、可移植性差。虽然杜克大学(Duke University)的杰夫·蔡斯(Jeff Chase)教授等人<sup>[6]</sup>扩展了集群操作系统的含义,但并未显式地给出其定义。在本文中我们将集群操作系统定义

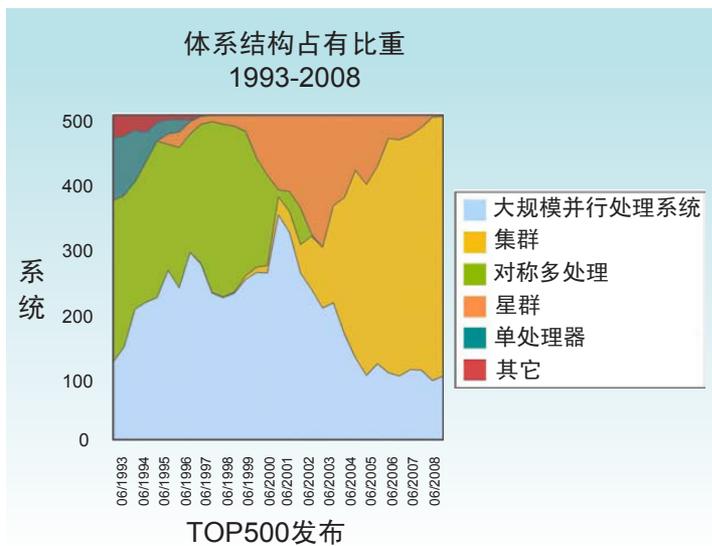


图1-a 世界高性能计算机Top500集群系统数目所占比重

<sup>1</sup> 世界高性能计算机Top 500的主要组织者Jack Dongarra<sup>[1]</sup>将集群系统定义为由独立的节点集成在一起构成的并行计算机系统:集群的每个节点能够独立操作,并且可以单独面向市场开发;集成节点的网络可以是商品化的,也可以是专门设计的。

<sup>2</sup> Massively Parallel Processing.

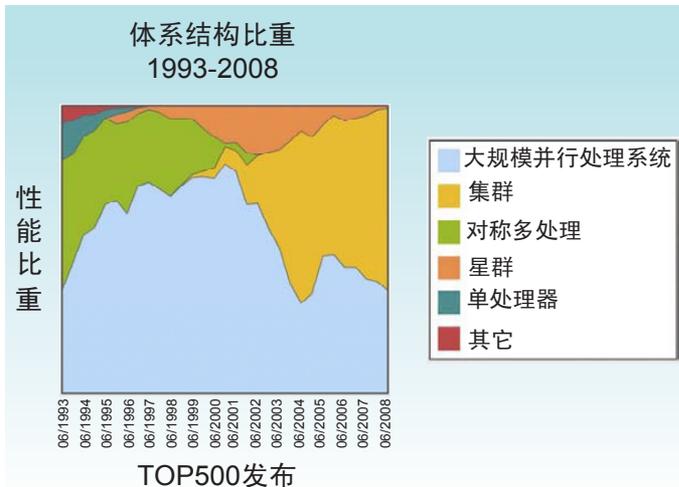


图1-b 世界高性能计算机Top 500集群系统性能所占比重

为：实现统一的资源管理和进程管理并且具有统一用户界面的集群系统软件。它可以通过扩展操作系统内核、定制系统软件或者基于中间件等不同途径来实现。

表1 2008年6月发布的世界高性能计算机Top500不同体系结构份额分布

体系结构	数量	占有比重 %	运算能力总和 (GF)
星群	2	0.40 %	94970
大规模并行处理系统	98	18.60 %	4166875
集群	400	80.00 %	7438173
总数	500	100.00%	11700018

## 具有代表性的集群操作系统

从实现方法的角度可以将集群操作系统分为三类：通过扩展操作系统内核实现的单一系统映像集群操作系统、通过定制节点操作系统和系统软件实现的定制集群操作系统以及基于中间件实现的“集群操作系统”。在本节，我们将逐一介绍三类集群操作系统的代表性成果。

