



# 关于云计算可用性的 定性与定量研究

A Qualitative and Quantitative Study on  
Availability of Cloud Computing

---

陈怀临 著

## 摘要



云计算服务在被越来越多的个人和企业所采用，但人们对于云计算服务在安全性，可靠性和服务响应确定性方面的担忧也与日俱增。虽然云服务提供商 (Clouds Service Provider) 通常都会承诺 SLA (Service Level Agreement) 的可用性 (Availability) 范围等，但许多云租户不太理解可用性的内在复杂性，因此在选择云平台时缺乏风险评估能力。

本文首次系统的定义和分析了云计算可用性 (Cloud Computing Availability) 的层次模型，提出了可用单元 AU (Availability Unit), 可用集合 AS (Availability Set), 单独可用性 SA (Standalone Availability) 和部署可用性 DA (Deployment Availability) 等概念，并对云计算 IaaS, PaaS 和 SaaS 各个层次 SA 和 DA 的算法关系展开了讨论。在此基础上，文章提出了扁平 AS (Flat AS) 和复合 AS (Composite AS) 为基本单元的参考设计模型，其中包括一字型架构，菱形架构，人字型架构，三角形架构和胖树架构。并对各种模型优缺点展开了相关的定性分析。文章最后，对亚马逊的 AWS 公有云体系结构做了相应的讨论，并针对 2006 年到 2013 年以来 AWS 被外界所报道过的服务事故做了相应的统计调查和一些定量分析。

关键词：云计算，数据中心，AWS, SLA, 可用性，可用单元，可用集合，扁平结构，复合结构

## 致谢



特别感谢北极光创投，宽带资本、云基地提供了各方面的支持和鼓励。

对中云传媒的吴曼女士，顾洪文先生，覃宾先生和佟玲女士在笔者写作期间提供的帮助和鼓励深表谢意。

另外，非常谢谢山石网络科技研发副总裁蒋东毅先生，高级研发总监尚进博士，99Cloud CEO 张淳先生，MoPaaS Cloud CEO 鲁为民博士，盛大云副总裁姜林博士，TryStack China 技术顾问蒋清野先生对文章的审阅，讨论和提供了一些珍贵的相关资料。

最后，也非常感谢许多朋友，包括弯曲评论网友在笔者写作期间对文章中错误之处的指出和提供了许多宝贵的意见。

# 目录

<b>1. 云计算的挑战</b>	01
<b>2. 云计算可用性</b>	04
2.1 可用性	04
2.2 云计算分层结构可用性	09
2.3 云计算分层结构拓扑	16
<b>3. 云计算参考设计模型</b>	24
3.1 一字型架构	24
3.2 菱形架构	25
3.3 人字型架构	26
3.4 三角形架构	27



3.5 胖树架构 ..... 28

**4. 案例研究 —— 亚马逊 AWS ..... 30**

4.1 AWS 系统结构 ..... 30

4.2 AWS 部署模式 ..... 34

4.3 AWS 的高可靠性 ..... 36

4.4 AWS 服务宕机调查 (2006-2013) ..... 41

4.5 AWS 服务宕机数据统计分析 ..... 70

**5. 讨论和总结 ..... 73**

**参考文献 ..... 74**

# 1. 云计算的挑战：

云服务在被越来越多的企业所采用。据 Gartner 预测，2013 年公有云的市场份额将会以 18% 的增长率从 2012 年的 1110 亿美金增长至 1310 亿美金，如图 1 所示。



图 1 公有云服务市场和年增长率

在 IaaS (Infrastructure as a Service) 方面，增长速度为 47.3%，市场份额为 90 亿美金。2012 年，IaaS 增长了 42.4%。2016 年，公有云的市场大小会达到 2100 亿美金，增长率为 17.7%，而在 IaaS 方面会保持 41.3% 的增长率 [1]。

然而，随着大量中小型企业 CIO 在考虑把公司的数据和应用迁移到云计算平台上，伴随而来的是对云计算的服务质量 (Quality of Service) 的担忧。

UC Berkeley 计算机系 RAD 实验室的 Michael Armbrust 等在 2009 年 2 月发表了关于对云计算服务的论文——“Above the Clouds: A Berkeley View of Cloud Computing”。文中 Berkeley 提出了其理解的云计算概念模型，并提出了云服务必须克服的 10 大障碍 [2]，如图 2 所示。

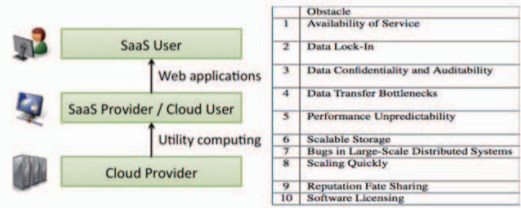


图 2 Berkeley 的云计算模型

[1] 此类标注请对照查看最后的参考文献

在这 10 大障碍中, 1(Availability of Service), 2(Data Confidentiality and Auditability), 5(Performance Unpredictability), 6(Scalable Storage), 7(Bugs in Large-Scale Distributed Systems), 8(Scaling Quickly) 都与云计算质量紧密相关. Berkeley 在对可用性 (Availability) 的解释中, 还特别提到了 DDoS 攻击对云计算带来的危害和需要防范的措施.

另外, 据来自 Newvem 的调查数据报告, 有 35% 的亚马逊的 AWS 用户对宕机基本上没有防范措施; 40% 的 AWS 用户没有定期做数据的备份. TeamQuest 最近对许多企业的 CIO 做了一次调查, 接受调查的 CIO 有 40% 的表示他们在使用云计算的时候发生了机群宕机现象 [3].

2012 年, 许多著名的公有云计算数据中心都发生了重大的安全事故. 下面是一些典型的案例 [4][5]:

\*2012 年 2 月 29 日和 7 月 26 日, 微软的 Azure 发生事故, 时间分别为长达 9 个小时和 2.5 个小时, 许多北美和欧洲的用户无法正常管理和使用其公司正常业务, 有的彻底丢失了他们最新的数据.

\* 2012 年 6 月 14 日, 6 月 29 日, 10 月 22 日和圣诞节期间的 12 月 24 日, 亚马逊 AWS 发生了严重云服务缓慢和崩溃无法访问的问题, 影响的租户包括许多重要的互联网公司, 例如 Netflix, pInterest, twitter, Instagram 等等 [4]. 每次事故导致用户无法正常使用服务的时间长达 9 个小时和更多.

\* 2012 年 7 月 10 日, 著名的 SaaS(Software as a Service) 公司 Salesforce 的服务出现重大停顿. 其原因是提供 Salesforce 公司 IaaS 服务的公司 (Equinix) 的数据中心电源失效. Equinix 据说在 1 分钟内就恢复了电源. 但 Salesforce 花费了接近 9 个小时来完整的恢复其相关业务.

\* 2012 年 9 月 10 日, 著名的 DNS 服务提供商 GoDaddy 的数据中心服务暂停. GoDaddy 管理着接近 5 千万个域名和 5 百万个 WEB 站点. 这次服务无法正常使用的长达 7 个小时. 其原因被解释为路由器的数据被破坏. 也有媒体报道是 GoDaddy 遭遇到了强大的 DDoS 攻击. 但这一声称被 GoDaddy 否认.

\* 2012 年 10 月 26 日, 谷歌的 App Engine 云服务出现暂停, 时间长达 4 个小时. 事后谷歌没有发表具体原因解释.

\* 2012 年 10 月 26 日, 著名的云存储提供商 Dropbox 的服务出现暂停, 时间长达 10 个小时. 其具体原因不详.

由上可见, 伴随着云计算本身具备的无可争议的巨大价值, 云计算带来的诸多服务质量问题也正变得越来越明显.

