专注:云计算

在谷歌应用程序引擎上评估高性能计算

因nsbruck大学的Radu Prodan, Michael Sperk和Simon Ostermann

仅在基础设施即服务(infrastructure as a service, laas)上——在这些基础设施上，您可以轻松部署封装在虚拟机中的遗留应用程序和基准测试。我们提出了一种评估基于平台即服务(Paas)的高性能计算云平台的方法:谷歌应用程序引擎(GAE)。“GAE是一个简单的并行com开发框架，支持计算密集型HPC算法和应用程序的发展。底层谷歌基础设施透明地调度和执行应用程序，并为性能和成本分析生成详细的概要信息。GAE支持为较小的公司开发可伸缩的Web应用程序——这些公司负担不起在任何时候都能处理大流量峰值的大型基础设施的过量供应

谷歌应用程序引擎提供了相对较低的资源配置开销，并为少于一小时的任务提供了廉价的定价模型。

谷歌应用程序引擎

AE在谷歌的大型服务器基础设施上托管Web应用程序。它有三个主要组件:可伸缩的服务、运行时环境和数据存储。

GAE的前端服务处理HTTP请求并将它们映射到适当的应用程序服务器。应用程序服务器启动、初始化和重用传入请求的应用程序实例。在流量高峰期间，GAE自动分配额外的资源来启动新实例。应用程序的新实例数量和请求的分布取决于流量和资源使用模式。因此，GAE自动执行负载平衡和缓存管理。

每个应用程序实例都在沙箱中执行(一个从底层操作svstem抽象出来的运行时环境)。这可以防止应用程序执行恶意操作，并使GAE能够优化CPU和



资助机构和机构必须购买和提供昂贵的并行计算硬件来支持高性能计算(HPC)模拟。在许多情况下。物理托管成本以及运营、维护、折旧成本超过收购价格，使得整体投资不透明，无利可图。

通过一种新的商业模式

云计算是并行计算机的一种更便宜的选择，也比网格计算更可靠。然而，它仍然由商业和工业应用主导;它对并行计算的适用性在很大程度上仍未被探索。

到目前为止，对科学云计算的研究几乎集中在一起

IEEE软件I由IEEE计算机协会出版

0740 - 2012年7459/12/831.00 1 eeЕ

授权许可使用仅限于:拉筹伯大学。对n下载

ecember,2020, 08:22:22 UTC从EEE Xplor

限制适用。

同一物理机器上多个应用程序的内存利用率。沙箱还对程序员施加了各种限制:

应用程序不能访问底层硬件，只能访问网络设施。Java应用程序只能使用标准库功能的一个子集。

ona

应用程序不能使用线程。一个请求最多有30秒来响应clien

GAE应用程序使用资源，如CPU时间，I/0带宽，和在特定的与每个资源类型的CPU时间相关联的请求的数量，在模糊的术语中，相等于CPU周期的数量，a1.2-GHz Intel x86处理器可以每。在相同的时间内形成。关于资源使用的信息可以通过GAE应用程序管理Web接口获取。

最后，数据存储让开发人员能够在请求之外持久存储数据。数据存储不能在不同的从应用程序之间共享

一个并行计算

框架

为了支持使用GAE开发paral lel应用程序，我们设计了一个基于java的通用框架(se图1)。在我们的框架中实现一个新的应用程序需要对三个抽象接口(类)进行专门化:JabFactory。WorkJob。和结果(参见图2)

主应用程序是一个Java程序，它在用户的本地机器上实现Jobractory。Jobractory在多个工作任务中管理算法的逻辑和并行化。Work Job是一个抽象类，实现为每个从应用程序的一部分——特别是

