虚拟机竞价型实例价格统计建模

摘要 – 现如今，云计算资源的需求激增，消费者需要权衡价格，性能和可靠性之间的各种关系。 亚马逊的竞价型实例（SI）以较低的价格为用户创建竞争性竞价选项，然而服务的可靠性却欠佳。通常认为SI价格便宜，然而从文献中可以看出，其诸多特征并没有被深入挖掘。我们认为SI的表征是在现代市场的公共云环境中的随机调度算法和容错机制的设计的基础。在本文中，我们基于亚马逊EC2的四个数据中心的一年价格历史对SI进行了全面分析。 为此，我们根据现货价格和价格时间比（价格变动之间的时间）分析了所有不同类型的SI，并确定了现货价格的时间动态。此外，我们提出了一个适合这两个数据系列的统计模型。 结果表明，我们能够通过高斯分布与三个或四个分量的混合来模拟现货价格动态以及每个SI的价格时间分布。此模型通过广泛的模拟验证，证明我们的模型在实际工作条件下表现出良好的准确度。

关键词：云计算; 现货价格; 统计模型;

## 1.引言

由于云计算资源需求激增，我们越来越需要考虑和分析在如何权衡价格和性能之间的关系。我们知道，有一种类型的云服务（称为基础架构即服务（简称IaaS））以虚拟机（VM）的形式，采用按需付费，各种价格对应不同容量和存储。例如，Amazon提供按需和保留的VM实例，这些实例与固定的设定价格相关。但是，亚马逊可以根据自己的本地政策增加或降低这些价格[13]。在两个操作系统（即32个用于Linux和32个Windows）下，有64种不同类型的实例具有不同的容量和价格，亚马逊在四个数据中心提供这些数据，如表I所示（按价格排序）。在此表中，给出了Linux操作系统的价格，标有“m1”，“m2”和“c1”的实例分别是标准，高内存和高CPU实例。

2009年12月，亚马逊发布了一种称为竞价型实例（SI）的新型实例，用于销售亚马逊EC2数据中心的空闲的资源[3]。SI的价格和现货价格，取决于实例的类型以及每个数据中心内的VM需求。 事实上，spot实例是其他两类实例的替代品，它们为公共云用户提供了低价格，但不太可靠和竞争性的出价选项。 因此，另一方面，可靠性是云计算资源的一个重要的指标，这也体现了计算系统现在面临的诸多挑战。

为了更好的利用SI，用户提供的出价是一小时使用的最高价格。 每当SI的当前价格等于或小于用户出价，该实例对用户可用。 如果SI的价格高于用户的出价，出价事件（失败），则VM将自动终止，并且用户不支付任何部分时间。 然而，如果用户终止运行的VM，则必须支付整个小时的费用。亚马逊在创建虚拟机时按SI的市场价格对用户以小时为单位计费。

我们在市面上很难找到有关如何利用SI来降低云用户的效用计算的货币成本的书籍[12]。 SI的统计分析和建模没有出现在已有文献中，因此这将是我们的研究的重点。在本文中，我们根据现货价格和四个亚马逊数据中心（即美国西部，东南部，欧西部和东南部）的价格（价格随时间变动）提供有效的SI综合分析 ）。 此外，我们建议使用一个统计模型来捕捉亚马逊数据中心的波动现货价格。本文的主要贡献如下：

* 我们为亚马逊EC2数据中心中的所有SI提供统计分析。我们还确定现货价格在按小时计费与按天计费方面的时间相关性。
* 我们用高斯分布来模拟现货价格和每SI的价格时间比。还提出了模型校准算法来处理真实价格历史中观察到的价格趋势。
* 我们通过在真实工作条件下的模拟来验证我们提出的模型的准确性。

表I

AMAZON不同数据中心的在线需求价格（CENTS）。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Instances | us-west | us-east | eu-west | ap-southeast | EC2 Compute Unit | Memory (GB) | Storage (GB) |
| m1.small | 9.5 | 8.5 | 9.5 | 9.5 | 1 | 1.7 | 160 |
| c1.medium | 19 | 17 | 19 | 19 | 5 | 1.7 | 350 |
| m1.large | 38 | 34 | 38 | 38 | 4 | 7.5 | 850 |
| m2.xlarge | 57 | 50 | 57 | 57 | 6.5 | 17.1 | 420 |
| m1.xlarge | 76 | 68 | 76 | 76 | 8 | 15 | 1690 |
| c1.xlarge | 76 | 68 | 76 | 76 | 20 | 7 | 1690 |
| m2.2xlarge | 114 | 100 | 14 | 114 | 13 | 34.2 | 850 |
| m2.4xlarge | 228 | 200 | 228 | 228 | 26 | 68.4 | 1690 |

我们认为这项研究的结果将对公共云环境中的现货市场的随机调度算法和容错机制（例如，检查点和复制算法）的设计有很大的帮助。此外，该模型可以由其他IaaS云提供商使用，期望在不久的将来可以提供这样的服务。

本文结构如下：在本文第二部分，描述了建模的过程。第三部分讨论相关工作。第四部分则是研究现货价格的模式。在第五部分，我们给出了所有SI的全局统计。 然后我们在第六部分中说明现货价格和价格时间比的分布拟合。 在第七部分，我们提出了一种用于模型校准的算法。 我们通过第八部分中的仿真讨论了拟议模型的验证。 在第九部分中，我们总结了我们的贡献，并描述了未来的发展方向。

## 2.建模方法

在本节中，我们将描述我们将要分析和建模的两个变量。 在亚马逊的数据中心，SI具有由云提供商指定的两个变量（即现货价格和价格时间比），以及由用户确定的一个变量（用户的出价）。 在本研究中，我们专注于现货价格和价格时间比的分析和建模，作为两个高度易变的系统变量。 这些变量在图1中示出，其中P i是在时间t 1的SI的价格。 因此，价格时间比被定义为Ti = ti + 1-ti。现货价格（Pi）和价格时间比（Ti）的时间序列在以下部分进行分析和建模。

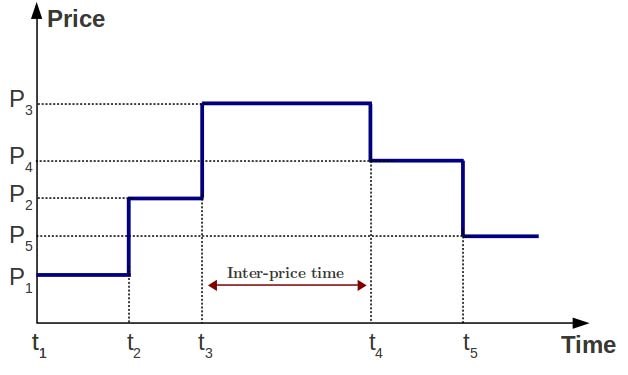


图1 竞价型实例的竞价价格

我们在本研究中使用的数据是从2010年2月1日到2011年2月中的所有亚马逊SI的一年价格历史。我们使用了前10个月（2010年2月至2010年11月）数据来进行建模。