因此对云计算的可用性的定性和定量分析逐渐变为一个兼有研究和工程价值的问题，有助于帮助 CIO 们评估一个云计算平台。

目前学术和工业界对云计算，特别是公有云的可用性方面还没有引起足够的重视，缺乏这方面的定性和定量分析工作。

本文首次系统的定义和分析了云计算可用性的算法模型，特别是对云计算的 IaaS, PaaS 和 SaaS 各个层次可用性的内在关系展开讨论。文章的最后，针对 2006 年到 2013 年以来 AWS 被外界所报道过的服务事故做了相应的统计调查和一些定量分析。

## 2. 云计算可用性 (Cloud Computing Availability)

云计算可用性是一个很广义的概念。本文定义云计算可用性如下：

**云计算可用性：**包括 IaaS, PaaS 和 SaaS 各个层面服务的连接, 可靠性, 延时, 数据泄露和丢失, 网络攻击以及其他任何意外而导致租户的业务不能满足期望, 或者更严重的业务完全暂停。云服务商通常是通过 SLA(Service Level Agreement) 来量化可用性的承诺, 给出相应的 Availability 的数值范围, 例如, 99.9% 或者 99.99 等等。

按照云计算层次的分类 [6], 我们认为云计算的 Availability( 简称  $Availability_{CS}$ ) 包括 IaaS 的 Availability( $Availability_{IaaS}$ ), PaaS 的 Availability( $Availability_{PaaS}$ ) 和 SaaS 的 Availability ( $Availability_{SaaS}$ )。

$$Availability_{CS} = \cap (Availability_{SaaS}, Availability_{PaaS}, Availability_{IaaS})$$

我们认为, 用户最终感知的云计算的可用性是与云计算 3 个层面的可用性紧密相关的。

在下面小节中, 我们首先来形式化定义一个云计算服务的可用性并做相应的算法讨论。然后, 对云计算分层模型中 IaaS, PaaS 和 SaaS 在可用性之间的关系做理论探讨。

### 2.1 可用性

假定在一个采样时间范围 (例如时间  $T$  小时内) 服务发生的不可用(Unavailable) 次数是  $N$ 。每次不可用之前正常运行的时间定义为 TBF(Time Before Failure)。每次用来恢复服务正常运行的时间定义为 TTR(Time To Repair)。



图 3 云计算服务的可用性

由图 3 可知，在采样时间 T 范围内，服务的可用性为：

$Availability_T = \sum_{i=1}^N TBF(i) / T$ ，其中， $\sum_{i=1}^N TBF(i)$  是在时间 T 内，云服务总共的可用时间。

因为  $T = [\sum_{i=1}^N TBF(i) + \sum_{i=1}^N TTR(i)]$ ,

所以， $Availability = \sum_{i=1}^N TBF(i) / [\sum_{i=1}^N TBF(i) + \sum_{i=1}^N TTR(i)]$   
 $= [\sum_{i=1}^N TBF(i) / N] / [(\sum_{i=1}^N TBF(i) + \sum_{i=1}^N TTR(i)) / N]$

因此，我们推导出在时间 T 小时里，云服务的可用性为：

**公式 1**  $Availability_T = MTBF_T / (MTBF_T + MTTR_T)$

其中：

$MTBF_T$ ：在时间 T 内，云服务的 Mean Time Before Failure[7].

$MTTR_T$ ：在时间 T 内，云服务的 Mean Time To Repair[8].

根据公式 1，我们可以定义一个云服务在基于采样周期 T 下，时间跨度为 K 下的 Mean Time Availability(MTA) 为：

**公式 2**  $MTA_K = \sum_{i=1}^M Availability(i) / M$ ，其中  $M=[K/T]$

假设一个云服务的 SLA 取样时间 T 是每天，或者说 24 个小时。如果考察 744 个小时的 MTA，根据上述公式，其 MTA 计算方法为：

由于  $[K/T] = 744/24 = 31$ ;

$MTA_{744 \text{ 小时}} = [(Availability(1) + Availability(2) + \dots + Availability(31)) / 31]$   
 $= [\sum_{i=1}^{31} Availability(i)] / 31$

**[推论 1] 云服务 Availability 的大小与 (MTTR/MTBF) 的比率成反比**

从  $Availability_T = MTBF_T / (MTBF_T + MTTR_T)$ ，我们很容易得到，  
 $Availability_T = 1 / (1 + MTTR_T / MTBF_T)$

定义 Mean Time Failure Ratio(MTFR) 代表 (MTTR/MTBF) 的比率。

$Availability_T = 1 / (1 + MTFR_T)$

显然，错误失败的比率越大，云服务的可用性就越小。其关系曲线可以简单表示如下：

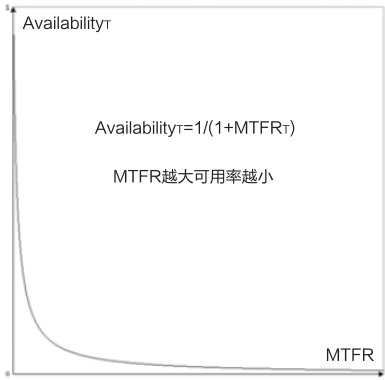


图 4 MTFR 与 Availability 的关系曲线

**[推论 2]** 如果云服务的上线正常运行时间和下线时间是一个线性关系，比率为  $a$ ，那么服务的可用性是一个常数  $[a / (1 + a)]$ ，不随着采样周期  $T$  变化。因此，SLA 可以不考虑星期，月或者年的影响。

假设  $MTBF = a \times MTTR + b$ ，其中  $a, b$  是常量， $a$  为云服务上线正常运行时间和下线时间的比率。

由公式 1 可知：

$$\begin{aligned} Availability_T &= (a \times MTTR_T + b) / (a \times MTTR_T + b + MTTR_T) \\ &= (a \times MTTR_T + b) / [(a + 1)MTTR_T + b] \end{aligned}$$

假设  $b$  足够小，上述推导可简化为：

$$\begin{aligned} Availability_T &= (a \times MTTR_T) / [(a + 1)MTTR_T] \\ &= [a / (a + 1)] \end{aligned}$$

由此可见，在这种情况下，云服务的可用性是一个常量，和采样的周期无关。

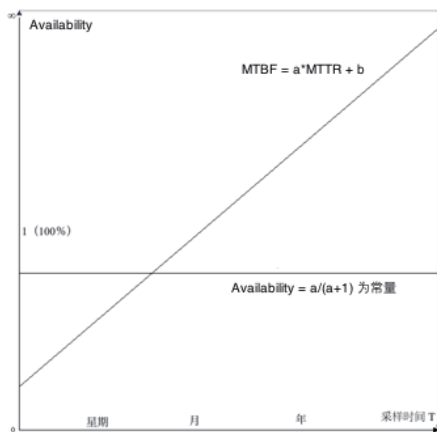


图 5 MTBF 和 MTTR 是线性关系时的 Availability

**[推论 3] 当一个云服务运行比较稳定的时候，云服务故障恢复的时间越短，云服务可用性越高。**

当一个上线的云服务逐渐成熟，能稳定的运行一个固定的时间，才出现异常。为简单起见，假设  $MTBF_T$  是一个常量  $\sigma$ ，意味着：

$$\begin{aligned} \text{Availability}_T &= \sigma / (\sigma + MTTR_T) \\ &= 1 / (1 + MTTR_T / \sigma) \end{aligned}$$

可以看出，当每次正常服务的时间是一个常量的时候，故障发生时修复的速度越快，时间越少， $\text{Availability}_T$  的值就越大。因此云服务可用性就越高。

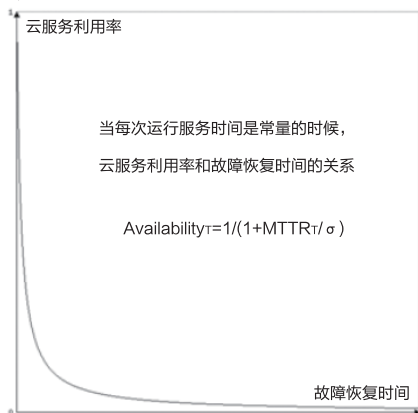


图 6 MTTR 与 Availability 的关系曲线



## 【案例讨论】

假设一个云服务提供商希望提供一可用性不低于某个范围值，例如 99.9% 或者 99.99% 等，从而获得商业上的竞争优势。

\* 如果 MTTR 是可控的，例如是可修护的 (Repairable) 部件，具有一个修复时间上限常量  $\Delta$ 。例如，云计算数据中心内的软件模块，操作系统或者数据库的补丁，安全漏洞等，上述的  $Availability_T = 1/(1 + MTTR_T/MTBF_T)$  可以简化为  $Availability_T = 1/(1 + \Delta/MTBF_T)$ 。那么系统的可用性就完全依赖于 MTBF，或者说，在时间 T 内服务上线的平均时间。此时作为云计算服务提供商可以通过拉大 T 的取样范围（例如月或者年）和 / 或提高云服务的稳定性，从而提供最大的 MTBF，以符合所期望的系统 Availability 参数。

\* 如果 MTTR 是不可控的，例如是必须更换的 (Replaced) 部件，如硬盘，服务器硬件或者电源失效等，这些意味着 MTTR 的时间分布不具备一个上限常量。这种情况下云服务提供商应该通过加大容灾处理，1+1 硬件容错等手段来确保 MTTR 的收敛，并在采样时间 T 方面采取保守策略，例如 (1) 对什么是不可用 (UnAvailability) 进行更严格的自定义，(2) 对可用性的等级采纳月，季度或者年为单位的承诺。