## 单一系统映像集群操作系统

代表性的单一系统映像集群操作系统包括：Kerrighed<sup>[7]</sup>、OpenMosix<sup>[8]</sup>和OpenSSI<sup>[8]</sup>等。

Kerrighed是法国国家信息与自动化研究所（INRIA<sup>3</sup>）开发的项目，其目的是在集群系统上提供类似对称多处理（SMP<sup>4</sup>）系统。Kerrighed由负责全局资源管理的核心态分布式服务组成，提供可配置的全局进程调度器，其默认的调度器策略为采用接收端驱动的抢先式进程迁移方案。当系统探测到负载不平衡时，进程将从高负载的节点向低负载的节点迁移。Kerrighed的进程迁移机制是基于进程克隆（Ghosting）、容器（Container）、可迁移的流（Stream）以及分布式文件系统实现的。

OpenMOSIX是在MOSIX<sup>5</sup>项目<sup>[9]</sup>基础上开发的。MOSIX采用了发送者驱动的抢先式进程迁移策略。MOSIX迁移机制基于代理进程（Deputy）。当一个进程从源节点迁移到目标节点时，对其他进程和文件系统的依赖将在源节点上通过代理进程维护。代理进程支持被迁移进程的系统调用，包括网络通讯和文件访问。全局的内存管理通过内存导引（Memory Ushering）算法实现。当节点的空闲内存低于阈值时，该算法得到激活，从而将进程迁移到有足够空闲内存的其他节点上。

OpenSSI<sup>6</sup>项目<sup>[8]</sup>开始于2001年，前身是基于Unixware的动能集群（Nonstop cluster for Unixware<sup>7</sup>）项目。OpenSSI的设计目标是提供集成其他开源集群技术的平台。当前的OpenSSI集成了开源的文件系统和磁盘管理

<sup>3</sup> Institut National de Recherche en Informatique et Automatique（法文）。

<sup>4</sup> Symmetrical Multi-Processing。

<sup>5</sup> MOSIX是针对LINUX内核附加的一个软件包，可以实现高计算能力的集群计算系统。

<sup>6</sup> SSI, Single System Imager, 单一系统映像。

<sup>7</sup> SCO公司开发的一种Unix操作系统。

系统（GFS<sup>8</sup>、OpenGFS、Lustre<sup>9</sup>、OCFS<sup>10</sup>和DRBD<sup>11</sup>）、分布式锁机制（OpenDLM<sup>12</sup>）和从MOSIX中衍生的进程迁移机制。

## 定制化集群操作系统

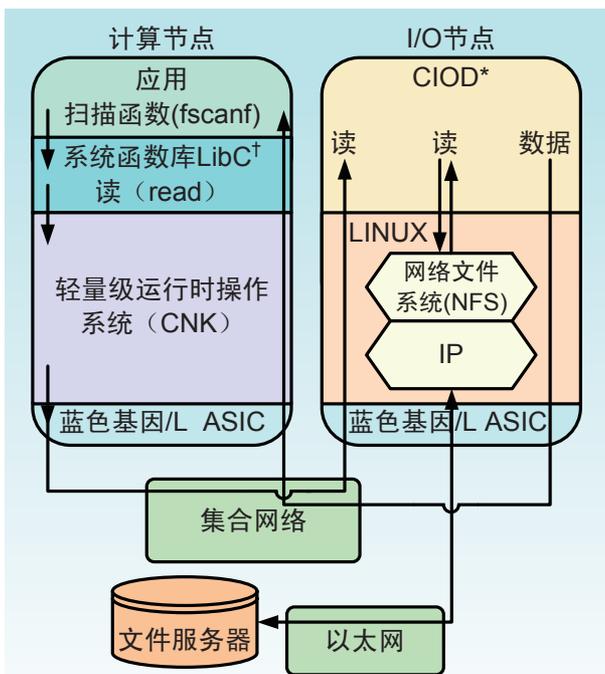
在高端科学计算领域通常采用定制化的集群操作系统。这类系统由定制化的节点操作系统和批作业管理系统组成。

