第 19 卷 第 1 期 上海大学学报(自然科学版) Vol. 19 No. 1
2013年2月 JOURNAL OF SHANGHAI UNIVERSITY (NATURAL SCIENCE) Feb. 2013
DOI: 10.3969/j.issn.1007-2861.2013.01.004

# CPSE-Bio: 基于云计算的生物问题求解环境

谢 江, 王旻超, 易荣贵, 夏上云, 张 武
(上海大学 计算机工程与科学学院, 上海 200444)

**摘要:** 生物信息学是结合计算机技术解决生物问题, 利用计算机能力加速生物研究的交叉性学科. 问题求解环境(problem solving environment, PSE)是一类面向科学问题求解的计算平台, 研究人员通过使用PSE可以高效地参与和开展科学研究. 由于生物数据规模通常很大, 而这些数据随着生物技术的发展仍在不断地增加, 因此, 传统的单机单系统PSE已无法满足生物计算需求. 介绍上海大学计算机工程与科学学院高性能计算研究所生物信息学研究团队将云计算技术与PSE相结合, 实现基于云环境的问题求解环境CPSE-Bio, 并对其中2个代表性模块, 即PPI(protein-protein interaction)多数据库网络查询(multi-database retrieval technology, MDRT)模块和蛋白质挖掘(protein mining, PM)模块, 进行性能分析和比较.

**关键词:** 生物信息学; 问题求解环境; 云计算
**中图分类号:** TP 39 **文献标志码:** A **文章编号:** 1007-2861(2013)01-0021-05

# CPSE-Bio: A Cloud-Based Biological Problem Solving Environment

XIE Jiang, WANG Min-chao, YI Rong-gui, XIA Shang-yun, ZHANG Wu
(School of Compute Engineering and Science, Shanghai University, Shanghai 200444, China)

**Abstract:** Bioinformatics is an interdisciplinary subject which combines biology with computer science to address biological problems. The purpose of problem solving environment (PSE) is to solve scientific problems and provide an effective platform for researchers. As the scale of biological data is huge and data increase rapidly with the development of the biology technology, it is hard for the traditional PSE based on a sequential computer system to meet the processing demand. This paper reviews the work of the Bioinformatics Group at the School of Computer Engineering and Science, Shanghai University. PSE with the cloud technology and implement a bioinformatics PSE named CPSE-Bio, based on cloud computing are combined. The performances of two main modules in the CPSE-Bio, multi-database retrieval technology (MDRT) and protein mining (PM), are evaluated and analyzed.

**Key words:** bioinformatics; problem solving environment (PSE); cloud computing

生物信息学(Bioinformatics)利用应用数学、信息学、统计学和计算机科学的方法来研究生物学问题[1], 处理和维护诸如基因序列、蛋白质序列、蛋白质交互网络(protein-protein interaction, PPI)和转录因子等多种生物数据. 相对于传统的生物学, 生物信息学利用计算机方法对数据进行检测、比对和模拟等操作, 从而提高研究效率. 自从生物信息学诞生以来, 许多研究算法得以实现和广泛应用. 这些算法覆盖了生物学研究的各种领域, 如生物聚类、网络匹配、序列比对等, 给生物研究带来突破. 然而, 这些算法都具有很强的专业性, 对于不具备专业背景的研究人员而言, 使用这些算法往往是费时费力的. 许多学者为此做了大量的工作, 其中问题求解环境是一个非常有效的手段.

问题求解环境(problem solving environment, PSE)是提供所有必需的计算组件, 用以解决某一个类别的问题, 即为使用者提供一种软件集成系统, 并针对具体问题构建数据管理系统、模型系统、可视化平台, 从而形成一种计算环境[1-3]. 自1970 年PSE 的相关研究开展

**收稿日期**: 2012-11-29
**基金项目**: 国家教育部博士点基金资助项目(20113108120022); 上海市科委重点资助项目(11510500300); 上海市重点学科建设资助项目(J50103)
**通信作者**: 谢 江(1971—), 女, 副教授, 博士, 研究方向为生物信息学、高性能计算. E-mail: jiangxsh@shu.edu.cn

以来, 越来越多的学者投入到了PSE的研究工作, 涌现出针对各个领域的PSE, 诸如NCAS[4-6], PDE, MART[7-8]和PSE Park[9]等.