## 【案例分析】

[例 1] 一个云业务持续运行的 MTBF 是 10,000 小时，但需要平均 10 个小时才能恢复正常运行，那么系统的可用性是多少？

$$Availability = 10,000/(10,000+10)=99.9\%.$$

[例 2] 如果要确保一个新的云业务的可用性是 99.99%，而且从内部测试可知平均运行时间大概可以保证 10,000 个小时才会发生错误，那么 IT 运维部门必须保证在平均多长时间修复任何崩溃？

从  $Availability_T = 1/(1 + MTTR_T/MTBF_T)$ ，可以推导出

$$MTTR_T = MTBF_T * (1 - Availability_T) / Availability_T$$

因此， $MTTR_T = 10000 * (1 - 0.9999) / 0.9999 = 1$  小时。

IT 部门必须在 60 分钟之内修复系统恢复云服务的上线，否则就无法达到给租户承诺的 SLA。

[例 3] 假设一个云业务必须保证 99.999% 的可用性，如果从内部测试评估认为每次业务出错恢复的时间大概为 12 个小时左右。那么对业务质量控制应该是什么？必须保证多长时间业务正常运行？

从  $Availability_T = 1/(1 + MTTR_T/MTBF_T)$ ，可以推导出

$$MTBF_T = (MTTR_T * Availability_T) / (1 - Availability_T)$$

因此,  $MTBF_T = (12 * 0.99999) / (1 - 0.99999)$   
= 1,199,988 小时 = 49,999.5 天  
= 7,143 星期 = 1786 个月  
= 149 年 !!!!!

这个业务必须能保证连续 149 年的无故障运行, 才能达到设计目标! 换言之, 5 个 9 的设计目标是不现实的.

## 2.2 云计算分层结构可用性

本小节考察云计算分层结构中 IaaS, PaaS 和 SaaS 可用性的内在联系. 首先提出并定义云计算中可用单元 AU(Availability Unit), 可用集合 AS(Availability Set), 单独可用性 SA(Standalone Availability) 和部署可用性 DA(Deployment Availability) 的概念, 然后定性和定量分析各种相关模型的可用性.

### 2.2.1 定义

根据 Berkeley 关于云计算模型的定义, 一个云服务可以简单划分为 SaaS, PaaS 和 IaaS 三层结构. 在这 3 层结构中, 本文定义如下概念:

**可用单元 AU(Availability Unit):** 某一层提供给上一层服务的最小基本服务单元. 是云计算服务的最小粒度. 同一层中可用单元之间互相独立, 一个可用单元的失效不会影响其他单元. 常见的 AU 的例子为 AWS 的 Availability Zone[9] 和 Windows Azure 的 Region[10]. 一个 AU 是一个逻辑概念, 对于 IaaS 而言, 可以是一个数据中心; 对于 PaaS, 可以是一个 WEB, 数据库的实例; 对 SaaS, 可以是一个软件应用.

**可用集合 AS(Availability Set):** 同一层中一个或者多个 AU 的联合, 并作为逻辑上的一个整体提供服务给上一层. AS 具备一个仲裁 (Arbiter) 的功能模块. Arbiter 用来实时监测该 AS 中的 AU 可用性, 并根据相应的算法, 仲裁哪个 AU 可以提供最好的, 或者最适合的服务. 当一个 AS 具备多于一个的 AU 时, 该 AS 被认为具备非单点失效属性. Arbiter 本身可以是一个分布式的结构, 例如基于 LVS 集群的负载均衡 [11], 或者 AWS 的 LBS 服务 [12] 等. 本文假设 Arbiter 本身具备足够的弹性, 因此不讨论 Arbiter 的单点失效问题. 另外, 不失一般性, 本文假设 Arbiter 的算法基于提供最大可用性的算法, 即在一个可用集里, Arbiter 总是试图为服务申请者提供最大的可用性, 简称为 BA(Best Availability) 算法.

AU 和 AS 的逻辑拓扑关系如下图所示:

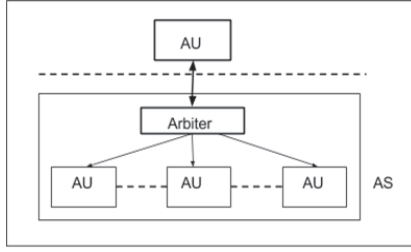


图 7 可用单元和可用单元集

对于一个 AU 和 AS, 定义其单独可用性和部署可用性如下:

**单独可用性 SA(Standalone Availability):** 假设在为其提供服务的下一层 AU 或者 AS 的可用性是 100% 的情况下, 一个 AU/AS 自身的可用性。例如, 一个 SaaS/PaaS 服务在没有迁移到公有云和上线之前, 在私有环境下或者实验室环境下测试时的可用性。

**部署可用性 DA(Deployment Availability):** 在为其提供服务的下一层 AS/AS 的可用性是动态变化的情况下, 一个 AU/AS 表现出来的实际可用性。例如, 一个 SaaS 或者 PaaS 服务被部署在一个第三方的 IaaS 环境下实际能达到的可用性; 一个 IaaS 被部署在一个第三方的物理数据中心, 在计算, 存储和网络环境变化下所能实际提供的可用性。

### 2.2.2 AU 和 AS 的 SA 可用性

假设一个 AS 里有  $m$  个 AU, 并形式表达为  $AS = \{AU_1, AU_2, \dots, AU_m\}$ , 每一个 AU 都是等同的服务模块 (软件或者硬件), 定义 AU 和 AS 的 SA 具有如下属性:

$$\forall AU_{ij} \quad (i, j = 1, 2, \dots, m),$$

$$\text{公式 3} \quad SA_{AU_i} = SA_{AU_j} = A$$

其中  $A$  是一个常量, 例如, 99.99%, 或者 99.99999%.

由于在任何一个 AS 里, 每个 AU 是等同的设计, 和基于同样的部署, 因此, 共同拥有一个相同的单独可用性。

$$\text{公式 4} \quad SA_{AS} = 1 - (1 - A)^m$$

一个 AS 的 Arbiter 的缺省算法是为服务申请提供最合适的可用性。当 AS 中的所有  $m$  个 AU 都同时失效无法工作的时候, 这个 AS 作为一个整体才失效 [13][14].

显然,  $m$  个 AU 同时失效的概率是  $(1-A)^m$ . 因此, AS 能提供的最大可用性是  $1-(1-A)^m$ .

下面我们讨论几个边界条件:

当  $A=0$  的时候, 即每个单独的 AU 都是失效的, 因此  $1-(1-A)^m=1-1=0$ . 因此作为一个整体 AS 自然也是无效的.

当  $A=1$  的时候, 即每个单独的 AU 都是 100% 的无故障概率, 因此  $1-(1-A)^m=1$ . 因此作为一个整体 AS 自然也是 100% 可靠的.

从  $1-(1-A)^m$ , 可以容易得出, 因为,  $0 \leq A \leq 1$ , 因此一个 AS 的单独可用性大于任何一个单独 AU 提供的可用性  $A$ . 或者说,  $1-(1-A)^m \geq A$ .

**定理: 一个 AS 的单独可用性大于任何一个其内部 AU 的单独可用性.**

**证明:**

我们通过简单的归纳法来证明.

当  $m=2$ ,  $1-(1-A)^m = 1-(1-A)^2$ . 显然  $(1-A)(1-A) \leq (1-A)$ . 因此,  $1-(1-A)^2 \geq 1-(1-A) = A$

从下图曲线, 我们也可用显然看出函数分布图.