这10个月的数据以及最近2个月的数据则用于验证模型的正确性。亚马逊为每个数据中心自由提供现货价格历史，也可通过其他第三方提供[1]。 我们没有使用2010年2月之前的数据，因为[2]中报告的算法存在一定的问题。此外，我们只使用来自所有数据中心的Linux操作系统的SI。 由于空间限制以及结果的相似性，我们只展示了一个数据中心（即eu-west）的研究结果。 有兴趣的读者可以参考本文[9]的扩展版本，了解更多关于其他数据中心的讨论。

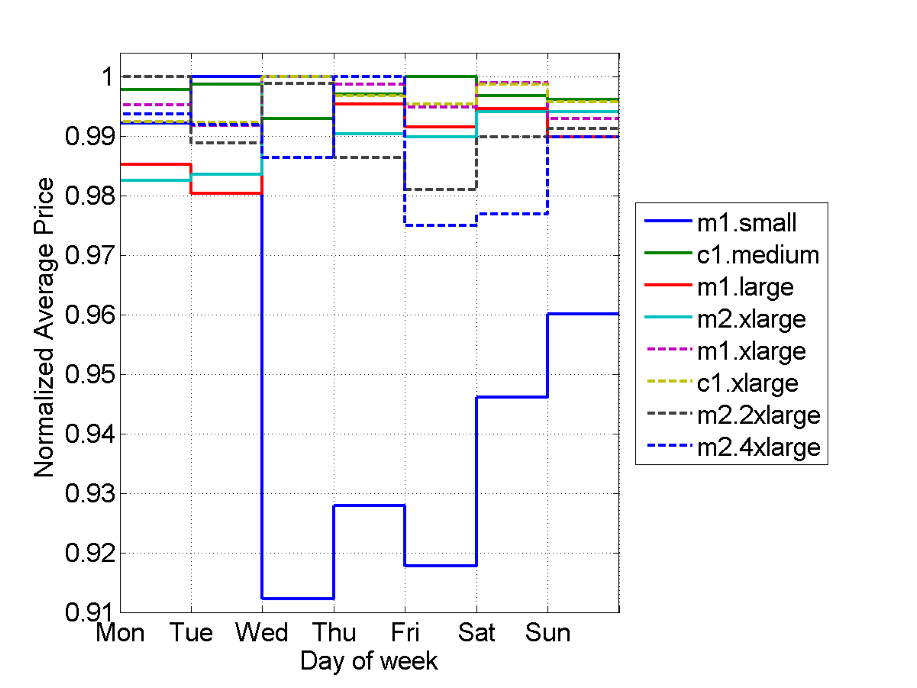
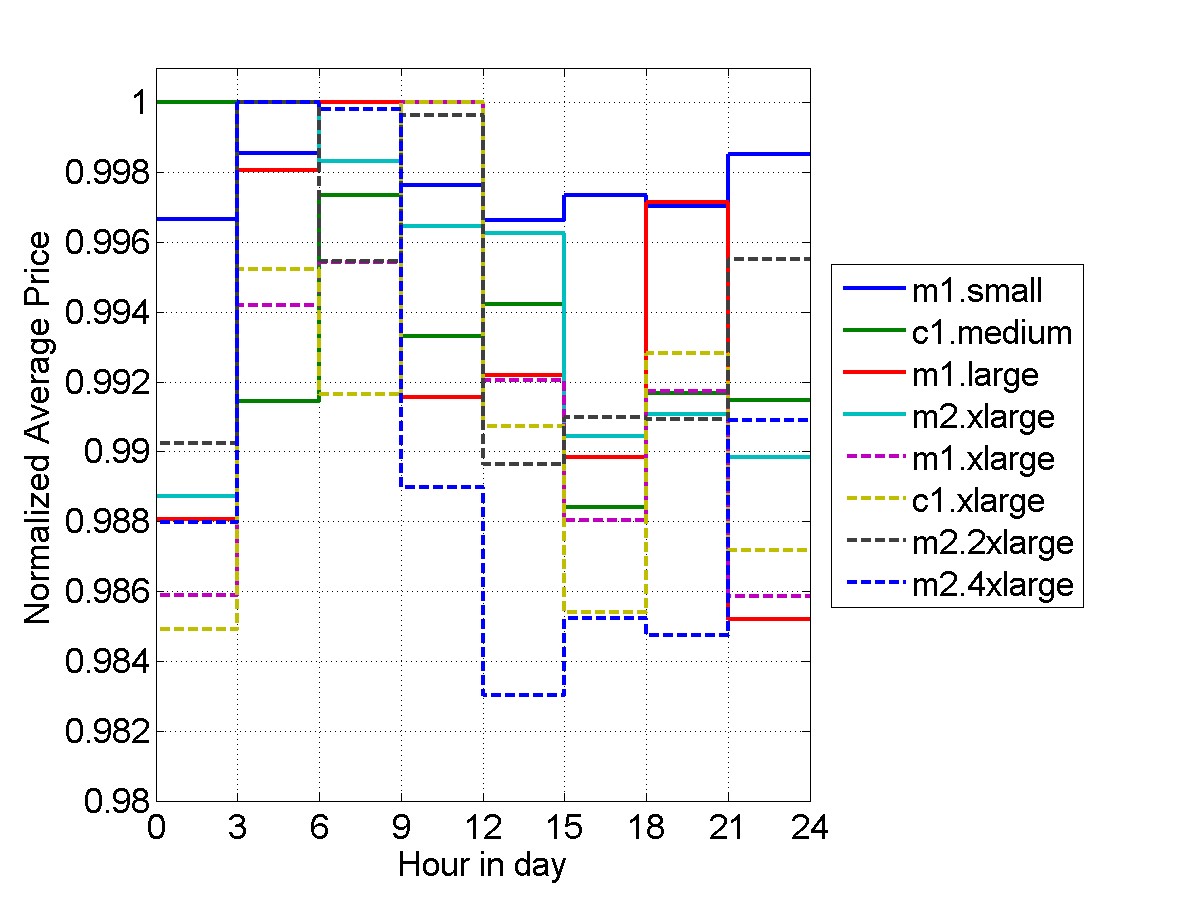
## 3.相关工作