随着互联网技术不断发展, 信息数据量的不断增长, PSE研究面临了新的挑战, 即大数据处理困难. 因此, 传统的单机单系统PSE逐步向多机分布式系统PSE进行过渡, 并取得了长足的发展, 如基于网格计算的PSE, WebFlow[10], Cactus[11]和GridPort[12]. 就生物技术而言, 生物技术的发展推动了生物数据的不断增加. 以蛋白质数据为例, UniProt数据库[13]目前已经包含了大约2 000万条蛋白质序列数据, 相比于2011年已增加了近2倍的数据量. 因此, 对数据的计算与维护带来了巨大的挑战. 上海大学计算机工程与科学学院高性能计算研究所生物信息学研究团队近年来对生物信息学PSE进行了大量研究, 先后实现基于网格计算的PSE-Bio[14-20]和基于云计算的CPSE-Bio[21-23], 并开发了数据管理、生物网络匹配和数据可视化等模块.

本工作将重点介绍近年来上海大学计算机工程与科学学院高性能计算研究所生物信息学研究团队对生物信息学PSE的研究工作, 在回顾了在PSE开展的相关工作后, 选取CPSE-Bio中的两个云计算支持模块、多数据库PPI网络查询(multi-database retrieval technology, MDRT)模块[22]和蛋白质挖掘(protein mining, PM)模块[23]进行分析. 最后通过实验比较, 发现MDRT模块在与云计算结合后能够在性能上得到提升, PM模块在结合了云虚拟化技术后其计算资源得到了充分的利用.

## 1 相关工作

上海大学计算机工程与科学学院生物信息学研究团队自2007年以来对生物信息学PSE进行了相关的研究. 2007年, 谢江等[17]提出了一个基于网格的生物信息学问题求解环境PSE-Bio. PSE-Bio在传统3层平台架构(Portal层、Middleware层和Resource层)上做了改进, 新增加了Agent层, 从而使得网格资源对用户具有较好的透明性. 当任务提交后, Agent层会自动部署任务执行, 而用户无需与网格进行任何交互, 只需监控任务执行状态.

2008年, 谢江等[18]又对PSE-Bio中的工作流作分析, 并对Agent层做了进一步的细化. 他们将Agent层划分为Interface Agent, Task Agent和Resource Agent. Interface Agent是管理和维持用户和平台进行交互的接口, 每一个用户的访问都对应了一个Interface Agent. Task Agent对平台中的任务进行管理, 每个任务都有一个相对应Task Agent, 用来控制它的启动、挂起、终止和迁移等操作. Resource Agent用于管理数据资源, 接收Task Agent发送的请求, 返回所需数据资源. 通过对原Agent层进一步细化, 使得平台内部资源得到了更好的封装, 也使得用户能够更有效地对资源进行监控. 为了提高平台的兼容性, 赵志康等[14]将Web Service技术应用到了PSE-Bio中. 通过利用JNI技术对服务端的事务处理逻辑进行封装, 使得多语言集成得以实现.

2009年, 毛国勇等[15]在PSE-Bio的可视化方面提出了利用SVG技术来实现数据的可视化, 分别设计了数据对象和图形对象. 数据对象是所有返回信息的封装, 存放一些重要的可视化信息, 如起始结点、终止结点、边的数目、点的数目和顶点的度等. 图形对象则是由Root对象及其子对象Node和Line组成, 3个对象分别都具有相关的属性和操作, 并通过这些属性和操作, 用户得到良好的可视化效果. 为了达到更好的交互效果, 毛国勇等[16,19]提出将SVG和AJAX技术相结合的方法.

Cytoscape是广泛应用于生物信息学的可视化工具, 是一种对生物数据进行可视化的有效手段[24]. 俞雷等[20]对Cytoscape做了相关研究, 并开发了基于Cytoscape的插件BNMatch. BNMatch通过NBM算法[25]对2个相似的生物分子网络进行分析, 并将结果进行对比可视化, 将相匹配网络中的相似结点和相似边分别用不同的形式表现出来.