假设  $1-(1-A)^m \geq A$ . 下面来考虑  $1-(1-A)^{(m+1)}$ .

$$1-(1-A)^{(m+1)}=1-(1-A)^m (1-A).$$

因为  $0 \leq A \leq 1$ , 因此  $0 \leq 1-A \leq 1$ ,  
因此,  $(1-A)^m (1-A) \leq (1-A)^m$

$$\text{因此, } 1-(1-A)^{(m+1)}= 1-(1-A)^m (1-A) \geq 1-(1-A)^m \geq A$$

因此, 我们得到一个 AS 的单独可用性大于任何一个其内部 AU 的单独可用性

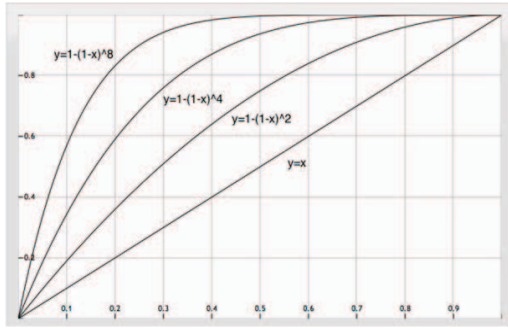


图 8 AS 与 AU 单独可用性分布

### 2.2.3 AU 和 AS 的 DA 可用性

我们认为，一个云计算服务的实际可用性依赖于其底层支撑服务的可用性。例如，在一个公有云的 IaaS 上，部署一个 PaaS 服务。该 PaaS 服务的实际可用性是随着 IaaS 服务的可用性变化的。如果 IaaS 能提供的可用性是 99.9%，尽管 PaaS 服务的设计的单独可用性是 99.9%，那么该 PaaS 对 SaaS 层的服务可用性是  $99.9\% \times 99.9\%$ ，即减少到 99.8%。极端情况是，如果该 IaaS 服务下线，无论该 PaaS 服务设计的单独可用性大小如何，其部署可用性都减少为 0。

一个 AU 和为其提供服务的 AS 的部署可用性关系可以形式表达为：

$$\text{公式 5 } DA_{AU} = SA_{AU} \cdot DA_{AS}$$

其中， $SA_{AU}$  为这个 AU 服务的单独可用性； $DA_{AS}$  为这个 AU 部署的可用集 AS 的部署可用性。

显然，当  $DA_{AS} = 100\%$  的时候， $DA_{AU} = SA_{AU}$ 。

在得到一个 AS 中的每个 AU 的 DA 可用性依赖于其下层服务可用性的量化关系之后，可以很容易地定义和推导出一个 AS 的 DA 可用性。

假设一个 AS 里有  $m$  个 AU，即  $AS = \{AU1, AU2, \dots, AU m\}$ 。根据公式 4 可知，每个 AU 的 DA 是依赖于底层 AS 的可用性的，因此 AS 中的  $m$  个 AU 可能具备不同的 DA 可用性。根据 Arbiter 的最大可用性算法，可以导出一个 AS 对外的 DA 可用性为：

**公式 6**  $DA_{AS} = 1 - (1 - DA_{AU1}) * (1 - DA_{AU2}) * \dots * (1 - DA_{AUm})$

假设  $DA_{AUi} = \text{Max}(DA_{AU1}, DA_{AU2}, \dots, DA_{AUi}, \dots, DA_{AUm})$ ,

我们可以容易证明得到,

$$DA_{AUi} \leq DA_{AS} \leq 1 - (1 - DA_{AUi})^m$$

其内涵是: 一个 AS 作为一个整体, 其动态部署可用性  $DA_{AS}$  大于其系统中其任何一个 AU 的部署可用性, 但这个 AS 的可用性具备一个上限, 是  $1 - (1 - DA_{AUi})^m$ , 其中  $AUi$  是该 AS 中可用性最大的 AU.

**证明:**

因为任何一个  $0 \leq 1 - DA_{AUj} \leq 1, (j = 1, \dots, m)$ ,

所以,

$$(1 - DA_{AU1}) * (1 - DA_{AU2}) * \dots * (1 - DA_{AUi}) * \dots * (1 - DA_{AUm}) \leq (1 - DA_{AUi})$$

因此,  $1 - (1 - DA_{AU1}) * (1 - DA_{AU2}) * \dots * (1 - DA_{AUi}) * \dots * (1 - DA_{AUm}) \geq 1 - (1 - DA_{AUi}) = DA_{AUi}$

即,  $DA_{AUi} \leq DA_{AS}$

同理,

$$\begin{aligned} & (1 - DA_{AU1}) * (1 - DA_{AU2}) * \dots * (1 - DA_{AUi}) * \dots * (1 - DA_{AUm}) \geq \\ & (1 - DA_{AUi}) * (1 - DA_{AU1}) * \dots * (1 - DA_{AUi}) * \dots * (1 - DA_{AUi}) = (1 - DA_{AUi})^m \end{aligned}$$

因此,  $1 - (1 - DA_{AU1}) * (1 - DA_{AU2}) * \dots * (1 - DA_{AUi}) * \dots * (1 - DA_{AUm}) \leq 1 - (1 - DA_{AUi})^m$

即,  $DA_{AS} \leq 1 - (1 - DA_{AUi})^m$

下面具体讨论云计算基于 SaaS, PaaS 和 IaaS 的 3 层模型并定义其可用性的关系.

不失一般性，假设一个 SaaS AS 部署在一个 PaaS AS 上；该 PaaS AS 部署在一个底层的 IaaS AS 上。其逻辑结构如图 9 所示。

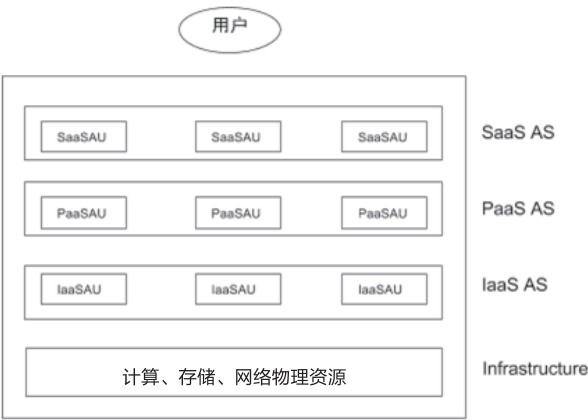


图 9 基于 AS 的云计算模型

对于 PaaS 而言，上图的 SaaS 层是一个复合的 Pseudo AU 或者是抽象的 SaaS AU. 因此，从公式 5, 容易得出，其  $DA_{SaaS}$  和下层的 PaaS AS 的关系是：

**公式 7**  $DA_{SaaS\ AS} = SA_{SaaS\ AS} \cdot DA_{PaaS\ AS}$

同理，PaaS AS 和 IaaS AS 的关系是：

**公式 8**  $DA_{PaaS\ AS} = SA_{PaaS\ AS} \cdot DA_{IaaS\ AS}$

其中：

- SaaS AS: SaaS AU 的单元集合；
- PaaS AS: SaaS AU 使用的 PaaS 的服务单元集合；
- IaaS AS: PaaS AU 使用的 IaaS 的服务单元集合。

显然，

- \* 一个 SaaS 服务的部署可用性一定小于等于其依赖的 PaaS 服务的部署可用性。
- \* 一个 PaaS 服务的部署可用性一定小于等于其依赖的 IaaS 服务的部署可用性。

性 .

**定理：**  $DA_{SaaS\ AS} \leq DA_{PaaS\ AS} \leq DA_{IaaS\ AS}$

**[ 讨论 ]:**

在一个云服务中，假设 SaaS 和 PaaS 的 SA 是 99.9%，IaaS 的 DA 是 99.9%。

PaaS 提供给 SaaS 的 DA 可用性为：99.9% \* 99.9% = 99.8%。

SaaS 最终能提供给用户的 DA 可用性只有 99.9%\*99.9%\*99.9% = 99.7%!

换言之，作为一个 PaaS 服务提供者，如果将服务部署在这样的一个 IaaS 环境中，该 PaaS 的实际部署可用性的最大值是 99.8%；作为一个 SaaS 服务提供者，如果将服务部署在这样的一个 IaaS 和 PaaS 环境中，该 SaaS 的实际部署可用性的最大值是 99.7%。

通过这种量化分析，可以很好地指导一个云计算服务提供商去定义其相应的 SLA 的可用性。

当一个 SaaS 或者 PaaS 服务要部署在第三方的 PaaS 或者 IaaS 服务上的时候，如何去推算服务提供商的服务质量 SLA？

下面讨论 SaaS 和 PaaS 的关系。PaaS 和 IaaS 的关系类同。

通过公式 4,6,7,8, 可以很容易的得出，

$$DA_{PaaS\ AS} = DA_{SaaS\ AS} / SA_{SaaS\ AS} \\ = [1 - (1 - DA_{SaaS\ AU\ 1}) * (1 - DA_{SaaS\ AU\ 2}) * \dots * (1 - DA_{SaaS\ AU\ m})] / [1 - (1 - A)^m]$$