通用的节点操作系统提供了丰富的功能，并且支持众多应用。但是这种通用性在高端计算领域要付出性能代价。文献[10]指出面向高性能计算的专用操作系统传输消息的性能要比Linux快4~6倍。因为应用需求、硬件和相关编程模型的多样，开发专用操作系

统的代价非常高，所以文献[11]考虑了高端科学计算的特征和需求，采用构件的方法开发专用的高端科学计算操作系统。它在分形（Fractal）通用编程模型的基础上，针对高端科学计算的需求加强和扩展了分形模型，从而使用户可以方便地根据应用需求、应用特征以及相关的编程模型构造专用的操作系统。

这方面的一个实例就是蓝色基因/L（BlueGene/L）的节点操作系统<sup>[12]</sup>可以根据用户的需求进行定制。蓝色基因/L节点操作系统的设计遵循了一个基本原则，即软件结构应该反映硬件结构。计算节点运行“轻量级运行时操作系统CNK<sup>13</sup>”。该系统只支持单个用户的单个应用程序运行在计算节点上，本身并不提供调度和进程上下文切换。而每个I/O节点则运行完整的Linux操作系统映像，包括多处理和文件系统。多个计算节点和1个I/O节点组成1个处理集合（Processing Sets, Psets）。“轻量级运行时操作系统CNK”的大部分调用直接转发给I/O节点执行，图2反映了二者之间的关系。这些简化使得系统能够充分利用硬件的特性，从而在不牺牲安全性的情况下提升系统的性能，比如严格的空同共享可允许应用使用高效的非阻塞用户态通讯，而不影响安全。每个应用线程对应1个处理器，保证了并行程序确定性的执行。

蓝色基因/L的作业管理系统<sup>[13]</sup>的优化是结合系统的体系结构进行的。为了达到更好的应用性能，用户提交作业时，需要指定期望的分区规模和形状。调度器选择合适的计算节点和I/O节点，然后通过控制网络将这些节点配置成隔离的分区。一旦创建好分区，作业就可以通过I/O节点启动。



<sup>†</sup>LibC: Libraries for C,  
<sup>\*</sup>Console I/O Daemon, 控制台输入输出后台程序

图2 CNK和I/O节点操作系统之间的关系<sup>[12]</sup>

<sup>8</sup> Google File System, 谷歌公司设计并实现的一个分布式文件系统，也是基于大量安装有Linux操作系统的普通PC构成的集群系统。

<sup>9</sup> 开放源代码的集群文件系统。

<sup>10</sup> Linux上的Oracle集群文件系统。

<sup>11</sup> 磁盘镜像技术，是由内核模块和相关脚本而构成，用以构建高可用性的集群。

<sup>12</sup> Open Distributed Lock Manager。

<sup>13</sup> Custom solution。

## 基于中间件的“集群操作系统”

基于中间件的集群操作系统可以根据实现方法分为集成化、构件化和服务化三类。

很多研究组和开源项目开发了独立的集群系统软件，比如监控系统Ganglia<sup>14</sup>、构造工具Rock和作业管理系统OpenPBS。海洋（Océano）<sup>14</sup>项目是面向服务器中心（Server Farm）的可扩展管理基础设施，也是效用计算（Utility Computing）的先驱。很多研究组集成开源的集群软件构造集群软件包，比如开放源码的集群应用资源（OSCAR<sup>15</sup>）项目、泰国农业大学（Kasetsart University）的可扩展集群环境（SCE<sup>16</sup>）以及日本真实世界计算项目的Score。在集成化的集群操作系统中，最有代表性的系统是OSCAR，其目的是将最好的集群实践集成在一个软件包里。但这些不同的集群软件之间大部分存在功能冗余，并且不同的子系统提供的信息也可能不一致。

在面向高端科学的计算领域中，来自美国国家实验室的一些课题组采用构件技术来开发万亿次高性能计算的管理软件，其代表性计划是SSS<sup>17</sup>项目<sup>[15]</sup>。其目的是针对缺少高效的万亿次计算资源的管理系统软件，开发与机器无关的面向科学计算的可扩展系统软件。