自2007年以来, 云计算[26]得到广泛关注. 云计算是基于并行计算、分布式计算和网格计算的进一步发展, 具有更好的特性和应用前景. 程笑等[21]将云计算与PSE-Bio相结合, 搭建了基于云环境下的生物信息学求解环境CPSE-Bio, 并实现了大规模生物数据管理. 通过部署基于MapReduce的计算集群, 对不同表形的生物数据进行统一和管理. 为了能更有效地管理生物数据, 易荣贵等[22]在CPSE-Bio中实现了MDRT. MDRT使用在云环境中搭建的联合数据库, 使用户可以更高效、更高质量地查询PPI数据. 虚拟化是云计算中一个重要的技术, 通过虚拟化技术, 可以更有效地利用云计算中的计算资源. 2012年, 夏上云等[23]利用云虚拟化技术, 在CPSE-Bio中通过搭建虚拟化计算平台来提高计算资源利用率. 他们选取了CPSE-Bio中的PM模块, 通过将PM模块在虚拟化前后平台上执行性能比较, 发现虚拟化技术能够有效地提高资源利用率.

## 2 MDRT模块设计分析

MDRT模块[22]是CPSE-Bio中的云计算支持模块, 其主要目的是在多个PPI网络数据库中查询所需的PPI网络数据. 目前, 互联网上已公布了超过130个PPI网络数据库[27]. 大部分的PPI数据都来源于3个途径: 高通量芯片、计算机预测和文本挖掘. 因此, 这些数据都具有许多不同的表现形式且存在大量的冗余. 为此, 许多学者尝试减少冗余和统一数据. 如PIR

(The Protein Information Resource)尝试着对蛋白质名称进行统一[28], IMEx (The International Molecular Exchange Consortium)在减少数据冗余上做了大量工作[29]. MDRT 的结构如图1所示, 其中“➡”表示工作流, “➪”表示数据库更新操作.

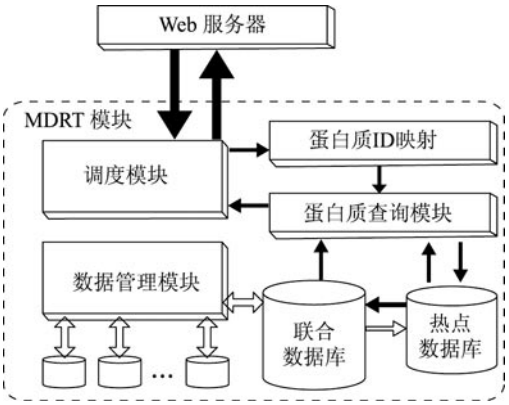


**图 1** MDRT 模块结构

**Fig. 1** Architecture of MDRT

MDRT主要由7个基本部分组成: Web Server、任务调度、蛋白质ID映射、查询、数据库管理、联合查询数据库(integrated query database, IQD)和热点查询数据库(hot spot dataset, HSD), 其中IQD 和HSD 是MDRT 的关键部分, 用于整合多个PPI 数据库, 并通过MapReduce 编程模型对PPI 数据进行查询检索. IQD 包含了大量的PPI 数据, 存储在HDFS 中, 而HSD 是一个较小的热点数据库, 用于存储查询率较高的数据. 当用户将需要查询的PPI 数据提交到服务器后, 服务器将用户提交信息分发到CPSE-Bio 的计算结点中. 然后每个计算结点首先在HSD 中进行快速查询, 如果未命中, 则在其相应的IQD 数据块中进行查询. 最后主结点对查询结果进行归约, 返回给用户.

为了构建本地的IQD, 将互联网上发布的多个PPI数据库进行整合. 对于大规模的数据量, 采用MapReduce 编程模型. 首先将互联网中的PPI 数据根据不同物种进行划分, 并针对不同的数据格式设计转换接口, 并将其转化为统一格式. 最后, 将这些数据进行归约并存储在HDFS 中.

在大部分的PPI 数据库中, PPI 数据主要存储蛋白质交互信息、蛋白质别名、探测方式和相关引用等. 对于IQD 中的PPI 数据, 其主要目的是用于查找蛋白质之间是否具有相互作用关系, 因此, 本研究对IQD 中的数据做了简化. 简化后每条数据包含3个字段: 蛋白质A、蛋白质B 和它们作用关系所在数据库的链接. 在大部分PPI 数据查询中, 研究者主要关心蛋白质之间的关系, 因此, 通过上述字段就能有效获取所需信息.

