

云计算环境下的弹性结构程序研究

陆建华¹, 王旻超², 武 星²

(1. 上海科技管理干部学院电子信息系, 上海 201800; 2. 上海大学计算机工程与科学学院, 上海 200072)

摘 要: 针对云计算环境下云程序的弹性结构进行研究, 结合 Anytime 算法、弹性计价模型和软件即服务层中的 SLA 计费架构, 将总业务费用分为接入费用、使用费用和补偿费用进行计算, 设计基于图像小波变换的弹性结构程序, 给出程序的执行步骤和状态转换图。分析结果表明, 该弹性结构程序能对程序迭代次数进行控制, 较好满足用户需求。

关键词: 云计算; Anytime 算法; 弹性程序; 云程序; 服务等级协议

Research on Elastic Structure Program Under Cloud Computing Environment

LU Jian-hua¹, WANG Min-chao², WU Xing²

(1. Department of Electronic Information, Shanghai Institute of Science & Technology Management, Shanghai 201800, China;

2. School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China)

【Abstract】 This paper researches the elastic structure of the cloud program in the cloud computing environment. It combines the Anytime algorithm, an elastic billing model and a Service Level Agreement(SLA) framework in Software as a Service(SaaS), implements an elastic structure cloud program which is based on the wavelet algorithm and presents the working process and state transition diagram. Analysis result shows that the elastic structure program can control program iterative times according to user demand, better satisfy the needs of clients.

【Key words】 cloud computing; Anytime algorithm; elastic program; cloud program; Service Level Agreement(SLA)

DOI: 10.3969/j.issn.1000-3428.2012.21.013

1 概述

随着互联网时代信息数据量的快速增长, 计算机硬件架构所能提供的计算能力早已无法满足大规模海量数据的科学研究计算。不仅如此, 摩尔定律^[1]的逐渐失效也验证了并行计算正渐渐成为日后科学计算的主要发展趋势。而云计算是分布式计算、并行处理和网格计算的进一步发展, 通过互联网向用户提供基础架构服务、平台服务、软件服务, 其核心思想是将本地已有的资源通过虚拟化技术, 通过互联网以出租形式提供服务。早在 20 世纪 60 年代, McCarthy J 指出^[2]计算迟早有一天会变成一种公用基础设施, 即将计算能力作为一种像水和电一样的公用事业提供给用户。在 IBM 和 Google 宣布在云计算领域合作之后, 云计算得到了广大学者的关注, 并迅速成为产业界和学术界的研究热点。其中, 在产业界较具有代表性的云计算系统主要有: Amazon EC2, Google App Engine, Apache Hadoop 等; 而 OpenNebula 与 Scientific Cloud 是学术界中较为著名的科研项目。

计算机科学产业的不断进步推动了计算机程序的巨大变革, 从串行程序到并行程序, 再逐渐演变为云程序。

在 1976 年, Wirth N 提出串行程序的主要结构为: 算法+数据结构=程序^[3]。随着计算机网络技术不断发展与革新, 计算机的计算方式也逐步向网络计算、分布式计算、并行计算进行过渡。云弹性是云计算最显著的特点之一, 所谓云弹性是指云资源可以根据用户的需求进行动态的资源调整。不仅云可以是弹性的, 云程序也可以具有弹性的结构。一个弹性程序的结构不同于传统结构, 弹性结构程序的最大特点是可以在运行的任意时刻终止并能够有相关的结果输出。本文对云程序弹性结构的设计与分析进行研究, 对弹性经济模型和云计算中软件即服务(Software as a Service, SaaS)层中的服务等级协议(Service Level Agreement, SLA)计费架构进行分析。实现一个应用于图像小波变换中的弹性云程序, 程序可对图片进行相应缩放, 程序可以在任意时刻停止。最后对程序的执行步骤和状态转换图做出分析。

2 弹性经济模型与 SLA 设计

2.1 弹性经济模型

云计算^[4]是以付费使用的形式向用户提供各种服务的分布式计算系统, 所谓云计算的经济模型, 是符合基本的

作者简介: 陆建华(1963—), 男, 讲师、硕士, 主研方向: 云计算, 电子政务; 王旻超, 硕士研究生; 武 星, 讲师、博士

收稿日期: 2012-04-09 **修回日期:** 2012-05-24 **E-mail:** lujianhua@hotmail.com

价值与价格规律(云计算资源的价格围绕使用资源的价值波动, 时间序列函数)并考虑到用户需求模型(在基本价格基础上, 同时请求云计算资源的用户越多, 资源价格随之增加)。根据对云计算经济模型定义的了解, 结合 Anytime 算法^[5], 提出一种新的云计算经济模型, 即为弹性经济模型。所谓弹性经济模型, 其思想是随着用户投入成本的增加, 所给出问题的解的质量不断提高。