凤凰（Phoenix）系统<sup>[16][19][20][25]</sup>和其前身Clustone<sup>[17]</sup>是第一个基于服务的集群操作系统，由中国科学院计算技术研究所国家智能中心开发。其基本思路是针对科学计算和信息服务的公共需求，开发公共服务——所谓“集群操作系统核心”<sup>[19][20]</sup>；在集群操作系统核心内部解决管理系统的可扩展和容错；基于集群操作系统核心，开发面向不同用户需求的集群使用环境和编程环境。目前，基

于“凤凰”集群操作系统核心开发的集群使用环境包括：集群监控、集群配置和部署系统、批处理系统、信息服务运行时环境以及 workflow 应用调度系统等。

## 需求驱动的集群操作系统发展和展望

毋庸置疑，无论是高端的科学计算，还是网络信息服务，集群系统已经成为主流的计算平台。在新的用户需求驱动下，集群操作系统开始逐步与硬件分离，成为重要的基础平台软件，并且在不断发展和演化。对这一趋势最好的见证是云计算（Cloud Computing）下的集群系统软件的发展。在2006年前后，学术界普遍认为商用化集群系统已成为大路货，集群系统软件已成为一个成熟的领域（Mature Field）。而在同一时期，以类谷歌计算（Google-like computing）为代表的海量数据处理向集群系统软件提出了新的挑战需求。谷歌公司在满足这些需求的基础上，基于低成本的商用化集群系统发展了以谷歌文件系统、MapReduce<sup>18</sup>和BigTable<sup>19</sup>为代表的核心集群系统软件技术，为满足其持续增长的业务需求提供了强有力的保障。这些成果均发表在计算机领域的顶级会议SOSP<sup>20</sup>和OSDI<sup>21</sup>上。卡内基·梅隆大学（CMU<sup>22</sup>）计算机学院院长兰德尔·布赖恩特（Randal E. Bryant）认为这些工作表明在云计算领域，工业界领先学术界。大卫·帕特森（David Patterson）在《美国计算机学会通讯》（Communication of ACM）上专门为MapReduce技术撰写了“数据中心是计算机

<sup>14</sup> 神经中枢。

<sup>15</sup> Open Source of Cluster Application Resource。

<sup>16</sup> Scalable Cluster Environment。

<sup>17</sup> System Support Station。

<sup>18</sup> Google开发的C++编程工具，用于大规模数据集（大于1TB）的并行运算。

<sup>19</sup> Google内部开发的一个用来处理大数据量的系统。

<sup>20</sup> Symposium on Operating Systems Principles，是ACM的会议，每两年一届。

<sup>21</sup> Operating Systems Design and Implementation，每两年一届。

<sup>22</sup> Carnegie Mellon University。

(Data center is computer)” 的评论。

在下文中，我们将从用户需求的角度出发阐述集群操作系统可能的发展和演化。

## 集群系统用户需求的维度

新的用户需求将是集群操作系统发展的重要推动力。集群系统的用户需求可以从以下几个维度进行刻画：使用模型（usage model）、资源使用方式、用户的计算机知识层次和应用特征。

我们将使用模型定义为集群系统用户需求的X维度。集群系统主要有两种使用模型，即面向能力（Capability）和面向容量（Capacity）<sup>[11]</sup>。面向能力的使用模型有以下典型特征：定制精简的节点操作系统，范围有限的应用种类，每个应用可以使用完整规模的系统，允许程序员充分利用系统特性来精细地优化和微调代码。面向能力的使用模型可以进一步细化为专用（Dedicated）或者空间共享（Space-Sharing）的使用模型。面向容量的使用模型可支持更广泛的应用和更为灵活的分时共享，因其采用通用的操作系统，所以可提供更复杂的服务，如动态加载和共享库等。