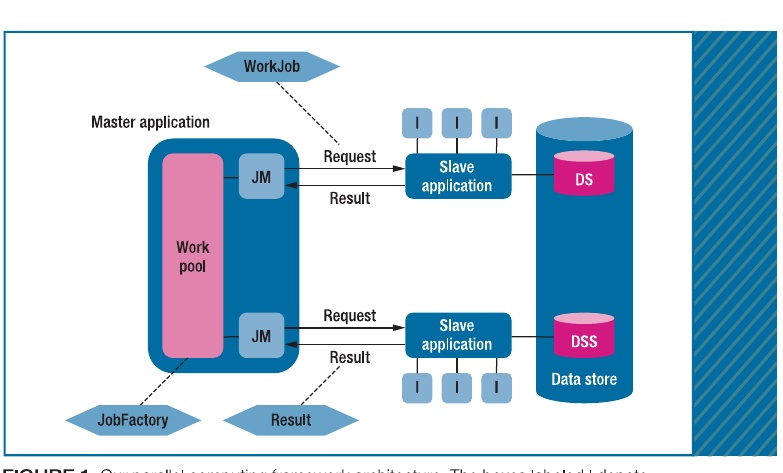


图1所示。我们的并行计算框架架构。标记为I的方框表示

多个奴隶实例。主应用程序负责在作为GAE Web应用程序实现的并行从服务器之间生成和分发工作，并负责实际的计算。

方法，它执行实际的计算iob。每个从属应用程序作为单独的GAE应用程序部署，因此具有不同的URI。从应用程序提供了一个简单的HTTP接口，并接受数据请求或计算性iob请求。请求

HTTP消息头存储请求的类型

作业请求包含一个提交到从属应用程序并进行扩展的作业。如果多个请求被提交到同一个从属应用程序，GAE将自动启动和管理

我使用多个实例来处理当前的负载:程序员无法控制实例。(从理论上讲，一个从属应用程序就足够了;然而，我们的框架可以在多个从属应用程序之间分配作业，以解决更大的问题。)

data reauest将所有作业共享的数据传输到由usesharedData方法指定的持久数据存储区)。

它使用多个并行HTTP请求来实现GAE的最大HTTP有效负载大小(1 Mbyte)并提高带宽利用率。fetchsharedData方法根据需要从数据存储中检索共享数据。

在一个明确的请求，从应用程序删除整个数据存储内容。清除请求通常在运行之后发生，无论运行成功还是失败。

一个ping请求立即返回并确定从应用程序是否仍然在线。如果从服务器离线，则主服务器将回收该作业并将其提交给另一个实例。

WorkJob执行

将工作任务映射到资源遵循一种动态工作池方法，这种方法适合作为黑盒在沙箱资源上运行，并且执行时间不可预测。每个从应用程序在主应用程序上下文中都有一个关联的作业管理器。它从全局池请求工作任务，并将它们提交给它的从属服务器

|IEEE SOFTWARE 53

授权许可使用仅限于:拉筹伯大学。于2020年12月06日08:22:22 UTC从IEEE Xplore下载。限制

专注:云计算

JobFactory公共接口。公共WorkJob getWorkJob0;t(Resu public Result getEndResulto;公共布尔useSharedData (l

公共可序列化的getSharedDatag:

公共抽象类WorkJob扩展了可序列化的私有int id;

公共int getld (;

公共空间setld (int);

公共运行结果();

公共空间fetchSharedData ()

public抽象类实现Serialzable (private int id;

私人长cpuTime;

公共长getCPUTime ():

公共setCPUTimellong无效)

公共int getld);

公共空间setld (int);

图2。我们的并行计算框架接口的Java代码。JabFactory实例化主应用程序，WorkJob实例化从应用程序，Result表示从计算的最终结果。

开发服务器在单独的线程中管理并行请求。

资源配置

资源配置开销是发出HTTP请求和接收HTTP响应之间的时间间隔。TCP网络下的各种因素影响开销(例如，将请求分配给应用服务器的负载平衡，如果不存在实例，则包括实例的初始化)。

为了测量开销，我们以300kbyte的步骤发送了有效负载在O到2.7 mbvte之间的HTTP ping请求，对于每种大小重复50次，然后取平均值。开销并没有随着pavload线性增加(见图3)，因为TCP在更大的负载下获得了更高的带宽，我们以秒为单位测量了开销;基于laas的基础设施，比如Amazon Elastic Compute Cloud (EC2)，显示了以分钟计算的延迟