其中 A 是 SaaS 里每个 AU 的相同的单独可用性。

不失一般性，假设各个  $DA_{SaaS\ AU}$  的值是一个常量  $A'$ ，因此，一个 PaaS AS 的可用性为：

**公式 9**  $DA_{PaaS\ AS} = [1 - (1 - A')^m] / [1 - (1 - A)^m]$

其中 A 是 SaaS 里每个 AU 的相同的单独可用性。

$A'$  是 SaaS 在测量中的样本的部署可用性。

**公式 10**  $DA_{IaaS\ AS} = [1 - (1 - A')^m] / [1 - (1 - A)^m]$



其中 A 是 PaaS 里每个 AU 的相同的单独可用性。  
A' 是 PaaS 在测量中的样本的部署可用性。

基于公式 9 和 10, 就可以在大规模部署一个 SaaS 或者 PaaS 在第三方的 PaaS 或者 IaaS 服务上之前, 通过小规模抽样测试, 并根据结果来预测该服务提供商的服务可用性。从而做出相对正确的评估和商务决定。

## 2.3 云计算分层结构拓扑

该小节讨论基于 AS 的分层部署拓扑, 介绍扁平 AS(Flat AS), 复合 AS(Composite AS) 的两种通用拓扑; 然后定义两种结构下云计算中服务收敛比(Over Subscription)及其计算方法; 最后定性和定量分析两种拓扑下的可用性及优缺点。

### 2.3.1 定义

**扁平 AS(Flat AS):** 一个 AS 为一个扁平 AS, 如果使用该 AS 服务的上层 AU 是部署在该 AS 内部的任何一个 AU 上。从图论的角度看来, 是一个 Mesh 的全连通的 M:N (M 为申请部署服务的 AU; N 为提供服务 AS 中 AU 的总数) 的映射。

在图 10 所示的一个扁平 AS 的拓扑结构中, 上一层的 2 个 AU(例如一个 PaaS 服务)通过全连接的方式部署在 4 个下层的 AU 上(例如一个 IaaS 服务)。根据定义, 这个 4 个 IaaS AU 形成对于 PaaS 层一个扁平 AS。

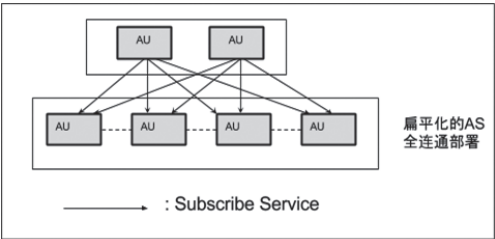


图 10 扁平 AS 拓扑

**复合 AS(Composite AS):** 一个 AS 为一个复合 AS, 如果该 AS 由多个扁平 AS 构成。

图 11 所示为一个复合 AS 的拓扑结构图。上一层的两个 AU(例如一个 PaaS 服

务)通过全连接的方式只部署在 2 个下层的 AU 上(例如一个 IaaS 服务), 根据定义, 这个 4 个 IaaS AU 分裂并形成对于这两个 PaaS AU 的 2 个扁平 AS(每个 AS 含有 2 个 AU). 这两个小的扁平 AS 构成一个复合 AS 来提供服务。

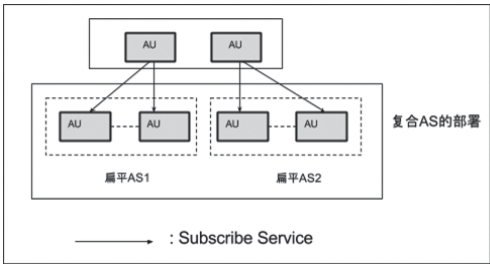


图 11 复合 AS 拓扑

扁平 AS 的一个特例是当其内部的扁平 AS 都只含有一个 AU 的时候, 上层 AU 和提高服务的 AU 构成了一个 M:1 的映射, 如图 12 所示:

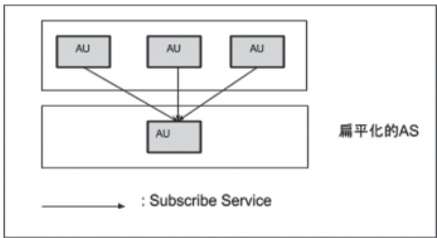


图 12 扁平 AS 拓扑特例 -M:1 映射

复合 AS 的一个特例是当其内部的扁平 AS 都只含有一个 AU 的时候, 上层 AU 和提高服务的 AU 构成了一个 1:1 的映射, 如图 13 所示:

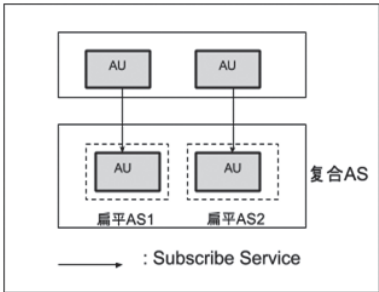


图 13 复合 AS 拓扑特例 -1:1 映射

图 12 和图 13 的拓扑都具备单点失效的性质。一旦底层服务 AU 失效，依赖于其服务的上层 AU 也完全失效。

我们认为，任何一个云计算的 3 层模型都可以通过扁平 AS 或者复合 AS 的结合，以递归的方式构建。

另外，从图 10 到图 13 的各种扁平 AS 和复合 AS 的拓扑，可以很容易看到上层 AU 对 AS 的服务申请的数量是不同的。这种区别对一个云服务对 AS 的可靠性和可用性会产生影响。为了形式化刻画这种服务申请的区别，定义云计算 AU 和 AS 的收敛状态和收敛比 (Subscription Ratio) 如下：

#### **AU 收敛比：**

任何一个 AU 的设计都具备一个能支持多少个服务部署的上限值，定义该值为 AU 容量 (AU Capacity), 记作 C。

一个 AU 服务当前已经支持的服务的数量，定义为订阅量 (Subscription Amount), 记作 S。

$S \leq C$  时，称为低收敛状态 (Under Subscription)

$S > C$  时，称为高收敛状态 (Over Subscription)

定义云计算 AU 收敛比为：

**公式 11**  $AU_{sub} = S / C$

显然，一个 AU 的  $(S / C)$  值越大，它所需要承载的服务就越多，耗费的资源和设计复杂性就越大。

假设一个扁平 AS 有 N 个 AU, 定义云计算 AS 的收敛比为：

**公式 12**  $AS_{sub} = (\sum_{i=1}^N AU_{i sub}) / N$

类似地，一个复合 AS 的收敛比可以定义为：

**公式 13**  $AS_{sub} = (\sum_{i=1}^M AS_{i sub}) / M$ ，其中 M 为一个复合 AS 中的扁平 AS 的数目。

### **2.3.2 扁平和复合 AS 的可用性**

下面讨论扁平 AS 和复合 AS 拓扑下的 SA 和 DA 可用性。

不失一般性，考虑图 10 和图 11 的拓扑。假设需要服务的 AU 位于一个 PaaS 层，分别为 PaaS AU1, PaaS AU2，并具有相同的  $SA_{PaaS A'}$ ；提供服务的 IaaS 层分别为 IaaS AU1, IaaS AU2, IaaS AU3 和 IaaS AU4，并具有相同的  $SA_{IaaS}$ 。

对于扁平 AS 拓扑，因为其作为一个单一的整体向上提供服务，根据公式 6:

$$DA_{IaaS AS} = 1 - (1 - DA_{IaaS AU 1}) * (1 - DA_{IaaS AU 2}) * (1 - DA_{IaaS AU 3}) * (1 - DA_{IaaS AU 4})$$

$$\text{假设 } DA_{IaaS AU 1} = DA_{IaaS AU 2} = DA_{IaaS AU 3} = DA_{IaaS AU 4} = A'$$

$$\text{因此, } DA_{IaaS AS} = [1 - (1 - A')^4]$$

因此，这个 PaaS 作为一个 AS 的可用性 DA 为：

#### 扁平拓扑可用性：

$$DA_{PaaS AS} = SA_{PaaS} \cdot [1 - (1 - A')^4]$$

由公式 4, 可推出，

$$DA_{PaaS AS} = [1 - (1 - A)^2] \cdot [1 - (1 - A')^4]$$

对于复合 AS 拓扑，由于 PaaS 是分别部署在两个小的扁平 AS1 和 AS2 中，

#### 复合拓扑可用性：

$$\text{由公式 4, } DA_{PaaS AS} = 1 - (1 - B)^2$$

B 定义为是 PaaS 的一个 AU 和支撑其服务的两个 IaaS AU 的串行模式下的可用性：

$$B = A * [1 - (1 - A')^2]$$

因此，

$$DA_{PaaS AS} = 1 - (1 - A * [1 - (1 - A')^2])^2$$

**推论 4: 在最大可用性算法下，一个扁平 AS 和一个复合 AS 提供的 DA 可用性近似等价。**

下图所示为假设  $A = 0.99$  的情况下，扁平 AS 可用性函数为：

$$y = (1 - (1 - 0.99)^2) * (1 - (1 - x)^4)$$

和

复合 AS 可用性函数为：

$$y = 1 - (1 - 0.99 * (1 - (1 - x)^2))^2$$

在区间  $(0, 10)$  的分布。

从图 14 可以观察到，其分布基本上一致。

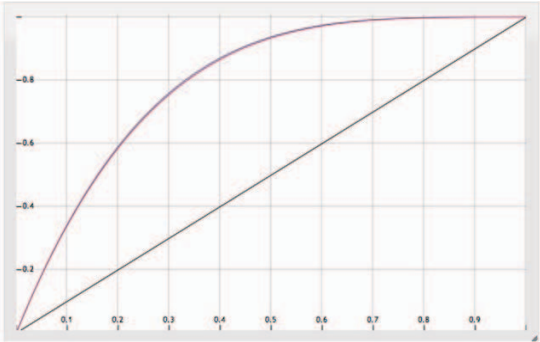


图 14-1 扁平 AS 和复合 AS 的 DA 可用性比较 [ 区间 0 -1]