据我们所知，这是我们第一次为公共云计算资源分析和建模。然而，有一些论文调查了SI的使用以减少效用计算的货币成本。

Yi *et al* [14]介绍了一些降低SI成本的检查点的机制。他们使用EC2现货实例的真实价格历史，并说明了自适应检查点如何降低货币成本和提高工作完成时间。 在[4]中，提出了一种在使用SI的SLA约束下优化性能、成本和可靠性的决策模型。 他们使用真实的价格历史和工作量模型来演示他们提出的模型如何用于对SI进行最佳出价，以达到具有期望的置信水平的不同目标。

Chohan *et al*在[6]中提出了一种利用SI来加速MapReduce任务的方法。他们提供了一个马尔可夫链来预测寿命的预期寿命。他们认为，具有容错机制对于在SI上运行MapReduce作业至关重要。此外，在[12]中，作者提出了一种混合云架构来租用SI并用于管理本地集群的峰值负载。他们提出了一些供应策略，并在货币成本节省和截止期限违规数量方面调查了SI与按需实例的利用率。

虽然目前的文献表明，SI在货币成本方面是按需或储备实例的良好替代方案，但社区的用户和研究人员并不清楚SI的特征。 因此，对于这项研究，我们提出了设计统计模型，以便更好地了解亚马逊数据中心的价格机制。



(a) Hour-in-day (b) Day-of-week

图2. 西部数据中心现货价格模式

## 4.价格模式

在本节中，我们检查了在西部数据中心中不同SI的价格的小时和星期时间动态。我们使用与[11]相同的方法来显示一个SI的价格如何在一周或每周的每一天每小时变化。由于我们有GMT时区的价格历史，我们调整时区为当地时间。这种调整可以揭示现货价格对数据中心的本地时间的依赖性。在图2（a）中，我们每天创建8个3小时时隙，并确定每个时隙中每个SI的平均价格。然后，我们用所有天的最大平均价格对这个平均值进行归一化。 注意，3小时抽样的频率可以增加到1小时抽样，一天有24个时隙。 然而，由于亚马逊数据中心的现货价格最初每2-3小时变化一次，因此只会增加样本量，而不会对价格动态产生太大影响（参见第五部分）。

在图2（b）中，我们应用相同的过程来获得一周内所有时间段的平均价格。 在图2（a）中，我们可以看到y轴在[0.98 , 1.0]的范围内，其中在每天的前半部分（[0 , 12]）有递增趋势，后半部分则呈现平缓。

图2（b）中的y轴对于欧西数据中心具有较宽的[0.91 , 1.0]范围。 从这个图可以看出，除了周末价格的下降，并不能找到现货价格的任何特定模式。 然而对于其他亚马逊数据中心，在星期二几乎所有数据中心都呈现出最高价格。 此外，最低的价格在星期六，但在星期天，我们再次观察到所有SI的价格上涨。这些事实在美国东部和东南亚的数据中心更为明显[9]。

## 5.全球统计和分析

在接下来的部分，我们按照表二中的现货价格检查跟踪的基本统计以及表三中的价格时间比分析了在西部数据中心中不同SI的价格历史。表中的统计数据是平均值，修整平均值（在丢弃10％的极值之后的平均值），中值，标准偏差（Std），方差系数（CV），四分位数范围（IQR），最大值， 第三时刻），峰度（第四时刻）和样本数。

这些表显示三种类型的描述性统计。 第一类型的统计（平均值，中值，修剪平均值）揭示了分布的集中趋势。 第二类型的统计量（CV，IQR，最小值，最大值）反映了分布的扩展。 第三类型的统计（峰度，偏度）表示分布的形状。

首先，我们发现SI的平均价格可以低至按需实例的44％（对于作为最便宜的数据中心的东南数据中心，该百分比为38％）。 这表明有一些机会以不可靠的代价降低公用事业计算的货币成本。 此外，一些SI（如m1.large）的最大价格大于相应的按需实例（特别是在US-east数据中心）的价格。 因此，即使用户的出价与按需价格一样高，我们仍然可能具有出价事件的概率。

这些表中的结果显示，对于每个迹线，现货价格的平均值和中值之间的比率以及SI的价格时间比接近1。 这表明高斯分布可能是一个很好的选择模型。 然而，偏度和峰度值显示底层分布是右倾斜和短尾。 因此，高斯分布可能不是使用的代表性模型，并且更好的分布应该是有序的。

表2

欧洲西部数据中心的价格统计数据（CENTS）

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instances | Mean | TrMean | Median | Std | CV | IQR | Max | Min | Skewness | Kurtosis | No. |
| m1.small | 4.00 | 4.00 | 4.00 | 0.19 | 0.05 | 0.20 | 9.50 | 3.80 | 9.44 | 242.97 | 3702 |
| c1.medium | 8.00 | 8.00 | 8.00 | 0.27 | 0.03 | 0.40 | 10.10 | 7.60 | 0.28 | 3.91 | 3812 |
| m1.large | 16.04 | 16.02 | 16.10 | 0.85 | 0.05 | 1.00 | 50.00 | 15.20 | 21.55 | 792.41 | 3875 |
| m2.xlarge | 24.04 | 24.03 | 24.10 | 1.03 | 0.04 | 1.40 | 57.10 | 22.80 | 12.91 | 387.69 | 3763 |
| m1.xlarge | 32.05 | 32.01 | 32.10 | 1.60 | 0.05 | 2.00 | 76.00 | 30.40 | 15.34 | 415.47 | 3917 |
| c1.xlarge | 32.04 | 32.03 | 32.10 | 1.07 | 0.03 | 2.00 | 45.00 | 30.40 | 0.54 | 8.27 | 3658 |
| m2.2xlarge | 56.04 | 56.04 | 56.20 | 1.83 | 0.03 | 3.42 | 76.00 | 53.20 | 0.25 | 4.99 | 4001 |
| m2.4xlarge | 112.08 | 112.08 | 112.50 | 3.62 | 0.03 | 6.80 | 150.00 | 106.40 | 0.21 | 4.55 | 3912 |