计算实例(GAE指定使用哪个实例)，并返回部分结果

我们将队列大小与每个从服务器关联，以指示它可以同时处理的副Jel作业的数量。尺寸应与未计算的可用处理芯数量相对应。在某一时刻找到最佳的规模是困难的，原因有两个。首先，GAE不发布其硬件信息;其次，应用程序可能与其他竞争的应用程序共享硬件。因此，我们在每次实验前都进行了热身训练，来近似给定时间的队列大小。

从应用程序序列化工作lob的结果，并将它们包装在HTTP响应中，由主程序收集和组装响应。结果与工作作业具有相同的惟一标识符。calculationlime字段存储在run()中用于性能评估的有效计算时间

失败

GAE环境可能有三种类型的故障:超出配额、脱机从属应用程序或conP丢失

nectivity。为了应对这种故障，主服务器实现了一种简单的容错机制，根据故障类型使用相应的指数回退超时将失败的工作重新提交给相应的从服务器。

基准

我们开始使用一组基准来评估GAE，以提供将并行应用程序调度到其资源上的重要信息。为了帮助用户了解使用沙箱资源从本地并行计算机转移到远程云的代价，我们在Karwen- el上部署了一个GAE开发服务器，这是一台拥有16 gb内存和4个2.2 ghz双核Opteron处理器的本地计算机。属性不生成额外的沙箱实例

即时Compilatio

Java virtual machine的just-in-ime (JIT)编译将字节代码中经常使用的部分转换为本机代码，显著提高了性能。为了观察JIT的编译效果，我们实现了一个简单的Fibonacci数生成器。我们在延迟1秒的情况下，连续向GAE提交了50次，始终使用相同的问题大小。我们在不运行实例的情况下设置从应用程序，并在每个工作任务的run()中测量有效计算时间。如前所述，GAE根据应用程序最近的负载生成应用程序的实例(请求越多，实例越多)。为了标记和跟踪实例，我们使用了一个单例类，它包含一个随机初始化的静态标识符字段。

图4显示了7个实例处理了50个请求。而且，每个实例中的前两个请求花费的时间比其他请求要长得多。后

IEEE SOFTWARE I网站www.COMPUTER.ORG/SOFTWARE

授权许可使用仅限于:拉筹伯大学。于2020年12月06日08:22:22 UTC从IEEE Xplore下载。限制适用。

它的编译速度快了三倍以上。

蒙特卡罗模拟

一种近似的方法是通过一个简单的蒙特卡洛模拟，将一个圆变成一个正方形，在正方形中产生p个均匀分布的随机点，然后计算圆周上的m个点。我们可以估计T=4 m/p。我们在GAE上运行此算法

从GAE获得一致的度量很困难，有两个原因:首先，程序员无法控制从实例。其次，对同一个Web应用程序的两个相同的连续请求可能在不同位置的完全不同的硬件上执行。为了使偏差最小化，我们重复了所有实验10次，消除了异常值，并取了所有运行的平均值。

运行模拟。我们为每个应用程序进行了一个预热阶段，以确定队列大小并消除JIT编译的影响。我们先按顺序执行计算算法，然后通过在Jobractory工作池中生成相应数量的工作来增加并行作业的数量。我们选择了一个有2.2亿个随机点的问题，它产生的顺序执行时间略低于30秒的限制。

对于每个实验，我们测量和分析了两个指标。第一个是计算时间，它表示run()的平均执行时间。第二个是平均开销，它表示总执行时间和计算时间之间的差异(特别是由于请求延迟)。