当用户操作云程序时, 程序会根据用户的需求(CPU 个数、内存大小、带宽大小等)以及当前云需求量, 计算出所需费用, 然后根据用户所给费用给予最近似的计算值。这里的所需费用是指根据一种云计算经济模型所得出的范围值, 而不是一个单一的数值。

根据云程序的价格变化规律, 对云程序价格曲线函数做如下定义:

(1) 价格变化曲线 $P(t)$ 是一个严格单调递增函数, 满足 $\forall t_1, t_2 \in N$, 且 $t_2 > t_1 > 0$, 恒有 $P(t_1) < P(t_2)$ 。

(2) 对于函数 $F(t) = P(t)$, 有 $\forall t_1, t_2 \in N$, 且 $t_2 > t_1 > 0$, 恒有 $F(t_1) > F(t_2)$ 。

(3) 对于函数 $\lim_{x \rightarrow \infty} P(t) = C$, 其中, C 是一个大于 0 的常数。

由此可以定义云程序价格变化曲线, 如图 1 所示。利用上述云程序价格变化曲线, 可以得到一个价格范围: [BottomPrice, TopPrice]。

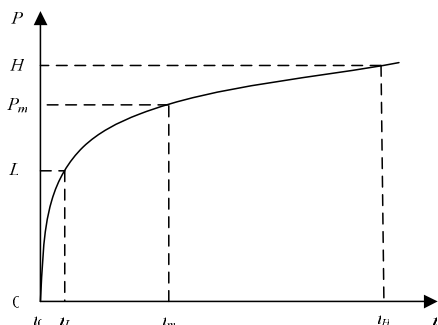


图 1 云程序价格变化曲线

BottomPrice 是运行云程序所需要的最低价格, 即用户操作云程序所需支付的费用必须高于这个价格, 否则程序无法运行; TopPrice 是运行云程序的最优解价格, 当用户操作云程序支付的费用高于这个价格时, 云程序将返回最优解。

通过上述的定价模型, 可以精确地得到云程序执行每一次迭代的所需的价格。云程序可以根据用户所支付的费用进行相应次数的迭代计算, 最终返回给用户最近似的解。

2.2 SLA 架构设计

服务级别协议^[6]是指提供服务的企业与客户之间就服务的品质、水准、性能等方面所达成的双方共同认可的协议或契约。SLA 是一项介于云供应商与云消费者之间并且独立于两者的重要第三方协议^[7], 如图 2 所示。

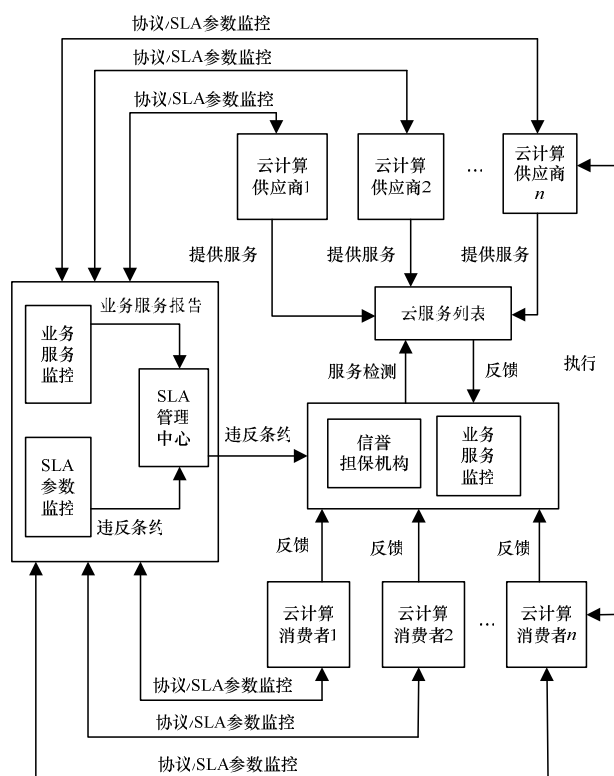


图 2 适用于云计算的 SLA 架构

SLA 的存在可以对云供应商的服务质量起到有力的监控, 一个清晰的 SLA 描述给云消费者提供了重要的保障。不仅如此, SLA 中也应该包含对于出现违反协议情况时的应急措施与补救办法。