表3

欧洲西部数据中心企业时间的统计（以小时为单位的价值）

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instances | Mean | TrMean | Median | Std | CV | IQR | Max | Min | Skewness | Kurtosis | No. |
| m1.small | 1.96 | 1.61 | 1.35 | 2.66 | 1.35 | 0.30 | 109.08 | 0.02 | 19.94 | 727.54 | 3701 |
| c1.medium | 1.91 | 1.59 | 1.34 | 1.86 | 0.97 | 0.32 | 22.81 | 0.02 | 4.53 | 30.63 | 3811 |
| m1.large | 1.88 | 1.57 | 1.33 | 1.79 | 0.95 | 0.31 | 30.94 | 0.02 | 5.02 | 42.02 | 3874 |
| m2.xlarge | 1.79 | 1.53 | 1.34 | 1.56 | 0.87 | 0.30 | 22.83 | 0.02 | 4.93 | 38.54 | 3762 |
| m1.xlarge | 1.86 | 1.58 | 1.34 | 1.78 | 0.96 | 0.31 | 38.20 | 0.02 | 7.34 | 101.43 | 3916 |
| c1.xlarge | 1.99 | 1.56 | 1.34 | 7.22 | 3.63 | 0.30 | 378.19 | 0.02 | 44.38 | 2169.40 | 3657 |
| m2.2xlarge | 1.82 | 1.55 | 1.33 | 1.60 | 0.88 | 0.31 | 29.02 | 0.02 | 5.11 | 45.75 | 4000 |
| m2.4xlarge | 1.86 | 1.58 | 1.34 | 1.71 | 0.92 | 0.31 | 26.51 | 0.02 | 5.20 | 44.28 | 3911 |

此外，我们可以观察到，由于更高的方差系数值，价格时间比比现货价格更加可变。 此外，修剪平均值的分析证实价格时间比有更大的变异性。因此，我们可能需要更高自由度的分布，来为这些迹线的价格时间比建模。值得注意的是，除了eu-west之外，所有数据中心的最小价格时间比几乎是一个小时，这是大约几分钟，可以在表3中看到。 此外，在西部数据中心，SI的设定价格平均稳定在不到两个小时，其他数据中心的持续时间约为23小时[9]。 所以我们设置3小时时间段以检查图2（a）中现货价格的模式。

## 6.分配配件

经过全球统计分析，我们首先检查现货价格和价格时间比的概率密度函数（PDF）。 然后，我们通过期望最大化（EM）算法对高斯混合分布（MoG）分布进行参数拟合，以建模两个时间序列。 我们考虑了其他分布，如Weibull，Normal，Log-normal和Gamma分布。 然而，高斯分布的混合显示出相对于其他的更好的适应性[9]。

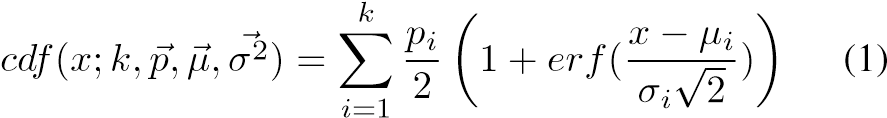
A.概率密度

在西方数据中心中每个SI的现货价格的PDF如图3所示。我们可以很容易地观察到概率密度函数中的双模态。 此外，所有SI的价格分布，除了m1.small，几乎是对称的。 m1.small的例外情况可能是因为此实例的不同使用模式是每个数据中心中最便宜的资源。

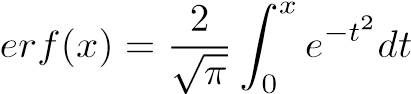
在图4中表示了在西欧的每个SI的价格时间比的PDF。显然，当与现货价格的PDF（几乎）相等的峰值相比时，在密度函数中存在单个主导模式（峰值）。大多数SI具有大约两个小时的峰，这确认了前一部分的结果（参见表3中的平均值栏）。在第七节中研究了这些密度函数中非常尖锐的峰的原因。 从两个时间序列的绘制的密度函数的观察，我们决定提出高斯分布的混合物作为近似这样的密度形状的良好候选者的进一步加强。

B.参数估计和拟合优度测试

在本节中，我们对具有k个分量的高斯分布的混合进行参数拟合，其定义如下：



其中，p是具有k个项的分量的均值，方差和概率的向量。 此外，erf（）是错误函数，其定义如下：



为了最大化在参数μ方面的数据似然性，并且其中k是先验给出的，我们采用期望最大化（EM）算法，其是一般的最大似然估计。使用基于模型的聚类（MBC）进行参数拟合，其由Fraley和Raftery [7]介绍。 MBC是一种方法论框架，可用于数据聚类以及（多）变量密度估计。一个假设是数据具有几个分量，每个分量由概率分布产生。基于模型的聚类使用贝叶斯模型选择来在部件数量方面选择最佳模型。相反，我们使用拟合优度（GOF）测试来确定最佳模型，因为我们对模型中的组件数量进行了估计。基于对密度函数的观察，我们选择2和4之间的分量数（2≤k≤4）。我们使用视觉方法（即标准概率概率（PP）图）和Kolmogorov-Smirnov（KS）和AndersonDarling（AD）检验作为定量度量来测量所得模型的拟合优度。