结果。图5显示了GAE上的串行解决比Karwendel上大约慢两倍，这是由于GAE的标准数学库中较慢的随机数字生成例程。

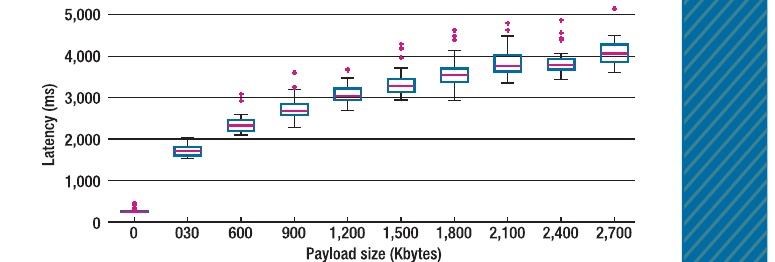


图3。资源配置开销不会随着有效负载线性增加，因为TCP在更大的有效负载下获得了更高的带宽。

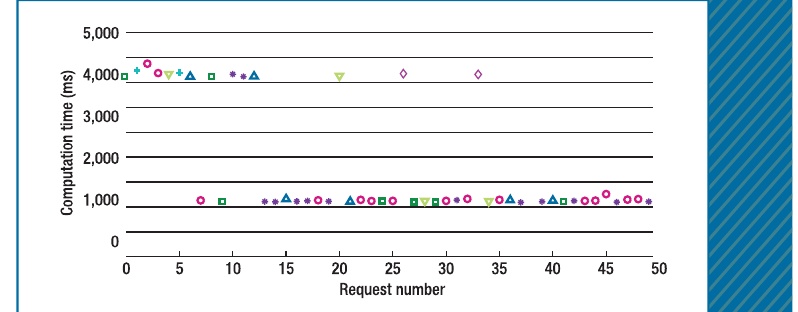


图4。计算时间和请求到实例的映射。每个实例中的前两个请求花费的时间比其他请求长得多。在即时编译之后，代码的执行速度几乎快了四倍。

在Karwendel上，由于主服务器和从服务器之间快速的本地网络，任务和结果的传输几乎没有开销。因此，平均计算时间和总执行时间几乎是相同的，直到8个并行作业(Karwendel有8个核)。在此之前，几乎是线性加速。使用超过8个并行作业会产生负载不平衡，从而导致加速恶化，因为两个作业必须共享一个物理核心。

在这两种情况下，GAE展示了一个恒定的数据传输和大约700毫秒的总开销，这解释了它较低的加速速度。在GAE服务器或Internet网络上造成的随机后台负载

不同机器大小的执行时间略有不规则。

这种经典的可伸缩性分析方法不适合GAE，因为30秒的限制让我们只能执行相对较小的问题(其中Am dahl定律限制了可伸缩性)。为了消除这个障碍并评估GAE在计算更大问题方面的潜力

।在lems中，我们使用Gustafson法则来根据机器的大小比例增加问题的大小。我们观察了对执行时间的影响(对于理想的加速，这应该保持不变)。我们将作业分配给10个GAE从属应用程序，而不是一个，以获得足够的配额(以分钟为单位)。

在这个例子中，我们从1。8亿的随机问题开始

2012年3 / 4月|IEEE SOFTWARE 55

地址:拉筹伯大学。于2020年12月06日08:22:22 UTC从IEEE Xplore下载。限制适用。

授权许可使用

专注:云计算

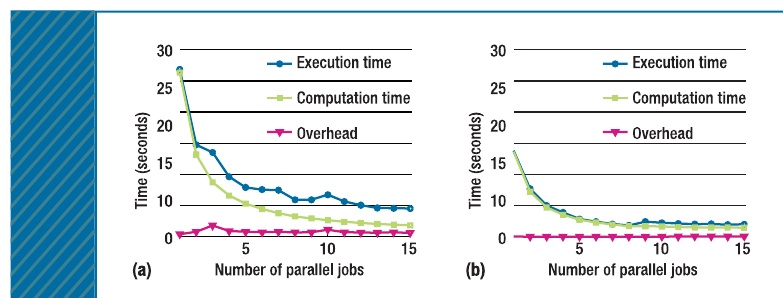


图5。在(a)谷歌应用程序引擎(GAE)和(b)本地机器Karwendel上计算n的结果，GAE上的串行执行比在Karwendel上慢两倍，这是由于GAE的标准数学库中较慢的随机数字生成例程。