下面是两张张区间放大图，可用看到蓝色的表示扁平 AS 的可用性分布；红色的是复合 AS 的可用性分布。基本上相等，没有显著区别。

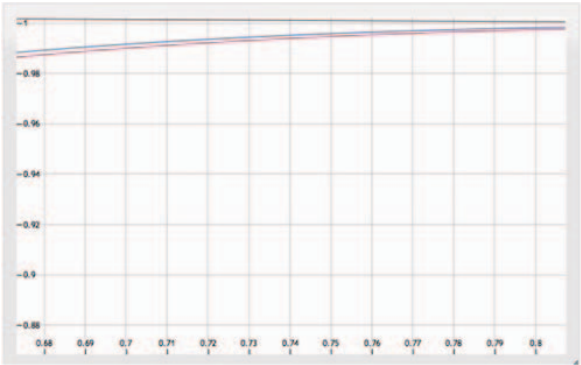


图 14-2 扁平 AS 和复合 AS 的 DA 可用性比较 [ 区间 0.5 -0.8]

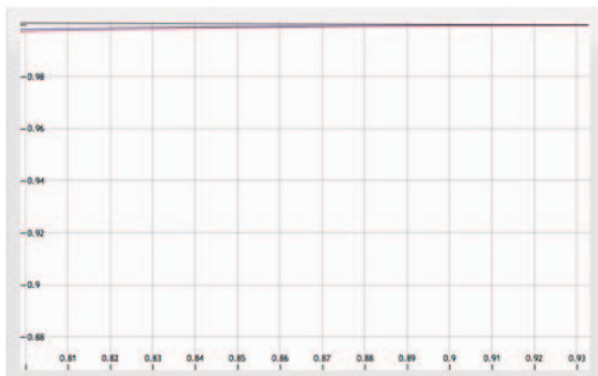


图 14-3 扁平和复合 AS 的 DA 可用性比较 [ 区间 0.8 -0.99]

既然扁平 AS 和复合 AS 都能提供类似的最大可用性，其区别是什么呢？在实际部署中的选择折衷是什么？

**推论 5: 扁平 AS 提供给上层申请服务的 AU 的可用性是均匀和一致化的；复合 AS 提供给上层申请服务的 AU 的可用性分布可以是不均匀的。**

### 【讨论】

在扁平 AS 拓扑下，各个需要服务的 AU 的可用性的分布是均匀的。一个均匀化的服务层（例如 PaaS）可以为它的客户提供更好的负载均衡服务（例如 SaaS）。PaaS AS 的 Arbiter 的算法可以是单纯的 RR(Round Robin) 模式，极大的降低了 PaaS 服务的复杂性。

在复合 AS 拓扑下，部署在不同的扁平 IaaS AS 下的上层 PaaS AU 的最大可用性可能是不同的。PaaS AS 必须提供足够强健的 Arbiter 服务，例如 BA 算法和相应的容错机制。其缺点除了 Arbiter 的复杂性外，还会导致内部 AU 的负载出现压力过载的危险，对 PaaS 设计的鲁棒性带来很大的挑战。

如图 15 所示，在一个不均匀的可用性的环境下，可能会导致一个上层的 AU( 红色标注的 ) 负载压力过大，使得上层服务 AS 出现单点失效的风险加大。特别是当其他 AU 节点的可用性都小于承诺的 SLA 的可用性时，所有的服务压力都会被迫牵引到红色的 AU 中。

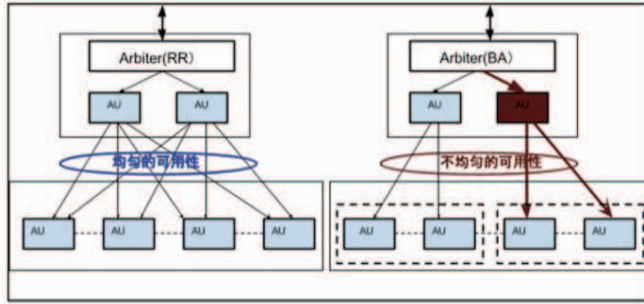


图 15 扁平 AS vs 复合 AS

下面讨论扁平 AS 的缺点 . 在这之前 , 首先定义云计算分层模型的收敛性问题 .

**推论 6: 在支持同等的上层 AU 数量的情况下 , 一个扁平 AS 具备更高的收敛比 (Over-Subscription Ratio) . 一个数据中心需要更多的资源 .**

**[ 证明 ]**

假设一个扁平 AS 拥有  $N$  个 AU, 并为上层  $K$  个 AU 服务 .

由公式 9 和 10, 可以得出  $AS_{\text{扁平 sub}} = [\sum_{i=1}^N (S/C)]/N$

根据扁平 AS 的全连通定义, 该扁平 AS 的每个 AU 的  $S$  等于  $K$ , 所以, 该扁平 AS 的收敛比是:

$$AS_{\text{扁平 sub}} = [\sum_{i=1}^N (K/C)]/N = K/C$$

可见, 一个扁平 AS 的收敛比与其 AU 数目无关, 只依赖于上层服务申请的数量和每个 AU 的容量  $C$ .

考虑一个同样拥有  $N$  个 AU 的复合 AS. 不失一般性, 假设该 AS 由  $M$  个小的扁平 AS 组成; 在每个扁平  $AS_i$  上请求服务的上层 AU 数目相应为  $k_i$ , 并且满足  $\sum_{i=1}^M k_i = K$ .

由公式 10 和 11, 可以得出该复合 AS 的收敛比为:

$$AS_{\text{复合 sub}} = (\sum_{i=1}^M AS_i)/M = [\sum_{i=1}^M (k_i/C)]/M \\ = K / (C \times M)$$

因为  $M \geq 1$ , 而且  $AS_{\text{扁平 sub}} = K/C$ , 所以,  $AS_{\text{复合 sub}} \leq AS_{\text{扁平 sub}}$  可以看到, 扁平 AS 的收敛比大于等于复合 AS 的收敛比 .

在扁平 AS 的拓扑图 9 中, 其 AU 和 AS 的收敛比为:

$$AU_{\text{sub}} = 2/C;$$

$$AS_{sub} = (\sum_{i=1}^4 AU_{i_{sub}})/4 = 2/C$$

而在复合 AS 的拓扑图 10 中，其 AU 和 AS 的收敛比为：

$$AU_{sub} = 1/C ;$$

$$AS_{AS1_{sub}} = (\sum_{i=1}^2 AU_{i_{sub}})/2 = 1/C ;$$

$$AS_{AS2_{sub}} = (\sum_{i=3}^4 AU_{i_{sub}})/2 = 1/C$$

因此，复合 AS 的收敛比为：

$$AS_{sub} = (\sum_{i=1}^2 AS_i)/2 = (1/C + 1/C) / 2 = 1/C$$

显然，扁平 AS 的收敛比是复合 AS 的两倍。

### 【讨论】

从对上述拓扑的分析可以看到，利用扁平化的服务部署可以使得每个上层 AU 都获得均匀的和最高的可用性，但同时带来的代价是高收敛性，这意味着对底层服务的容量和高可靠性的需求。

另外，由于云计算服务通常是按照服务实例和时间来付费，因此通过扁平化的服务部署在经济上需要更高的成本。



### 3 云计算参考设计模型

本小节依据基于 AU 和 AS 的讨论提出若干个参考设计模型 (Reference Design), 并分别分析相应的可用性和收敛比等。

假设一个公有云系统, 提供了 IaaS 服务; 一个第三方厂家在该 IaaS 基础上部署了 PaaS 服务。然后 SaaS 服务提供商在该 PaaS 上进行部署。