对于云计算而言, 因为云供应商所提供的服务形式与其网络服务供应商所提供的有较大差别, 所以现有的一些 SLA 架构并不完全适用于云计算服务。正如上文提到的, 云计算通过互联网向用户提供基础架构服务、平台服务、软件服务、存储服务等一系列服务, 那么对于云计算中所提供的不同服务, SLA 中所涉及的内容也应该有较大的不同, 如图 3 所示。

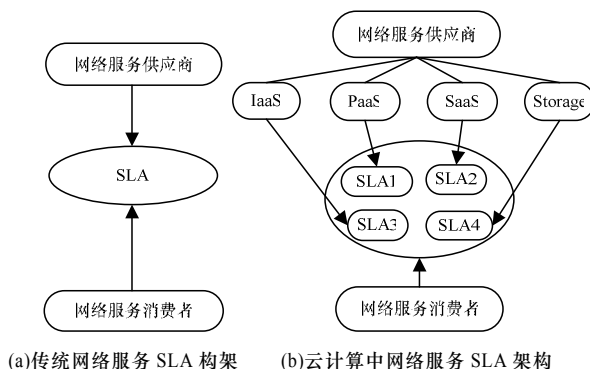


图 3 传统网络服务与云计算网络服务的 SLA 比较

通过图 3 可知, 对于云计算中的每一种服务, 都应该有一项单独的 SLA 协议内容与之对应。因为在这里是对云计算中的软件服务进行研究, 所以将对云计算中的 SaaS 所对应的 SLA 协议做进一步研究。而对于云计算中 IaaS 和 PaaS 层的 SLA 计算方式, 则不在本文的讨论范围内。

下面对 SaaS 层的 SLA 计费方式进行分析说明。对于云计算中 SaaS 层的计费, 将其设定成由接入费用 C_a 、使用费用 C_u 和补偿费用 C_p 三部分组成, 计算公式下:

$$E = C_a + C_u + C_p$$

其中, C_a 表示用户的接入费用。因为在用户接入后, 云环境将为用户预定一部分资源, 所以无论用户是否使用此资源, 都应为此资源支付一定的费用。接入费用的计算公式如下:

$$C_a = \alpha \times T$$

其中, α 为接入因子, 单位为元/秒; T 为接入时间, 单位为秒。对于接入因子 α 的设定, 要让接入费用比最低服务质量的业务费用还要低, 同时, 使接入费用有一定的威慑力, 让用户在不使用网络时, 自觉地退出^[8]。

如上文所述, 对于使用费用 C_u 的定价将根据云价格变化曲线给出用户一个价格范围, 这个价格范围是根据用户所需配置的要求、所需解的要求以及当前使用人数所综合计算所得的。用户可以根据这个范围给出一个合适的价格, 而这个合适的价格就是使用费用 C_u 的定价。

对于补偿费用 C_p , 其主要作用是对没有达到承诺服务质量的业务进行补偿。软件服务是云计算所提供服务的形式, 即将云平台或云架构中自带的软件提供给云消费者使用。针对软件服务的质量, 可以对以下几个方面进行评定^[9], 如表 1 所示。

表 1 云计算中 SaaS 层的评测类型与描述

| 评测类型 | 描述 |
|--------|--------------------|
| 软件可靠性 | 评测软件在常规环境下的运行能力 |
| 软件实用性 | 评测软件的操作性和交互性 |
| 软件可扩展性 | 评测软件对使用用户人数增加的可适应性 |
| 软件实时性 | 评测软件的可获取性 |
| 软件通俗性 | 评测软件对不同使用人群的可适应性 |

在上文已经提到, SLA 是一项介于云供应商与云消费者之间并且独立于两者的重要第三方协议, 所以, SLA 协议可以通过上述几方面对软件进行评测, 然后通过 SLA 监控服务给出相应补偿费用的定价。

3 弹性云程序的实现

在日常生活中, 有时需要对于图片进行放大或缩小操作, 但有时并不是对整张图片的细节都感兴趣, 而是对部分图片的细节感兴趣, 那么只需要针对这部分细节进行操作就可以达到要求。在此背景下, 以上述的一种经济模型为基础, 结合 Anytime 算法和 SaaS 层的 SLA 架构, 实现一个应用于图像小波变换中的弹性云程序。程序的弹性特点主要体现在程序可以根据用户的需求及其支付费用对图片进行相应的缩放, 当用户支付费用扣尽或者已经达用户需求时, 程序自动停止, 给出相应的结果。

在实验中, 选取 Luna 作为实验图片。通过“位图选择”按钮, 对 Luna 的头部部分进行选择, 如图 4 所示。通过“局部放大”按钮, 可以对所选部分进行选择性放大,