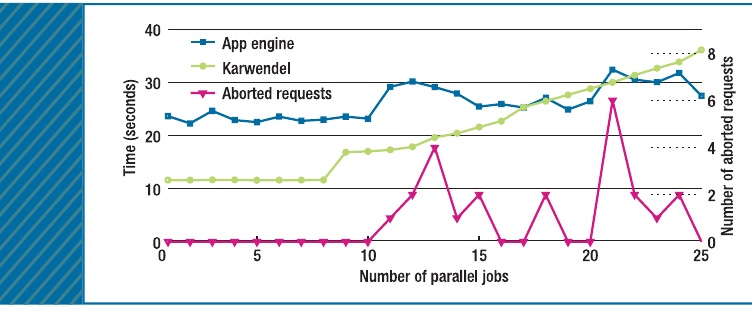


图6。可伸缩性的结果，GAE和Karwendel为比例增加的机器和问题大小。Karwendel的执行时间是恒定的，直到8个并行作业，这证明了我们的框架的良好可伸缩性。

由于谷歌的更大的硬件基础设施

成本规划方法

尽管我们在谷歌提供的免费每日配额内进行了所有的实验，但估算成本对于理解在现实生活中执行应用程序的价格仍然很重要。因此，除了r近似，我们实现了三种不同计算和通信复杂度的算法(见表1):

矩阵乘法，基于第一个矩阵的行向分布和第二个矩阵的全广播:曼德尔布洛特集的生成，基于逃逸时间算法;和

分数要避免超过30分。第二个限制。(对于大量的作业，GAE不能提供更多的资源并开始拒绝连接。)同样，Karwendel在8个并行作业之前的执行时间都是恒定的(参见图6)，这证明了我们的框架具有良好的可伸缩性。

从9个并行iobs开始，执行时间随着问题的大小稳步增加。GAE在10个并行作业之前显示了类似的良好可伸缩性。广告开始。传统并行作业稍微增加了执行时间。开销的请求中止(由于达到了配额)导致了大多数不正常行为。

对于超过17个并行任务，GAE的执行时间较低

秩排序，基于每个数组的单独秩计算。这可能会超过其他更快的顺序算法。

对于每个问题的大小，我们依次运行了100次实验，并对三种最常用的资源的成本进行了分析:CPU时间、输入数据和输出数据，这些数据是我们通过谷歌应用程序管理接口获得的。我们使用2011年1月10日的谷歌价格:1S$0.12 per outgoing Gbvte。$0.10每传入Gbyte，和$0.10每CPU小时。我们没有对数据存储配额进行分析，因为总体CPU小时包括了它的使用情况。

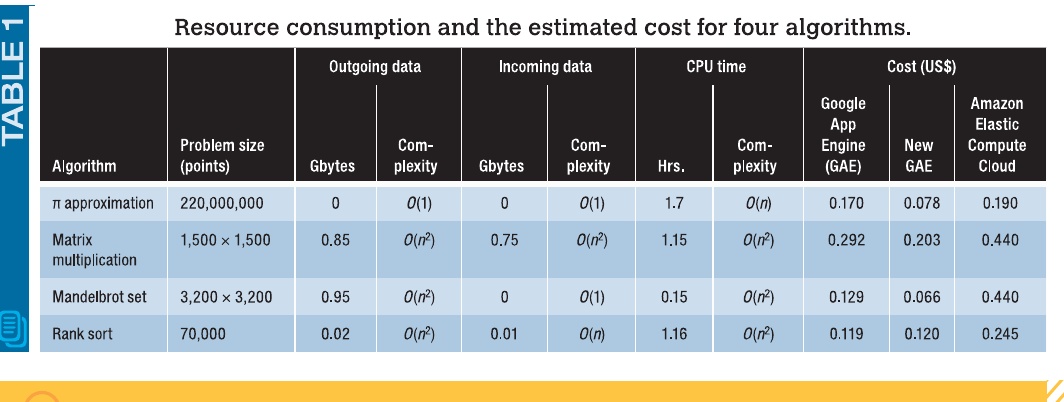
正如我们所预期的，n近似是计算最密集的，并且几乎没有数据传输成本。令人惊讶的是，与CPU时间相比，秩排序消耗的带宽很小，即使完全未排序的rray必须传输到从服务器，而每个元素的秩必须传输回主服务器。曼德尔布罗特集生成器显然是由必须传输到主服务器的图像数据量控制的。对于n近似，我们一般可以对ap进行近似采样129。109个随机点，因为算法是线性计算。对于其他算法。精确的估计更加困难，因为资源消耗不会随着问题的大小线性增长。然而，我们可以使用表1中列出的资源复杂性来大致估算执行新问题的成本。