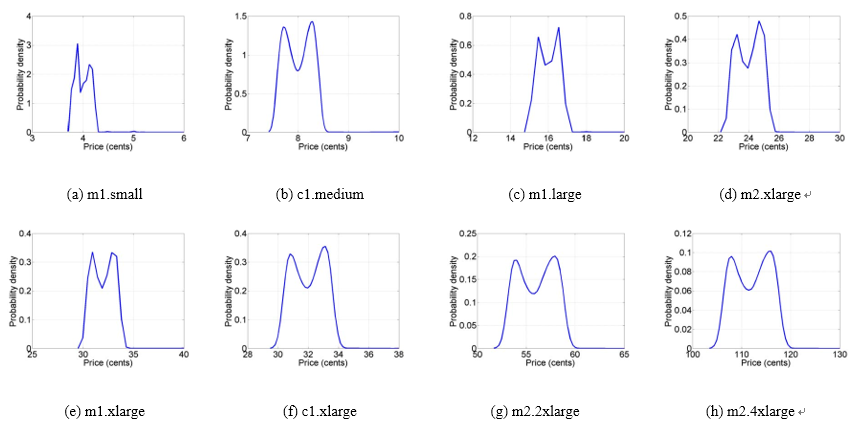


图3 在西部数据中心的所有SI的现货价格的概率密度函数

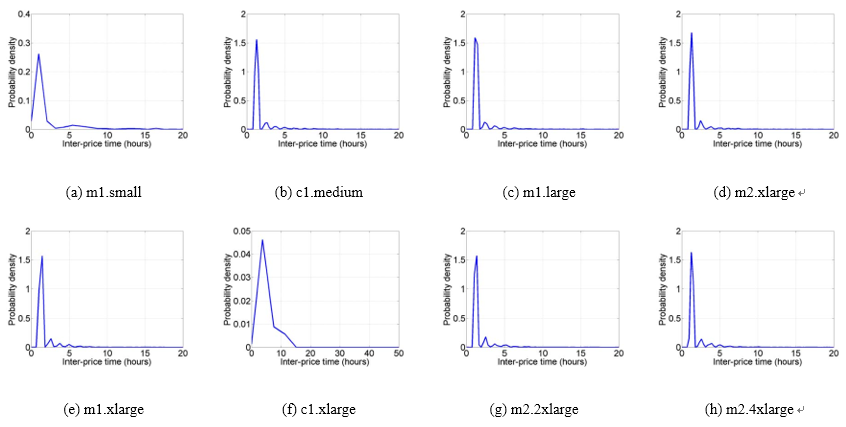


图4 在西部数据中心中所有SI的价格时间比的概率密度函数

在参数估计之后，我们必须通过GOF测试检查每个拟合的质量。 首先，我们分别给出了图5和图6中欧西数据中心的所有SI的现货价格和价格时间比的分布拟合的图形化结果。 在这些图中，图线越接近线y = x，拟合越好。 在每个图中，x轴是经验分位数，而y轴是拟合的分位数。 基于这些数据，高斯分布与三个或四个分量的混合可以适应现货价格和西南数据中心SI的价格时间比。 唯一难以适应的实例，特别是现货价格，是m1.small实例。

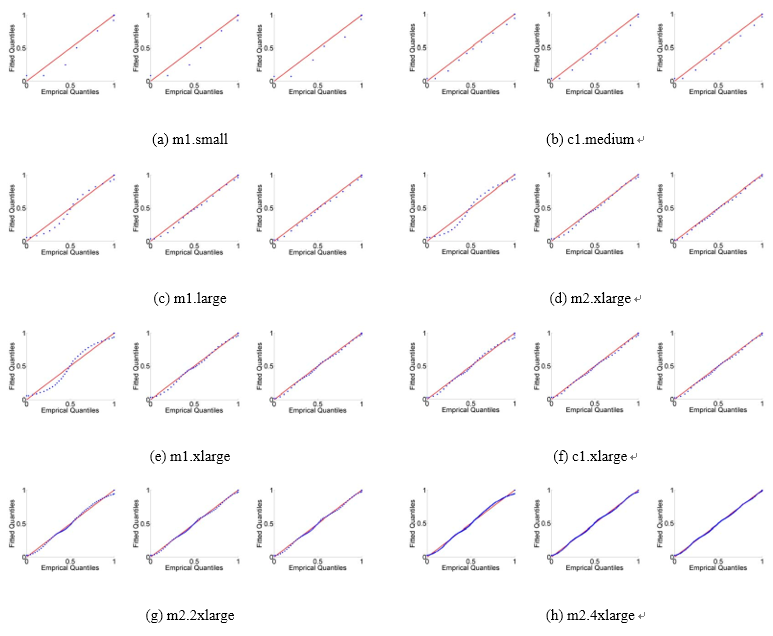


图5 高斯混合物（k = 2，k = 3，k = 4）在西部现货价格的PP图（X轴：经验分位数，Y轴：拟合分位数）

为了更定量，我们还报告了两个GOF测试（即KS和AD测试）的p值。 我们随机选择每个迹线的50个子样本，并迭代计算p值1000次，最后获得平均p值。 这种方法类似于作者在[10]中使用的方法。

GOF测试的结果分别列于表4和表5中现货价格和欧洲西部的价格时间比。 此外，在每行中突出显示最佳拟合。 在某些情况下，我们有两个获胜者，因为每个GOF测试有一个最佳拟合。这些定量结果强烈证实了PP图的图形结果。表4第一行中的p值表示m1.small实例的现货价格难以拟合，即使有四个分量。 这也是其他数据中心的情况，特别是对于us-east数据中心[9]。 由于MoG分布中的参数数量为3k + 1（参见等式1），因此我们在模型的准确性和复杂性之间进行权衡。 使用更少的组件，分析变得更简单，给出了在某种程度上牺牲精度的现货价格和价格时间比的合理适合性。 这将大大有助于理解第一步的数据系列。

通过这种理解，可以设计更好地适应具有许多组件的数据系列的模型。 因此，为了简化和均匀性，在本文的其余部分，我们选择具有三个分量（k = 3）的模型用于现货价格和价格时间比进一步分析。在所有数据中心中，现货价格的MoG分布参数和2≤k≤4的价格时间比报告在[9]中。

表4 来自KS的P值的AD测试价格

|  |  |  |  |
| --- | --- | --- | --- |
| Instances | MoG (*k* =2) | MoG (*k* =3) | MoG (*k* =4) |
| m1.small | 0.016 0.791 | 0.017 0.789 | 0.053 0.803 |
| c1.medium | 0.211 0.779 | 0.217 0.791 | 0.224 0.790 |
| m1.large | 0.113 0.678 | 0.319 0.752 | 0.354 0.754 |
| m2.xlarge | 0.139 0.616 | 0.356 0.721 | 0.415 0.734 |
| m1.xlarge | 0.134 0.570 | 0.369 0.708 | 0.431 0.706 |
| c1.xlarge | 0.394 0.681 | 0.444 0.705 | 0.421 0.707 |
| m2.2xlarge | 0.420 0.648 | 0.469 0.682 | 0.450 0.672 |
| m2.4xlarge | 0.429 0.617 | 0.463 0.637 | 0.476 0.653 |

## 7.模型校准

在本节中，我们研究现货价格和价格时间比的时间演变，这有可能导致我们获得更准确的模型。 为此，我们检查了2010年2月至2010年11月期间现货价格和价格时间比的散点图。由于空间限制，我们只提供m2.4xlarge实例的地块。 结果与数据中心内的其他实例类型一致。