放大的次数是根据用户需求或支付所用所决定的。而对于每次放大的费用, 是根据程序当前使用人数、使用 CPU 个数、使用内存大小等方面共同决定的。

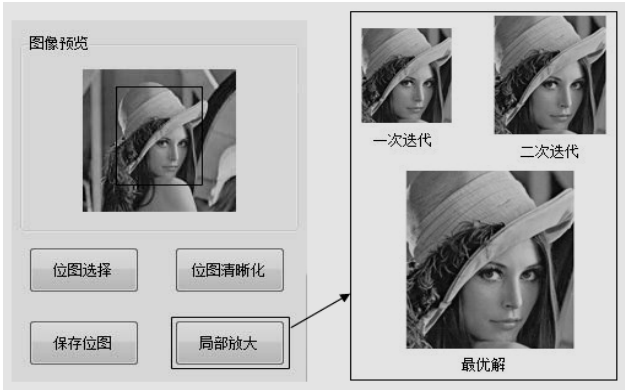


图 4 云程序实验结果

云程序开始执行时会根据当前云环境的使用人数确定此时的工作时段为闲时还是忙时, 然后提示用户输入所需的配置要求, 最后载入用户所需的图片。在云程序完成启动时的初始化后, 用户可以在图片上对部分细节进行局部放大等操作。

当用户完成操作后, 云程序会根据用户需求连接数据库, 提取完成一次小波变换计算的费用和完成最优解计算的费用并返回给用户, 用户可以根据所需结果支付不少于完成一次小波变换计算的费用。接着程序会提取进行下一次小波变换所需的费用, 2 次费用进行累加。若用户支付的费用仍大于此费用, 则程序继续提取进一步小波变换的费用, 进行累加, 直到累加费用大于用户支付费用, 并记录当前的小波变换迭代次数, 对用户选择部分进行小波变换的迭代计算。在完成计算后, 将当前的迭代值返回给用户。用户可以根据对当前解的情况选择继续或者终止, 从而获得最满意的解。图 4 右侧显示了经过了不同次数放大后的图片效果。

4 分析结果

在第 3 节中, 实现了一个基于小波变换的弹性云程序。在这节中将对所实现的程序做详细分析。程序可以根据用户需求给出相应结果, 程序执行状态转移如下:

步骤 1 云程序处于空闲状态, 等待用户登录并提交所需计算的问题。

步骤 2 获取用户所需计算问题, 等待用户提交 CPU 个数、内存大小、带宽大小等配置要求, 并进行相应的判断。若用户取消, 则程序退出, 返回空闲状态; 若用户输入有误或当前云环境无法满足用户所需要求, 则进入错误状态并通知用户, 等待用户返回确认; 否则, 程序进入下一正常状态。

步骤 3 根据用户所需 CPU 个数、内存大小、带宽大小等相关配置要求, 进行云程序初始化和当前云环境的配置。若初始化完成, 则进入下一正常状态。

步骤 4 根据用户所需计算问题和所需相关配置要求

连接数据库,读取进行一次计算的所需费用 L 以及计算至最优解的所需费用 H ,设置云程序中相关参数。

步骤 5 将计算所需的最低费用 L 与最高费用 H 返回给用户,程序进入挂起状态,等待用户支付费用 P 。若用户取消,则程序进入空闲状态。

步骤 6 获取用户支付费用 P ,计算 P 与 L 之间大小,若 P 小于最低计算费用 L ,则进入出错状态,退回用户支付费用 P 并通知用户,等待用户确认返回;否则计算 P 与 H 之间大小,若 P 大于 H ,则直接返回最优解给用户;否则进入下一状态。

步骤 7 从数据库中获取进行 N 次迭代所需的费用 P_1, P_2, \dots, P_N , 并计算进行到第 N 次迭代的总费用 $S_N = \sum_{i=1}^N P_i$, 比较 S 与 P 的大小。如果 $P > S$, 那么进一步计算进行到第 $N+1$ 次迭代的总费用 S_{N+1} , 直到 S_N 大于 P ; 否则对用户提交问题进行 N 次迭代,将当前迭代值返回给用户,并且提示用户是否需要进一步计算,程序进入挂起状态,等待用户返回。