资源使用方式定义为集群系统用户需求的Y维度，主要分为独占和共享两种。用户的计算机知识层次可定义为集群系统用户需求的Z维度，分为计算机专业用户和非计算机专业用户两大类。计算机专业用户以掌握高深而复杂的计算机技术为荣，而非计算机专业用户一般缺少足够计算机基础知识，因而难以了解底层的系统知识。我们将应用特征主

要分为科学计算和信息服务两大类，并且将其定义为C维度。

我们利用这些坐标维度来刻画不同的集群系统用户需求，并且在特定的需求背景下讨论集群操作系统的发展趋势。

## 集群操作系统展望

### 高端科学计算（High End Computing）

在高端科学计算领域，单个并行应用的可扩展性和性能应该成为主要的追求目标，因而所关注的内容应是面向能力和资源独占的高端集群系统。在这种背景下，系统对应用的适应以及系统资源的异构性可能会成为主要的趋势。例如，克雷（Cray）将自己的高效能计算机计划定义为适应性超级计算（Adaptive Supercomputing）。以地球科学为例，用户希望进行海洋、大气、生物圈和地球物理耦合的气候模型仿真<sup>[18]</sup>。当前这些计算模型仅仅在单一种类的处理器的结构上运行（如标量或向量）。但在气候仿真模拟中，部分代码本质上是串行的，这类串行代码在超威（AMD）的皓龙（Opteron）处理器上可以获得很好的性能。而其它大部分代码则可以很好地向量化，如动力学和辐射物理，可以运行在向量模态的多线程/向量处理（Multi-threaded /Vector Processing, MVP）加速器上。而云物理学则不能向量化，其大部分为分支和条件语句，适宜于运行在多线程的MVP加速器上。挑战性应用要想获得好的性能，需要能适应应用的异构资源。加州大学伯克利分校的科学家们也在文献[21]中，从性能加速比的角度阐述了采用异构资源提升挑战性应用性能的理由。

在高端科学计算领域，集群操作系统应遵循简单和更灵活（Simple & Smarter, SS）原则。一方面，系统要尽量简单，以降低在定制系统上开发的成本；另一方面，应用于管理异构资源的智能运行时系统又可成为一种让系统贴近应用的使能（Enabling）技术。

### 效用计算（Utility Computing）

效用计算的本质是在用户体验（服务质量）主导下的资源和软件的服务化。效用计算在使用模型上是面向容量的，并且鼓励



图3 高端科学计算与效用计算对比

共享资源。尼古拉斯·卡尔 (Nicholas Carr) 在HPCWire<sup>23</sup>上大力宣扬, 计算的本质正从独占拥有计算机和软件向效用 (Utility) 模型转化, 这种改变是一个渐进的过程。在过去几年里可以看到, 许多效用计算系统完全不同于过去自包含的计算系统。在这种演化过程中, 我们观察到三个主要的固定用户角色: 资源提供者、服务提供者和最终用户。例如作为资源提供者, 一些公司已经开始采用计算服务化 (Computing-as-a-Service) 模型, 比如升阳提供的Network.com<sup>24</sup>和亚马逊 (Amazon) 提供的弹性计算云 (elastic computing cloud, EC2) 和简单存储服务 (simple storage service, S3) 等。一些大的组织同时扮演资源提供者和服务提供者, 在本地运行数据中心的同时重建系统以聚集不同的计算负载。云计算作为效用计算的特例, 正是重构数据中心努力的代表性方向。以谷歌为例, 其自身的基础设施平台仍是商业机密, 据报道大约有45万台服务器分布在全球各地。谷歌通过定制的Linux和海量任务调度软件 (Global Work Queue) 将这些服务器联接为全球性的计算平台, 以较低的成本实现了高效的超大规模数据处理。与此同时, 一些大学正在外包高性能计算服务, 而仅仅保留作业执行服务提供者 (Job-Execution Service Provider) 的角色。大量的中小型公司为了节约成本外包计算资源, 而仅仅关注于自身提供的服务本身。