最后，我们估计了使用EC2的ml.small实例在Amazon FC2基础设施上运行相同实验的成本，这些实例的计算性能相当于一个EC2 compute uni，相当于一个1.2 ghz Xeon或Opteron处理器，后者类似于GAE，并支持直接比较。我们将实现的算法打包到基于xenonvirtual中

www.coMPUTER

ORG/SOFTWARE

授权许可使用仅限于:拉筹伯大学。于2020年12月06日08:22:22 UTC从IEEE Xplore下载。限制适用。



目

3П8六世

云性能方面的相关工作

对4个基于服务的商业基础设施科学计算云的分析表明，云的性能低于传统的科学计算。然而，分析表明，对于那些需要即时和临时资源的科学家来说，云计算可能是一个可行的替代方案。

Alexandru losup和他的同事研究了谷歌应用程序引擎(GAE)和Amazor弹性计算云(EC2)的长期性能变异性。结果表明，二年生和陈年生均有显著性差异，二年生和陈年生均有显著性差异。Che的研究人员得出结论，GAE和EC2的表现不同。在不同的大规模应用中

Christian Vecchiola和他的同事从高性能计算应用的角度分析了不同的clouc供应商，强调Aneka平台即服务(Paas框架)。Aneka需要第三方部署云平台，不支持GAE

Windows Azure是一个可与通用电气相媲美的Pas提供商，但更适合解决科学问题。李杰和同事们将其性能与台式电脑进行了比较，但没有进行成本分析。

MapReduce框架为云计算提供了一种不同的方法。5.6 MapReduce是一个以大数据处理为目标的正交应用类。它不太适合于计算密集型的并行算法，例如那些

在小型数据集上操作。此外，它不支持更复杂的应用程序的实现，如递归和非线性问题或科学工作流

参考文献

1.A. losup等，“多任务智能计算的云计算服务性能分析”。EE反式。

Sstems。卷。2。no.6.20110n谷物。副茎和尾茎22 0,D。

2.A losup, N. Yigitbasi, D. Eema， <关于…的性能变量>

生产云服务，《第1期EE/ACM Int!》计算机协会。集群、云计算和网格计算(CCcrid 1I, I E CE D 0-1)

“高性能云计算:科学应用的观点”，《计算机科学与应用》，2003年。

Dp.4-16

j . Liet。“科学领域:更多的卫星数据分析

reduce ton Pipeline in Windos Azure Platform，" Proc. 2010 intl Symp. "并行分布式处理(ips10o，即E5, 20p10)。张建民，“云计算之应用”，国立台湾科技大学计算机科学研究所硕士论文

加州，圣巴巴拉，2009:www.cs。ucsb.edu/-cab/apers/mapscale。来

6.邱杰等，生命科学应用的云与集群混合计算范式，《生物信息》，第1卷，第12期，2010年，第3期;www.biomedcentral.com/content/pdf/471 - 2105 - 11 - s12 - s3.pdf。

J. Dean和S. Ghemawat，“MapReduce:大集群上的简化数据处理”，Comm。ACM，第51卷，no。J. Ekanayake和G. Fox，“Hligh利用云和云技术实现并行计算”，2008年1月，第10113页。云计算和软件服务。理论和技术。s.a.ahson和m.e Ilvas, eds..CRC出版社,2010年