为了验证MDRT 模块的有效性, 在CPSE-Bio 中选取了3个计算结点, 并部署Hadoop 环境(版本0.20.2), 其中1个作为主结点(单核X86 处理器, 内存1 GB, Ubuntu 10.10), 另外2个作为数据结点(双核X86 处理器, 内存4 GB, Ubuntu 10.10).

将云环境平台与单机单系统平台(双核X86处理器, 4 GB内存, Ubuntu 10.10)做查询性能上的比较. 分别用多组不同大小的实验数据进行比较, 实验结果如图2所示. 可以发现, 随着计算量的增加, 云环境平台的性能优势得到体现.

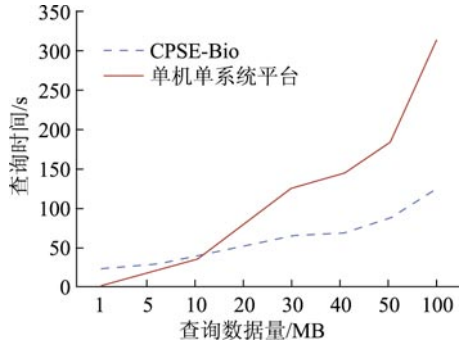


**图 2** 云环境平台与单机单系统平台查询性能比较

**Fig. 2** Comparison between cloud system and single system platforms

## 3　PM 模块与云虚拟技术

目前多核计算机被广泛应用, 使用多核计算机作为计算结点组成Hadoop 集群时, 分发给各个结点的任务执行由操作系统自行调度. 但是, 有时资源的竞争会导致计算任务在同一结点上的不同核之间进行迁移, 从而产生额外的时间花销, 造成系统性能的降低. 因此, 对于拥有多核的计算结点, 由于资源的竞争、核间的任务迁移和一些算法的特性, 导致每个核的利用率均在60%∼70%之间, 从而产生了部分的空闲[23].

对于上述问题, 本工作结合云虚拟化技术来改善资源浪费, 进一步提高计算性能. 在实验中, 将Hadoop 和Kernel-based Virtual Machine (KVM) 虚拟化技术[30-31]相结合, 将每个计算核虚拟成为一个计算结点, 将Hadoop 任务部署到每个计算核, 减少资源竞争和任务迁移所带来的额外负载. 为了验证虚拟化技术在提高计算核使用率上的有效性, 在CPSE-Bio上另外选取了3个相同架构计算结点(双核X86 处理器, 6 GB内存, Ubuntu11.04), 通过使用KVM 将其虚拟化. 虚拟化后的Hadoop 的部署结构如图3所示.

将原来的3个计算机点虚拟成为6个计算结点(见图3), 这样每个虚拟计算结点均具有单核X86处理器和3 GB内存; 然后, 对这6个新的虚拟结点进

行 Hadoop 配置, 选取其中 1 个作为 NameNode, 其余 5 个作为 DataNode. 为了验证虚拟化后计算性能的提高, 分别在虚拟化前后的不同平台上执行了相应的蛋白质挖掘模块 PM[23].

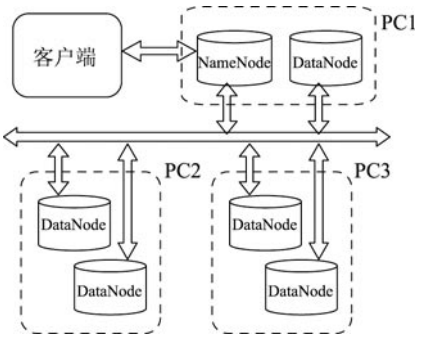


**图 3** 虚拟化后的 Hadoop 部署结构

**Fig. 3** Architecture of the Hadoop deployment on virtualized platform

PM的主要工作类似于文本字符的统计, 是对 PPI 数据库中的数据进行蛋白质个数进行统计. PM 模块是一个用于处理数据密集型的计算模块. 在任务开始阶段, 需要处理的蛋白质文本数据被分配到了各个计算结点中进行计算. 每个计算结点对所获数据进行分析与处理, 当完成数据处理后, 将计算结果返回给主结点并发送数据请求, 获取新的数据进行下一轮计算. 当所有数据计算完成后, 主结点对计算结果进行归约生成输出数据, 返回给用户.