本小节中, 每一个 SaaS, PaaS 或者 IaaS 模块缺省定义为一个 AU 单元。

#### 3.1 一字型架构 (Stick Architecture)

一字型架构是最简单的一种云服务的部署模型。如图 16 所示, 一个 SaaS 服务直接部署在一个 PaaS 服务上; 该 PaaS 服务部署在一个底层的 IaaS 上。

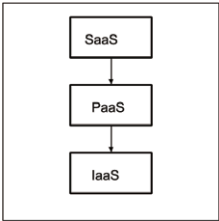


图 16-1 参考设计模型一：一字型架构 立体图

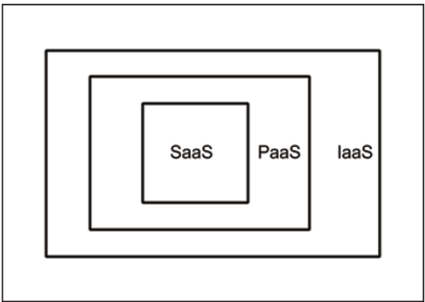


图 16-2 参考设计模型一：一字型架构 平面图

一字形结构的优点是简单并且成本低廉。其缺点很明显，存在着多个单点失效。例如 SaaS 和 PaaS 之间，PaaS 和 IaaS 都是单点失效的地方。系统不存在容错。

对于 IaaS 来说，其最大部署可用性为： $DA_{IaaS} = SA_{IaaS}$   
 对于 PaaS 来说，其最大部署可用性为： $DA_{PaaS} = SA_{PaaS} * DA_{IaaS}$   
 对于 SaaS 来说，其最大部署可用性为： $DA_{SaaS} = SA_{SaaS} * SA_{PaaS} * DA_{IaaS}$   
 $= SA_{SaaS} * SA_{PaaS} * SA_{IaaS}$

### 3.2 菱形架构 (Diamond Architecture)

菱形架构是一字型架构的变种，扩充了 PaaS 层的布署，从一字型结构中 PaaS 为一个 AU 演变为了含有两个 AU 的扁平 AS。如图 17 所示，一个 SaaS 服务直接部署在 2 个 PaaS 服务上；该 2 个 PaaS 服务部署在一个底层的 IaaS 上。

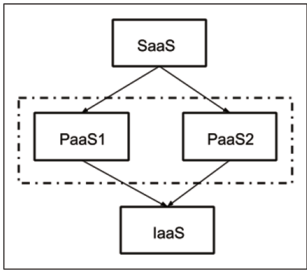


图 17-1 参考设计模型二：菱形架构 立体图

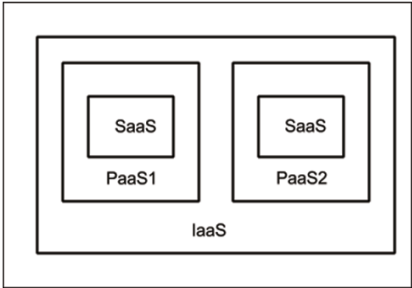


图 17-2 参考设计模型二：菱形架构 平面图

菱形架构的优点 PaaS 层的可靠性得到提高，和一字型架构相比，去除了 PaaS 层的单点失效性。缺点是 IaaS 层仍然是单点失效点。另外，由于增加了 PaaS 层的

部署，需要比一字型架构的成本要更高。

对于 IaaS 来说，其最大部署可用性为： $DA_{IaaS} = SA_{IaaS}$

对于 PaaS 来说，其最大部署可用性为：

由于  $DA_{PaaS1} = DA_{PaaS2} = SA_{PaaS} * DA_{IaaS}$ ，

因此，作为一个整体的 PaaS 层面的可用性  $DA_{PaaS} = 1 - (1 - DA_{PaaS1}) * (1 - DA_{PaaS2}) = 1 - (1 - SA_{PaaS} * SA_{IaaS})^2$ 。

对于 SaaS 来说，其最大部署可用性为： $DA_{SaaS} = SA_{SaaS} * DA_{PaaS} = SA_{SaaS} * [1 - (1 - SA_{PaaS} * SA_{IaaS})^2] = SA_{SaaS} * (SA_{PaaS} * SA_{IaaS})^2 + 2SA_{SaaS} * SA_{PaaS} * SA_{IaaS}$

### 3.3 人字型架构 (Bone Architecture)

人字形架构也是一字型架构的变种。其主要目的是提高 IaaS 层的可靠性。如图 18 和图 19 所示，一个 SaaS 服务直接部署在一个 PaaS 服务上；该 PaaS 服务部署在一个 2 个底层的 IaaS 上。

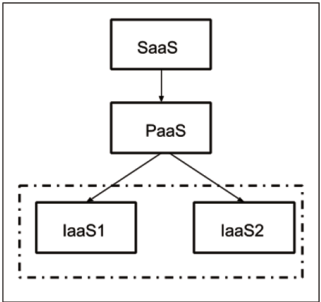


图 18-1 参考设计模型三：人形架构 立体图

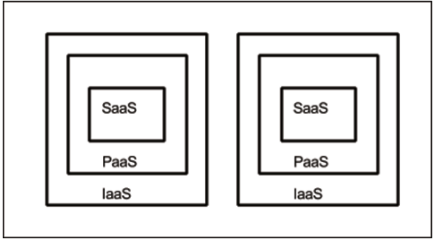


图 18-2 参考设计模型三：人形架构 平面图

人字形架构的优点 IaaS 层的可靠性得到提高，和一字型和菱形架构相比，去除了 IaaS 层的单点失效性。缺点是 PaaS 层仍然是单点失效点。另外，由于增加了 IaaS 层的布署，需要比一字型和菱形架构的成本要更高。

对于 IaaS 来说，其最大部署可用性为： $DA_{IaaS} = 1 - (1 - DA_{IaaS1}) * (1 - DA_{IaaS2}) = 1 - (1 - SA_{IaaS})^2$ 。

对于 PaaS 来说，其最大部署可用性为： $DA_{PaaS} = SA_{PaaS} * DA_{IaaS} = SA_{PaaS} * [1 - (1 - SA_{IaaS})^2]$ 。

对于 SaaS 来说，其最大部署可用性为： $DA_{SaaS} = SA_{SaaS} * DA_{PaaS} = SA_{SaaS} * SA_{PaaS} * [1 - (1 - SA_{IaaS})^2] = SA_{SaaS} * SA_{PaaS} * SA_{IaaS}^2 + 2SA_{SaaS} * SA_{PaaS} * SA_{IaaS}$

### 3.4 三角形架构 (Tri-Angle Architecture)

三角形架构是菱形和人字型架构的混合模式。其主要目的是提高 PaaS 和 IaaS 层的可靠性。如图 19 所示，一个 SaaS 服务直接部署在 2 个 PaaS 服务上；2 个 PaaS 服务部署在一个 4 个底层的 IaaS 上。但这 4 个 IaaS AU 是通过两个独立的 AS 来向上提供服务的。

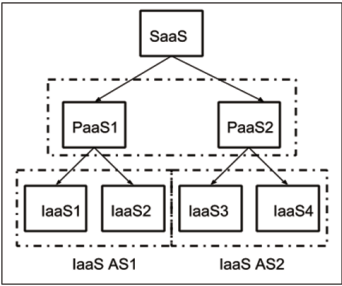


图 19-1 参考设计模型四：三角形架构 立体图

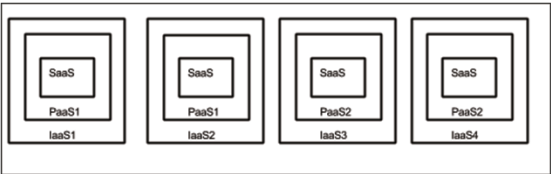


图 19-2 参考设计模型四：三角形架构 平面图

三角形架构的优点 IaaS 层的可靠性得到提高，和一字型，菱形和人字型架构相比，去除了 IaaS 层的单点失效性。缺点是 PaaS 层仍然是单点失效点。另外，由于增加了 IaaS 层的布署，需要比一字型和菱形架构的成本要更高。

对于 IaaS 来说，其最大部署可用性为略为复杂。其 4 个 IaaS AU 分为独立的 2 组，每两个 AU 组成了一个 IaaS AS。其中： $DA_{IaaS AS1} = 1 - (1 - DA_{IaaS1}) * (1 - DA_{IaaS2}) = 1 - (1 - SA_{IaaS})^2$ 。 $DA_{IaaS AS2} = 1 - (1 - DA_{IaaS3}) * (1 - DA_{IaaS4}) = 1 - (1 - SA_{IaaS})^2$ 。但需要注意的是，这两个 IaaS AS 是彼此独立的，因此不存在一个此整个 IaaS 的可用性  $DA_{IaaS}$