步骤 8 获取用户返回信息。若用户确认返回,则程序跳转至步骤 4; 否则程序进入空闲状态。

弹性云程序的状态转移如图 5 所示。

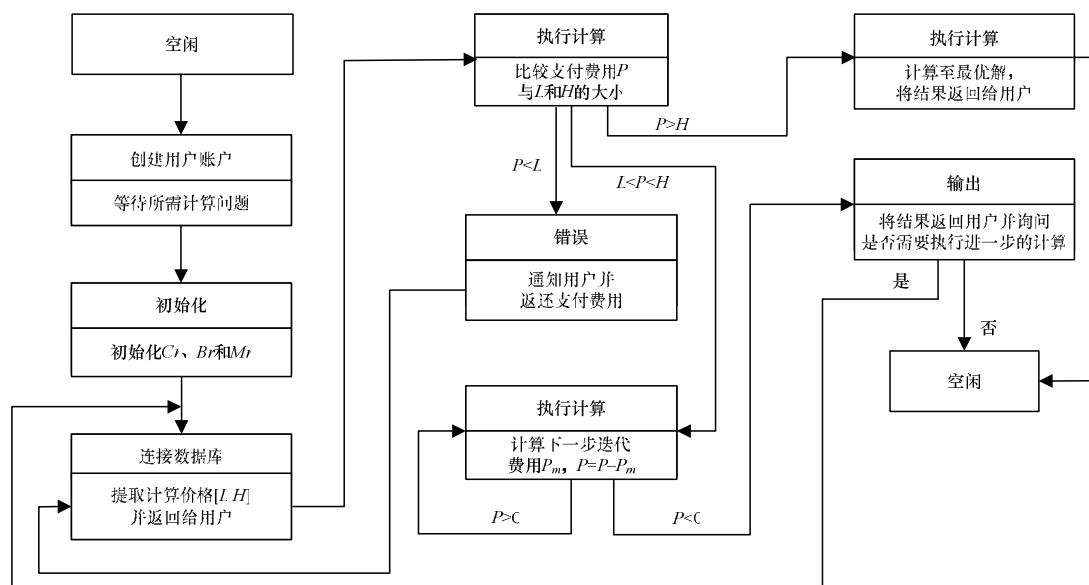


图 5 弹性云程序的状态转移

5 结束语

本文首先对弹性的经济模型做了简单概述与介绍,随后又针对云计算中 SaaS 层的特性,对适用的 SLA 计费架构进行分析。这种计费架构的主要思想是将总的业务费用分割成接入费用、使用费用和补偿费用 3 块相独立的费用进行计算。然后通过实现一个应用于图像小波变换中的弹性云程序,对云程序弹性结构的实现过程进行分析。通过分析,了解云程序的弹性特点,可动态根据用户需求对程序的迭代次数进行控制,并能够给出当前迭代结果。

今后将对以下工作做进一步研究:(1)针对弹性程序结构的特点,对弹性经济模型进行深入研究,提出一种完善的适用于弹性结构程序的计费模型。(2)针对云计算中 SaaS 层的特点,对 SLA 架构进行深入研究,并对弹性结构程序的 SLA 计费架构进行研究。

参考文献

- [1] Wikipedia. 摩尔定律[EB/OL]. (2010-10-26). <http://zh.wikipedia.org/wiki/%E6%91%A9%E5%B0%94%E5%AE%9A%BE%8B>.
- [2] Wikipedia. John McCarthy[EB/OL]. (2011-04-08). <http://en.Wiki>

[pedia.org/wiki/John_McCarthy_\(computer_scientist\)](http://en.Wiki).

- [3] Niklaus W. Algorithms+Data Structures=Programs[M]. [S. l.]: Prentice Hall, 1976.
- [4] 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.
- [5] Zilberstein S. Using Anytime Algorithms in Intelligent Systems[J]. AI Magazine, 1996, 17(2): 73-83.
- [6] Wikipedia. 服务级别协议[EB/OL]. (2010-10-26). <http://zh.wikipedia.org/wiki/%E6%9C%8D%E5%8A%A1%E7%BA%A7%E5%88%AB%E5%8D%8F%E8%AE%AE>.
- [7] Alhamad M, Dillon T, Chang E. SLA-based Trust Model for Cloud Computing[C]//Proc. of the 13th International Conference on Network-based Information Systems. [S. l.]: IEEE Press, 2010.
- [8] 张登银, 李正, 程春玲. 基于 SLA 的下一代网络计费方法[J]. 计算机应用研究, 2009, 26(7): 2713-2715.
- [9] Alhamad M, Dillon T, Chang E. Conceptual SLA Framework for Cloud Computing[C]//Proc. of the 4th IEEE International Conference on Digital Ecosystems and Technologies. [S. l.]: IEEE Press, 2010: 606-610.

编辑 陆燕菲