实验结果表明, 虚拟化后的计算性能明显提高(见图 4). 在实验中, 实验数据从 100 MB 到 4 GB变化. 在数据量为 100 MB 时, 2 个平台的性能差距并不明显, 但随着数据量的增加, 性能之间的差异也越趋明显. 可以发现, 未虚拟化平台的计算核利用率保持在 80% 左右, 而虚拟化平台的计算核利用率几乎始终保持在 100%, 计算资源得到充分利用. 在另一方面, Blocksize 大小的设定也对计算性能产生影响. 只有为处理不同数据文件的程序设置合适的 Blocksize 才能有效提升计算性能. 同样地, 结点之间的通信量也很大程度上影响了运行效率, 这是由于 Hadoop 不同于传统的 MPI 并行模型, 它主要用于数据密集型的计算. 对于通信量较少、数据较为密集的应用, Hadoop 往往能够具有较好的表现. 充分利用计算资源, 虚拟化技术才能得到较高的加速比. 相反, 对于通信量较大的计算密集型应用, Hadoop 的表现往往无法达到预期效果和得到较高的加速比.

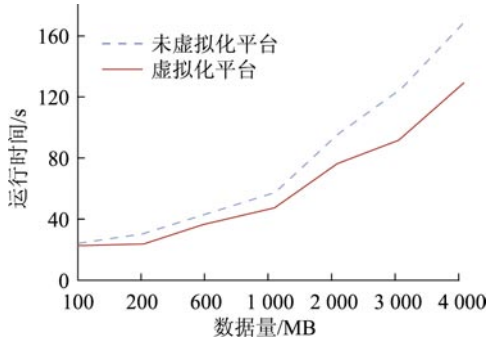


**图 4** 蛋白质挖掘程序在虚拟化前后平台的运行性能比较

**Fig. 4** Performance comparison between virtualized and non-virtualized platforms

## 4 结 束 语

问题求解环境的出现给研究人员带来了很大的便利. 从生物信息学角度来看, 生物技术的发展推动了数据量的快速增长, 由此导致了传统的单机单系统PSE已无法满足大规模问题求解需求. 上海大学计算机工程与科学学院高性能计算研究所生物信息学研究团队近年来针对生物信息学问题求解环境做了大量的工作. 自 2007 年以来提出了基于网格的生物问题求解环境 PSE-Bio, 并开发了多个计算和数据可视化模块. 随着云计算技术的日趋成熟, 基于网格的 PSE-Bio 也逐步向基于云计算的 CPSE-Bio 过渡. 本工作首先对近年来的工作进行了综述. 随后针对 CPSE-Bio 选取了其中的多数据库 PPI 网络查询模块和蛋白质挖掘模块进行了分析. 通过比较发现, MDRT 模块具有更高的查询性能. 在对 PM 模块实验中, 结合了 KVM 虚拟化技术后可以有效地提高资源的使用率, 并且对于低通信的数据密集型计算应用, 虚拟化技术可以最大化地利用计算资源, 提高运行效率. 结果表明, 利用云的良好特性, 实现了基于云的生物数据管理、多数据库 PPI 网络查询和蛋白质挖掘等模块, 并在实际运用中具有良好的表现.

**致谢** 研究生物信息学研究团队的 PSE 工作得到香港科技大学穆默教授、日本宇都宫大学 Shigeo Kawata 教授的支持与指导, 团队中的研究人员和研究生们也参与了研发工作, 在此予以感谢.

## 参考文献:

[1] 俞艳, 郭胜利, 何建华. 基于 Web 服务的土地适应性评价 PSE 设计与实现[J]. 武汉大学学报, 2006(6): 544-547.

[2] Hesper B, Hogeweg P. Bioinformatica: een werk concept [J]. Kameleon, 1970, 1(6): 28-29.

[3] Gallopoulos E, Houstis E, Rice J R. Computer as Thinker/Doer: problem-solving environments for computational science [J]. IEEE Computational Science and Engineering, 1994, 2(1): 11-23.

[4] Boonmee C, Kawata S. Computer-assisted simulation environment for partial-differential-equation problem, 1. data structure and steering of problem solving pro-

cess [J]. Transactions of the Japan Society for Computational Engineering and Science, Paper No. 19980001, 1998.

[5] Boonmee C, Kawata S. Computer-assisted simulation environment for partial-differential-equation problem, 2. Visualization and steering of problem solving process [J]. Transactions of the Japan Society for Computational Engineering and Science, Paper No. 19980002, 1998.