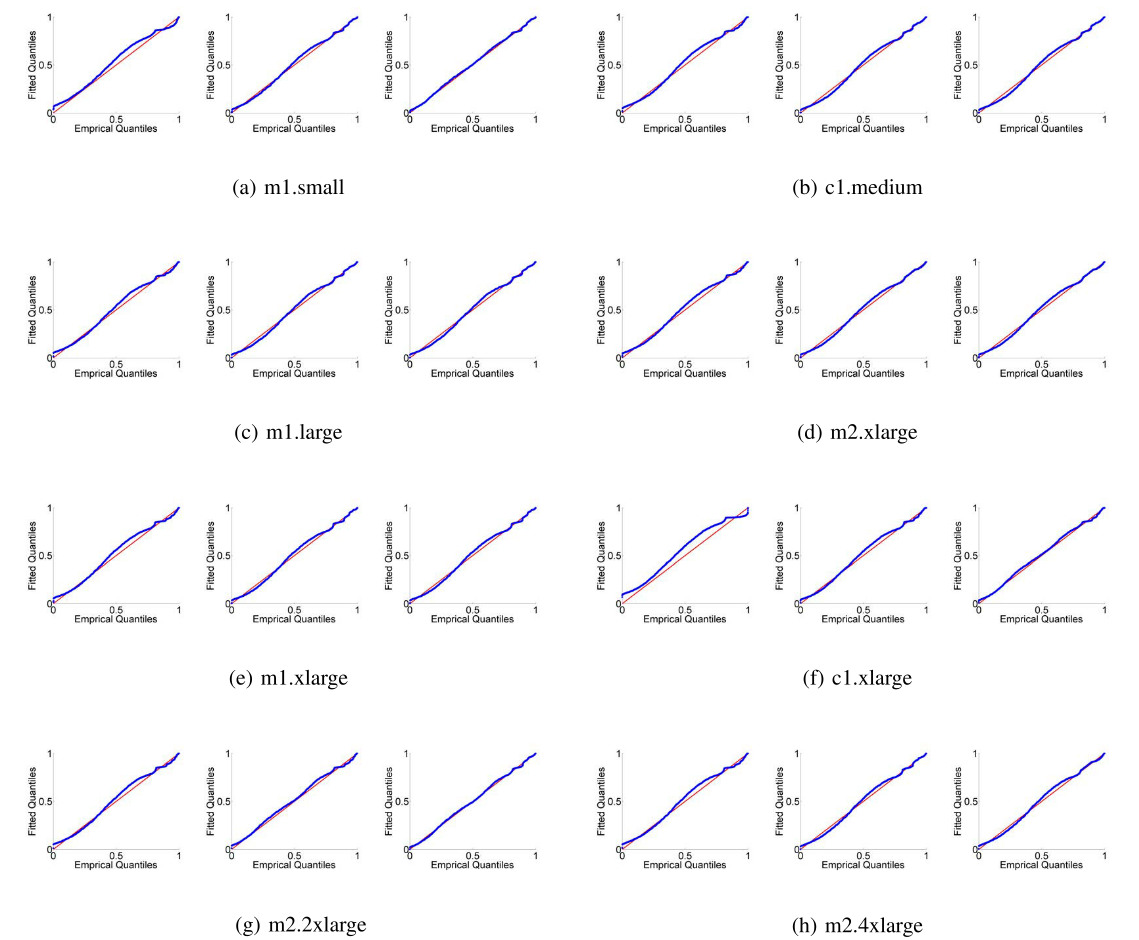


图6 高斯混合（k = 2，k = 3，k = 4）的西部价格时间比的PP图（X轴：经验分位数，Y轴：拟合分位数）

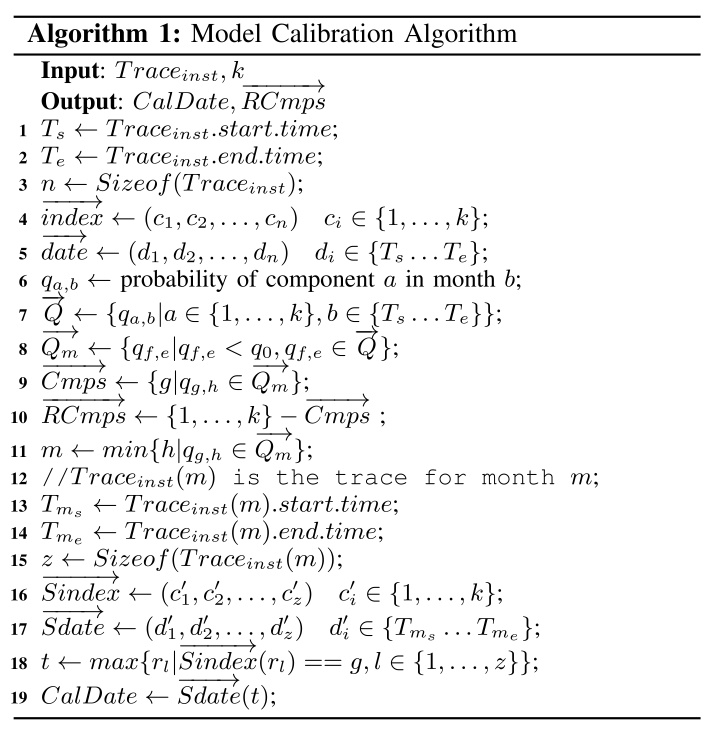


表5 来自KS的P值AD测试价格

|  |  |  |  |
| --- | --- | --- | --- |
| Instances | MoG (*k* =2) | MoG (*k* =3) | MoG (*k* =4) |
| m1.small | 0.347 0.476 | 0.415 0.592 | 0.489 0.627 |
| c1.medium | 0.382 0.546 | 0.390 0.566 | 0.380 0.566 |
| m1.large | 0.390 0.552 | 0.387 0.573 | 0.400 0.574 |
| m2.xlarge | 0.389 0.556 | 0.393 0.566 | 0.405 0.585 |
| m1.xlarge | 0.369 0.526 | 0.391 0.564 | 0.406 0.581 |
| c1.xlarge | 0.221 0.319 | 0.399 0.561 | 0.467 0.602 |
| m2.2xlarge | 0.376 0.532 | 0.426 0.570 | 0.463 0.610 |
| m2.4xlarge | 0.368 0.529 | 0.383 0.569 | 0.395 0.573 |

图7（a）描述了在价格历史期间西部数据中心中m2.4xlarge现货价格的散点图。 从图中可以看出，在现货价格中，它们在特定范围内均匀分布（这个范围取决于实例的类型），没有明确的相关性。 然而，现货价格的拥堵在7月中后增加，这是在西部数据中心的所有SI的情况。 为了证实这一观察，我们检查了图7（b）中该SI的价格时间比的散点图。 我们观察到，价格之间的时间在7月中旬之后突然缩短。 这意味着，价格变化的频率增加，而现货价格保持在小的价格范围内。 对数据中心内的其他SI的检查显示相同的结果。 这也是图4中价格时间比的密度函数非常尖锐的峰的原因。