在这种背景下, 我们认为面向效应计算的集群操作系统将按照以下方向发展: (1) 服务化: 新的集群操作系统需要支持资源提供者和服务提供者角色的分离, 需要支持在线的用户使用环境的构造及生命周期管理; (2) 虚拟化: 集群操作系统应能够根据应用特性动态构造虚拟计算环境; (3) 自主管理<sup>[18]</sup>: 管理复杂性随计算规模的增长成倍增长; 集群操作系统应能自动监控各类资源, 深度挖掘、分析、关联系统行为、预测系统运行趋势。集群操作系统应该在保证系统性能的情

况下, 通过自优化、自愈合和自配置等核心技术降低资源消耗、功耗和管理成本; (4) 以应用为中心: 集群操作系统能够提供保证应用服务质量的综合手段; 能够结合应用流程、数据依赖关系和资源消耗特征提供优化的运行时系统。

### 个人超级计算

随着多核系统的出现, 并行计算已经无处不在。英特尔负责研究的副总裁安德鲁·钱 (Andrew A. Chien) 将其称为普适并行计算 (pervasive parallel computing)<sup>[22]</sup>。集群因为其良好的可扩展性、可用性和高性价比, 将成为普及超级计算的主要平台。我们认为普及超级计算的主要目标用户应是各专业领域非计算机专业的工程和科研人员。个人高性能计算机<sup>[23]</sup>和个人并行托管环境 (Personal parallel Hosting Environment)<sup>[24]</sup>将是普及个人超级计算的两种主要方式。它们都面向非计算机专业用户, 需要支持的应用范围广, 使用模型都是面向容量 (Capacity) 的。个人高性能计算机的资源使用方式是独占的 (如图4所示), 而个人并行托管环境的本质则是在共享的集群平台上为非计算机专业用户提供定制的并行计算环境。

按照高性能计算普及方法的不同方式, 集群操作系统所关注的核心技术也不同。例如, 面向个人高性能计算机的集群操作系统需要解决管理的复杂性, 因而零配置、零管理、自恢复、降低认知困难的抽象及图形界



图4 个人高性能计算机和个人并行托管环境

<sup>23</sup> 是在高性能计算 (HPC) 产业中受到广泛认可、人气最旺的新闻和信息网站。

<sup>24</sup> 升阳公司的在线计算网站Network.com。

面是关键的核心技术。个人并行托管环境作为效用计算的一种特殊形式，托管环境提供者将负责系统的配置和管理。个人用户只需要关注编程，因而其关键技术在于托管环境的生命周期管理、动态资源供应、服务质量保证、资源计费和系统自愈合等技术。

## 结语

本文将集群操作系统定义为实现统一资源管理和进程管理并且具有统一用户界面的集群系统软件。在对相关工作进行调研的基础上，我们从实现方法角度将集群操作系统分为单一系统映像集群操作系统、定制化集群操作系统和基于中间件的“集群操作系

统”三类，并针对每一种集群操作系统，综述了其代表性的成果。

计算机作为一门实践的学科，需求驱动起着至关重要的作用，集群操作系统领域也不例外。本文通过定义和利用集群系统用户需求的四个维度：使用模型、资源使用方式、用户的计算机知识层次和应用特征，刻画了集群系统用户需求，并且在不同的需求背景下讨论了集群操作系统的发展趋势。在新的用户需求驱动下，集群操作系统开始逐步与硬件分离，成为重要的基础平台软件，并且将不断发展和演化。此外，本文作者还基于自身的判断阐述了集群操作系统在高端科学计算、效用计算和个人超级计算等领域的发展趋势和核心技术。■