[6] Kawata S, Boonmee C. Visual steering of the simulation process in a scientific numerical simulation environment NCAS [J]. Enabling Technologies for Computational Science, 2000, 548: 291-300.

[7] Mu M. PDE mart: a network-based problem solving environment for PDEs [J]. ACM Trans Mathematical Software, 2005, 31(4): 508-531.

[8] Mao G Y, Mu M. Grid-based PDE. Mart: a PDE-oriented PSE for grid computing [C]// The 1st International Conference on e-Science and Grid Computing. 2005: 464-469.

[9] Kobashi H, Kawata S. PSE park: a framework to construct problem solving environments [C]// The 12th PSE Workshop. 2009.

[10] Akarsu E, Fox G C, Furmanski W. Webflow-high-level programming environment and visual authoring toolkit for high performance distributed computing [C]// ACM/IEEE conference on Supercomputing. 1998: 1-7.

[11] Goodale T, Allen G, Lanfermann G. The cactus framework and toolkit: design and applications [C]// High Performance Computing for Computational Science—VECPAR 2002. 2003, 2565: 197-227.

[12] The Grid Portal Toolkit [EB/OL]. [2012-11-19]. http://gridport. npaci.edu.

[13] Apweiler R, Bairoch A. UniProt: the universal protein knowledge base [J]. Nucleic Acids Res, 2004, 32(1): 115-119.

[14] 赵志康, 谢江, 李松倍. 基于 Web Service 的蛋白质相互作用网络 PSE [J]. 计算机工程与设计, 2009, 30(18):4326-4329.

[15] 毛国勇, 张晓斌, 谢江. 面向生物信息学的网格问题求解平台 [J]. 计算机工程, 2010, 36(11): 253-255.

[16] 毛国勇, 张晓斌, 谢江. 基于 Web 的 PSE-Bio 交互式可视化 [J]. 计算机工程与应用, 2010, 46(31): 77-79.

[17] Xie J, Zhang X B, Zhang W. PSE-Bio: a grid enabled problem solving environment for bioinformatics [C]// The 3rd IEEE International Conference on e-Science and Grid Computing. 2007: 529-535.

[18] Xie J, Zhang Y W, Zhang W. Studies of agent composition model of PSE-Bio workflow [C]// The 4th IEEE International Conference on e-Science. 2008: 743-748.

[19] Mao G Y, Xie J. SVG-based interactive visualization of PSE-Bio [C]// The 2009 High Performance Computing and Application. 2010: 288-294.

[20] Yu L, Xie J, Cheng X. BNMatch: a cytoscape plugin for querying and visualizing matched similar networks [C]// The 2010 International Conference on Computer and Computational Intelligence. 2010: 476-478.

[21] Cheng X, Xie J, Yi R G. Data management and application on CPSE-Bio [C]// The 2011 Proceedings of the International Conference on Human-Centric Computing. 2011: 591-599.

[22] Xie J, Yi R G, Tan J. Multi-database retrieval technology on CPSE-Bio [C]// The 6th International Conference on Computer Sciences and Convergence Information Technology. 2011: 280-284.

[23] Xie J, Xia S Y. Virtual technology on CPSE-Bio [C]// The 7th International Conference on Computing and Convergence Technology. 2012: 1478-1481.

[24] Cytoscape [EB/OL]. [2012-11-19]. http://www.cytoscape.org.

[25] He H, Singh K. Closure-tree: an index structure for graph queries [C]// The 22nd International Conference on Data Engineering. 2006: 38.

[26] 张建勋, 古志民, 郑超. 云计算研究进展综述 [J]. 计算机应用研究, 2010, 27(2): 429-433.

[27] Bader G, Cary M, Sander C. Pathguide: a pathway resource list [J]. Nucleic Acids Res, 2006, 34(S): 504-506.

[28] Wu C H. The protein information resource [J]. Nucleic Acids Res, 2003, 31(1): 345-347.

[29] Orchard S. Protein interaction data curation: the International Molecular Exchange (IMEx) consortium [J]. Nature Methods, 2012, 9(4): 345-350.

[30] Borden T, Hennessy J, Rymarczyk J. Multiple operating systems on one processor complex [J]. IBM Systems Journal, 1989, 28: 104-123.

[31] Habib I. Virtualization with KVM [J]. Linux J, 2008, 166: 8.