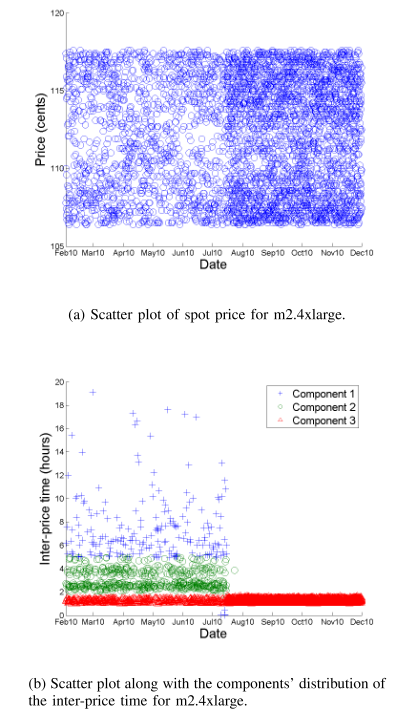


图7 m2.4xlarge的现货价格和价格时间比散点图

这种趋势可能是由于亚马逊在其定价算法中进行的一些细微调整。 值得注意的是，在其他亚马逊的EC2数据中心在不同的日期也观察到同样的问题。 在美国东部，发生在2010年8月，2011年1月在美国西部和东南部（数字绘制在[9]）。

关注图7（b）中给出的价格时间比（k = 3的MoG模型）的散点图，我们可以看出，在7月中旬之后，只剩下一个分量（即分量3），其他分量折叠成小带。由于这一观察在所有SI中是一致的，我们提出了一种模型校准算法（算法1）来找出折叠日期（其被称为校准日期）以及剩余分量。

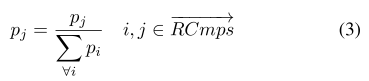
表6 模型校准结果（k = 3）

|  |  |  |
| --- | --- | --- |
| Instances | Calibration Dates | Remaining Components |
| m1.small | 24-July | 3 |
| c1.medium | 15-July | 1 |
| m1.large | 15-July | 3 |
| m2.xlarge | 13-July | 1 |
| m1.xlarge | 23-July | 1 |
| c1.xlarge | 23-July | 1 |
| m2.2xlarge | 23-July | 1,2 |
| m2.4xlarge | 15-July | 3 |

该算法需要跟踪SI（Traceinst）和组件数量（k）的价格时间比。高斯模型与k个分量的混合结果是指数。 另外，date是向量，其每个元素对应于索引的每个项目。 首先，算法计算整个轨迹中每个月份中每个分量的概率，然后找到一个列表（Qm），其中一个或多个分量的概率小于q0（第4-8行）。 q0是阈值，我们将其定义为低至0.01（即q0 = 0.01）。 不在此列表中的组件是剩余组件（第10行中的RCmps）。 列表中的第一个月是校准月，称为m（第11行）。 最后，在第m个月中组分的最后一次出现将是在第13-19行中获得的校准日期（CalDate）。

将该算法应用于欧西数据中心中的所有SI的结果在表VI中示出，其中所有校准日期在7月。 此外，对于除了m2.2xlarge之外的所有SI，校准日期后只剩余三个组件中的一个。

模型校准的最后一步是概率调整，其中剩余分量的概率必须被放大到一。 此调整可通过以下公式完成：



换句话说，在每个SI的校准模型中，我们只是改变校准日期后剩余分量的概率。 在下一节中，我们调查校准模型相对于实际价格历史以及未校准模型的准确性。

## 8.模型验证

为了验证提出的模型，我们使用CloudSim实现了一个离散事件模拟器[5]。 模拟器使用模型或价格历史记录来运行输入工作负载。 我们考虑用户从一种类型的SI请求一个VM并在该VM上运行整个作业的情况。 在SI上运行工作负载的总货币成本是要考虑的参数。

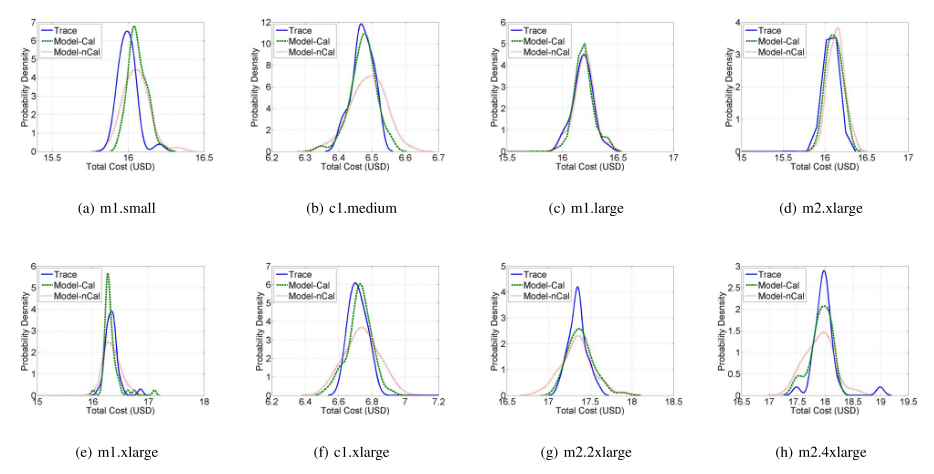


图8 模型跟踪中的所有SI的模型验证（2010年2月至2010年11月）

A.模拟设置

我们在实验中使用的工作负载是来自LCG Grid的工作负载跟踪，它来自Grid Workloads Archive [8]。 我们使用此跟踪的前1000个作业作为实验的输入工作负载，其长度足以反映不同SI的现货价格的行为。我们假设一个EC2计算单元相当于容量为1000 MIPS的CPU内核。则所选工作负载需要大约两周（约400小时）在单个m1.small实例上完成。 对于其他实例类型，我们考虑在EC2计算单元方面的计算能力的线性加速，它们列在表1中。对于每个实验，收集50个模拟循环的结果。

此外，我们假设对于每个模拟（例如按需价格），用户的出价非常高，其中在给定工作负荷的执行中我们没有任何出价事件。 我们使用具有三个分量（k = 3）的现货价格和价格时间比的模型来显示准确性和复杂性之间的交易。 在我们的实验中，模拟的结果是准确的，置信水平为95％。

B.结果和讨论

在下面，我们提出了两组不同的实验结果。 首先，我们讨论模型验证的结果，其中我们有包括在建模过程（即2010年2月至2010年11月）的价格历史。 其次，我们使用未包括在建模过程中的新的价格历史来报告模型验证的结果。 新的价格历史是从2010年12月到2011年2月中旬。