在ml.smll实例上部署和引导的机器。表1显示GAE的计算成本较低，主要由于基于心血管的

与EC2的每小时计费间隔相反的ayments。

谷歌最近宣布将改变其定价模式

带有新实例小时的CPU周期

单位。这个单元相当于一个应用程序实例运行一个小时，成本为0.08美元。此外,谷歌

2012 | MARCHIAPRIL IEEE软件

57

授权许可使用仅限于:拉筹伯大学。于2020年12月06日08:22:22 UTC从IEEE Xplore下载。限制适用。

专注:云计算



拉杜·普罗丹是印第安纳大学-布鲁克计算机科学研究所的副教授。他的研究兴趣包括racsina方法、编译技术和闹剧分析

ning mcо 作,performar ani姐姐,

并行和分布式系统的调度。Prodan拥有维也纳科技大学的技术服务博士学位。可以通过radu@dps.uibk.ac.at联系他。

迈克尔·斯珀克是英斯布鲁克大学的博士生。他的研究兴趣包括分布式计算和并行计算。斯珀克是Innspruck大学计算机科学和ner科学硕士。请致电prk michegmailcom。

西蒙·奥斯特曼(SIMON OSTERMANN)是印第安纳布鲁克大学计算机科学研究所的博士生。他的研究兴趣包括网格和云计算的资源管理和调度anasаut Ostermann有o因斯布鲁克大学计算机科学硕士学位。请通过simon@dps,uibk,ac.at联系他。

美国甲骨文公司

甲骨文美国公司软件工程师职位空缺。

独立工作，设计，开发，测试和调试计算机软件系统。参与创建强大的下一代web应用程序，以提供高度可视化和交互性的用户体验。分析、设计和构建软件应用的新特性和新功能。开发和维护Oracle软件产品系列的当前版本和未来版本。识别和开发应用程序和产品的改进。计划和实施应用程序和产品的可移植性。计算机科学、工程、数学、物理或相关技术专业本科或以上学历，5年以上计算机相关工作经验。经验必须包括:计算机软件的设计、开发、测试、实现;准备系统关键模块的详细技术设计;使用JSP、Struts、JSTL、JavaScript、AJAX设计、开发、实现多个模块的GUl;使用SOA设计、开发、实现Web服务客户端到接口;使用Hibernate开源框架和JDBC设计、开发、实现各种后端组件。经验可以同时获得。任何合适的教育、培训或经验的结合都是可以接受的。8:00AM-5:00PM Mon-Fri;$1万/r，标准com-任何优惠。请将简历发送至:美国甲骨文公司招聘就业处，收件人:美国甲骨文公司。作业编号H: ORA85286, PO。佐治亚州亚特兰大市56625号信箱30303。

每个申请将收取每月S9元的费用。新模型将主要损害Web应用程序，这些应用程序在稀疏请求高峰时触发额外的实例，然后保持空闲状态

表1给出了一个粗略的成本估计，假设15个并行任务和用于有用计算的实例利用率为80%。结果表明，新的定价模型有利于尝试充分利用所有可用实例的cpu密集型应用程序。此外，在新的定价模式下，我们可以期待免费资源持续时间更长

沃尔恩计划调查应用类这样

作为科学的工作流应用程序，将在我们的通用框架之上实现，并在GAE上运行，具有改进的性能。有关云计算性能的其他研究，请参见侧栏“云性能中的相关工作”。

致谢

奥地利科学基金proiect TRP 72-N23和Standortagentur Tirol雨云项目资助了这项研究。

参考文献

1.D.桑德森，编程谷歌应用程序引擎，O'Reilly Media, 2009。

2.王建民，“云计算服务在多任务科学计算中的性能分析”，清华大学学报。并行与分布式系统，第22卷，第2期。2011年6月，第931-945页。

3.M. Sperk，《云中的科学计算与谷歌应用引擎》，硕士论文，因斯布鲁克大学数学、计算机科学与物理学院，2011年;http:// dps.uibk.ac.al / -radulsperk。pdf。

4.“对Amdahl定律的重新评估”，《通信学报》，第31卷，第1期。1988年5月，第532-533页。

你也可以在http://ComputinaNow.computer.ora上免费获得一些计算机专业的文章和专栏。

cr

IEEE SOFTWARE I网站www.COMPUTER.ORG/SOFTWARE

授权许可使用仅限于:拉筹伯大学。于2020年12月06日08:22:22 UTC从IEEE Xplore下载。限制适用。