詹剑锋

中国计算机学会高级会员。博士，中国科学院计算技术研究所副研究员。研究方向为集群操作系统和可信高性能计算。



王磊

中国科学院计算技术研究所助理研究员。研究方向为效用计算。



涂碧波

中国科学院计算技术研究所助理研究员。研究方向为高效能并行计算。

## 参考文献

- [1] Jack Dongarra, Thomas Sterling, Horst Simon, Erich Strohmaier, High-Performance Computing: Clusters, Constellations, MPPs, and Future Directions, Computing in Science and Engineering, vol. 7 no. 2, pp. 51-59, Mar/Apr, 2005
- [2] Sanjay Ghemawat etc , The Google file system, SOSP 2003
- [3] Jeffrey Dean etc , MapReduce: Simplified Data Processing on Large Clusters, Sanjay Ghemawat, OSDI 2004
- [4] Fay Chang etc , Bigtable: A Distributed Storage System for Structured Data, Fay Chang, OSDI 2006
- [5] Rajkumar, Toni Cortes, Hai Jin, SINGLE SYSTEM IMAGE (SSI). International Journal of High Performance Computing Applications , Volume 15 , Issue 2 (May 2001) pp: 124 - 135
- [6] Jeffrey S. Chase etc, Dynamic Virtual Clusters in a Grid Site Manager, HPDC 03
- [7] Christine Morin etc, Kerrighed and Data Parallelism: Cluster Computing on Single System Image Operating Systems , IEEE Cluster 04
- [8] R. Lottiaux etc, OpenMosix, OpenSSI and Kerrighed: A Comparative Study, CCGrid 05

- [9] Barak A. and La'adan O., The MOSIX Multicomputer Operating System for High Performance Cluster Computing. Journal of Future Generation Computer Systems, Vol. 13, No. 4-5, pp. 361-372, March 1998.
- [10] Ron Brightwell etc, A Performance Comparison of Linux and a Lightweight Kernel. Cluster 2003
- [11] Jean-Charles Tournier, Towards a Framework for Dedicated Operating Systems Development in High-End Computing Systems, ACM SIGOPS Operating Systems Review , Volume 40 , Issue 2 (April 2006)
- [12] Jose Moreira etc, Designing a highly-scalable operating system: the blue Gene/L story, SC 06
- [13] Y. Aridor etc, Resource allocation and utilization in the Blue Gene/L supercomputer, IBM Journal of Research and Development, Vol.49, No.2/3, 2005
- [14] K. Appleby etc, Océano--SLA Based Management of a Computing Utility, Proceedings of the 7th IFIP/IEEE International Symposium on Integrated Network Management, IEEE, New York (2001)
- [15] Desai, R. Bradshaw etc, Component-Based Cluster Systems Software Architecture: A Case Study , Cluster 2004
- [16] Zhan, J., Wang, L., Tu, B., Wang, H., Zhang, Z., Jin, Y., Wen, Y., Chen, Y., Wang, P., Qiu, B., Meng, D., and Sun, N. 2007. The design methodology of Phoenix cluster system software stack. In Proceedings of the 2007 Asian Technology information Program's (Atip's) 3rd Workshop on High Performance Computing in China: Solution Approaches To Impediments For High Performance Computing (Reno, Nevada, November 11 - 11, 2007). CHINA HPC '07. ACM, New York, NY, 174-182
- [17] 孙凝晖,刘淘英. 支持网格的机群操作系统的设计. 计算机研究与发展, 2002年8月
- [18] 洪学海,詹剑锋,樊建平,张志宏. 应用驱动的高效能计算机系统的研究与发展 《计算机研究与发展》 2007.10
- [19] Jianfeng Zhan, Ninghui Sun, Fire Phoenix Cluster Operating System Kernel and its Evaluation, IEEE Cluster 2005
- [20] Jianfeng Zhan, Lai Wang etc, a layered design methodology of cluster system software stack, IEEE Cluster 2007
- [21] Krste Asanovic, Ras Bodik etc, , The Landscape of Parallel Computing Research: A View from Berkeley, Technical Report No. UCB/EECS-2006-183, December 18, 2006
- [22] Andrew A.Chien, pervasive parallel computing, keynote, PPOPP 07
- [23] 孙凝晖. 个人高性能计算机. 中国计算机学会通讯, 2006年,第2卷,第1期
- [24] Jianfeng zhan, Lei Wang etc, Taotao: facilitating hosting environment as a service, submitted to IEEE Cluster 08
- [25] 孟丹、詹剑锋等, 一体化机群操作系统Phoenix, 《计算机研究与发展》 2005.6