图8示出了模型验证结果，其中已经为所有类型的SI绘制了运行给定工作负荷的总货币成本的概率密度函数。 在每个图中，Trace，Model-Cal和Model-nCal分别指使用真实价格历史，校准后的模型和校准前的模型的结果。 基于这些图，所提出的模型以高精度匹配真实轨迹模拟，特别是用于校准模型。 如我们在这些图中可以看到的，在所有情况下，校准的模型与跟踪模拟更好地匹配。 正如我们所预期的，在m1.small实例的模型和跟踪模拟结果中存在差异。 然而，对于校准和非校准模型，对于所有SI，运行给定工作负载的平均总成本是非常准确的，其中最大相对误差分别小于3％。

此外，我们报告模型验证结果，其中我们使用2010年12月至2011年2月中旬的新价格历史，查看未来踪迹的模型质量。 新的价格历史的模拟结果绘制在图9中。结果表明，我们的三个组件的模型仍然符合跟踪模拟结果，除了m1.small实例。 如前所述，m1.small实例的现货价格很难适应，这是这种不准确的原因。 这意味着对于m1.small，我们应该使用具有更多分量的模型（例如k = 4）来获得更好的精度。 校准的模型再次匹配跟踪模拟与所有SI的非校准模型相比更好。 此外，对于校准和非校准模型，所有SI的平均总成本的最大相对误差小于4％。 因此，所提出的模型对于新的价格历史也是足够准确的。

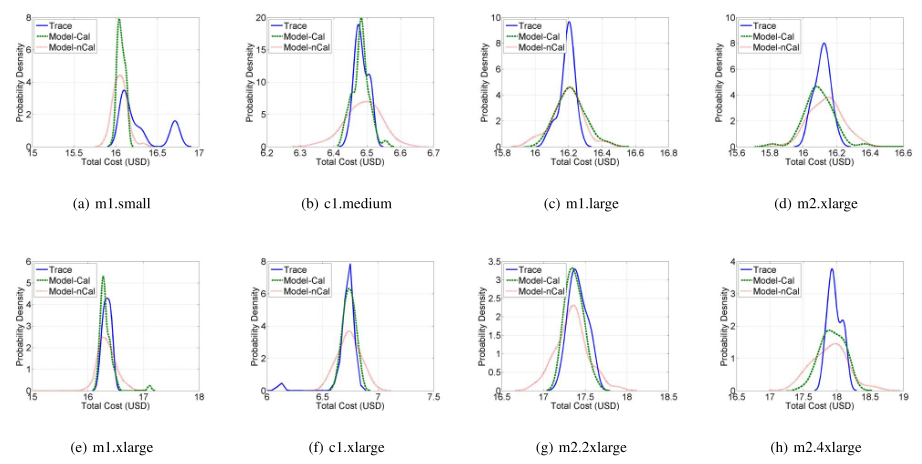


图9 新西兰地区所有SI的模型验证（2010年12月至2011年2月中旬）

我们考虑了在Amazon的EC2数据中心中为现货价格和价格时间比发现竞价型实例的模型的问题。其背后的主要动机是探索在设计随机调度算法和容错机制（例如检查点和复制算法）在现代市场的云环境中必不可少的SI的特征。我们研究了亚马逊数据中心的价格模式一年，并提供了全球统计分析，以更好地了解这些模式。基于这种理解和观察到的概率密度的双峰性，我们提出了一种具有高斯分布的模型，其具有用于八种不同类型的SI的3个或4个分量。提出的模型通过模拟验证，显示我们的模型预测在现场实例上运行作业的总价格具有良好的准确性。我们认为，所提议的模型有助于亚马逊EC2数据中心的现场实例的研究人员和用户，以及期待在不久的将来提供这样的服务的其他IaaS云提供商。

在未来的工作中，我们打算将用户的出价作为另一个参数，并研究它如何影响故障的分布。 此外，我们希望设计一个代理解决方案，利用不同类型的云资源来优化货币成本以及完成工作时间。

参考文献

1. Cloud exchange website. http://cloudexchange.org/.
2. Amazon Inc. Amazon Discussion Forums. https://forums.aws.amazon. com.
3. Amazon Inc. Amazon Elastic Compute Cloud (Amazon EC2). http:

//aws.amazon.com/ec2.

1. A. Andrzejak, D. Kondo, and S. Yi. Decision model for cloud computing under SLA constraints. In *18th IEEE/ACM International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 257–266, 2010.
2. R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience*, 41(1):23–50, 2011.
3. N. Chohan, C. Castillo, M. Spreitzer, M. Steinder, A. Tantawi, and C. Krintz. See spot run: using spot instances for MapReduce workflows. In *the 2nd USENIX conference on Hot topics in cloud computing*, HotCloud’10, pages 7–7, 2010.
4. C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2002.
5. A. Iosup, H. Li, M. Jan, S. Anoep, C. Dumitrescu, L. Wolters, and D. H. J. Epema. The Grid Workloads Archive. *Future Generation Computer Systems*, 24(7):672–686, 2008.
6. B. Javadi and R. Buyya. Comprehensive statistical analysis and modeling of spot instances in public Cloud environments. Research Report CLOUDS-TR-2011-1, Cloud Computing and Distributed Systems Laboratory, The University of Melbourne, March 2011.
7. B. Javadi, D. Kondo, J.-M. Vincent, and D. P. Anderson. Mining for statistical availability models in large-scale distributed systems: An empirical study of SETI@home. In *17th IEEE/ACM International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 1–10, 2009.
8. D. Kondo, A. Andrzejak, and D. P. Anderson. On correlated availability in internet distributed systems. In *9th IEEE/ACM International Conference on Grid Computing (Grid 2008)*, pages 276–283, 2008.
9. M. Mattess, C. Vecchiola, and R. Buyya. Managing peak loads by leasing cloud infrastructure services from a spot market. In *12th IEEE International Conference on High Performance Computing and Communications*, pages 180–188, 2010.
10. J. Varia. *Cloud Computing: Principles and Paradigms*, chapter 18: Best Practices in Architecting Cloud Applications in the AWS Cloud, pages 459–490. Wiley Press, 2011.
11. S. Yi, D. Kondo, and A. Andrzejak. Reducing costs of spot instances via checkpointing in the amazon elastic compute cloud. In *3rd IEEE International Conference on Cloud Computing*, pages 236 –243, 2010.