对于 PaaS 来说，其最大部署可用性为：由于  $DA_{PaaS1} = SA_{PaaS} * DA_{IaaS1} = SA_{PaaS} * [1 - (1 - SA_{IaaS})^2]$ 。 $DA_{PaaS2} = SA_{PaaS} * DA_{IaaS2} = SA_{PaaS} * [1 - (1 - SA_{IaaS})^2]$ ，因此整体  $DA_{PaaS} = 1 - (1 - DA_{PaaS1}) * (1 - DA_{PaaS2}) = 1 - [(1 - SA_{PaaS} * [1 - (1 - SA_{IaaS})^2])^2]$

对于 SaaS 来说，其最大部署可用性为： $DA_{SaaS} = SA_{SaaS} * DA_{PaaS} = SA_{SaaS} * [1 - (1 - SA_{PaaS} * [1 - (1 - SA_{IaaS})^2])^2]$

### 3.5 胖树架构 (Fat-Tree Architecture)

胖树架构是三角形架构的扩展。其主要目的每一个层面都是全映射的，从而最大程度的提高 PaaS 和 IaaS 层的可靠性。如图 20 所示，一个 SaaS 服务直接部署在 2 个 PaaS 服务上；2 个 PaaS 服务部署在一个 4 个平坦的 IaaS 上。

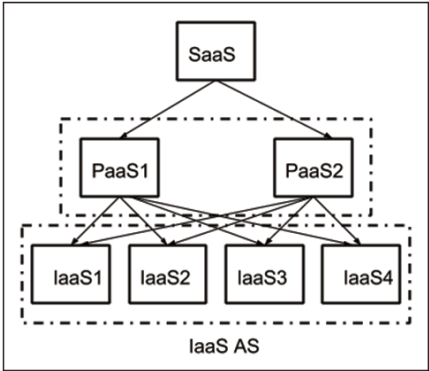


图 20-1 参考设计模型四：胖树架构 立体图

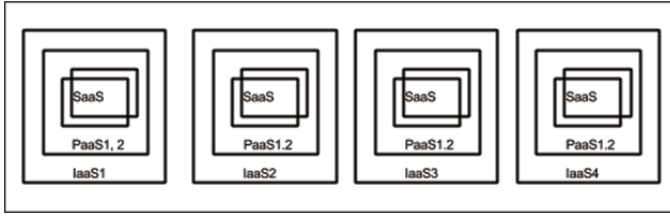


图 20-2 参考设计模型五：胖树架构 平面图

胖树架构的优点 PaaS, IaaS 层的可靠性得到最大的提高，缺点是成本最大，对调度仲裁机制，数据同步和容错机制的要求很高。特别是在跨物理数据中心的时候，系统会变的很复杂。

数据同步和容错机制的要求很高。特别是在跨物理数据中心的时候，系统会变的很复杂。

对于 IaaS 来说，其最大部署可用性略为复杂。其 4 个 IaaS AU 分为独立的 2 组，每两个 AU 组成了一个 IaaS AS。其中： $DA_{IaaS} = 1 - (1 - DA_{IaaS1}) * (1 - DA_{IaaS2}) * (1 - DA_{IaaS3}) * (1 - DA_{IaaS4}) = 1 - (1 - SA_{IaaS})^4$ 。提供了最大可能的可用性。例如，如果  $SA_{IaaS} = 99.9$ ,  $DA_{IaaS} = 99.9999$ 。

对于 PaaS 来说，其最大部署可用性为：由于  $DA_{PaaS1} = SA_{PaaS} * DA_{IaaS} = SA_{PaaS} * [1 - (1 - SA_{IaaS})^4]$ ， $DA_{PaaS2} = SA_{PaaS} * DA_{IaaS} = SA_{PaaS} * [1 - (1 - SA_{IaaS})^4]$ ，因此整体  $DA_{PaaS} = 1 - (1 - DA_{PaaS1}) * (1 - DA_{PaaS2}) = 1 - [(1 - SA_{PaaS} * [1 - (1 - SA_{IaaS})^4])^2]$

对于 SaaS 来说，其最大部署可用性为： $DA_{SaaS} = SA_{SaaS} * DA_{PaaS} = SA_{SaaS} * [1 - [(1 - SA_{PaaS} * [1 - (1 - SA_{IaaS})^4])^2]]$

和其他结构相比，显然胖树结构提供的各级可用性是最大的。但带来的系统复杂性，费用也是最高的。

## 4 案例研究 —— 亚马逊 AWS

### 4.1 Amazon AWS 系统结构

Amazon AWS 上线于 2006 年, 属于云计算服务中的 IaaS (Infrastructure as a Service ). 亚马逊 AWS 提供的数据中心群分布在全球各个区域 (Region). 分为 4 大区域. 其中包括: 北美区域 (North America Region), 亚太区域 (Asia and Pacific Region), 欧洲 (含中东和非洲) (Europe / Middle East / Africa) 和南美区域 (South America Region)[15], 如图 21 所示。



图 21 亚马逊数据中心群分布图

下面是亚马逊全球 4 大区域的具体分布：

#### 北美大区域：

分为 5 个区域：

US East (Northern Virginia) Region EC2 Availability Zones: 5 2006 年开始服务

US West (Northern California) Region EC2 Availability Zones: 3 2009 年开始服务

US West (Oregon) Region EC2 Availability Zones: 3 2011 年开始服务

AWS GovCloud (US) Region EC2 Availability Zones: 2    2011 年开始服务  
(注：GovCloud 是亚马逊为美国政府机构单独承运的一个云计算区域，位于美国的西北地区)

**亚太区域**

Asia Pacific (Singapore) Region EC2 Availability Zones: 2    2010 年开始服务

Asia Pacific (Tokyo) Region EC2 Availability Zones: 3    2011 年开始服务  
Asia Pacific (Sydney) Region EC2 Availability Zones: 2    2012 年开始服务

**欧洲区域**

EU (Ireland) Region EC2 Availability Zones: 3    2007 年开始服务

**南美区域**

São Paulo Region EC2 Availability Zones: 2    2011 年开始服务

另外，亚马逊为了提高业务的可靠性、稳定性和用户性能体验，也在全球各地建立了 CDN 的边缘加速节点数据中心 (Edge Location)。其分布区域如图 22 所示，主要是围绕着上述分布在全球的区域和可用区节点来建设的，从而可以加快用户存取数据的体验。

AmazonCloudFront and Amazon Route 53 services are offered at AWS Edge locations:			
North America	South America	Europe / Middle East / Africa	Asia Pacific
Ashburn, VA (2)	São Paulo, Brazil	Amsterdam, The Netherlands (2)	Hong Kong, China (2)
Dallas/Fort Worth, TX (2)		Dublin, Ireland	Osaka, Japan
Hayward, CA		Frankfurt, Germany (2)	Singapore (2)
Jacksonville, FL		London, England (2)	Sydney, Australia
Los Angeles, CA (2)		Madrid, Spain	Tokyo, Japan (2)
Miami, FL		Milan, Italy	
New York, NY (3)		Paris, France (2)	
Newark, NJ		Stockholm, Sweden	
Palo Alto, CA			
San Jose, CA			
Seattle, WA			
South Bend, IN			
St. Louis, MO			
*Denotes Amazon CloudFront only at this edge location			

图 22 亚马逊 Edge Location 分布图



为了最大的提高IaaS服务的可靠性，每个区域含有多个EC2可用区(Availability Zone). 可用区对应的就是专属于一个区域 (Region) 的分散的物理数据中心。其中，

- \* 一个区域可以含有多个可用区；
- \* 一个可用区只能属于一个区域。

AWS 目前区域和可用区的情况大概如下 [16]:

- 区域 :US – West (California + Oregon): 6 个可用区
- 区域 :US – EAST( North Virginia): 5 个可用区
- 区域 :South America (Sao Paulo): 2 个可用区
- 区域 :Europe – West (Dublin):3 个可用区
- 区域 :Asia Pacific (Singapore): 2 个可用区
- 区域 :Asia Pacific (Japan): 2 个可用区
- 区域 :Australia (Sydney): 2 个可用区
- 区域 : Gov. Cloud( 联邦政府云 )

在关于 AWS 的区域和可用区的关系方面，图 23 所示为一个比较清晰的拓扑关系 [17].

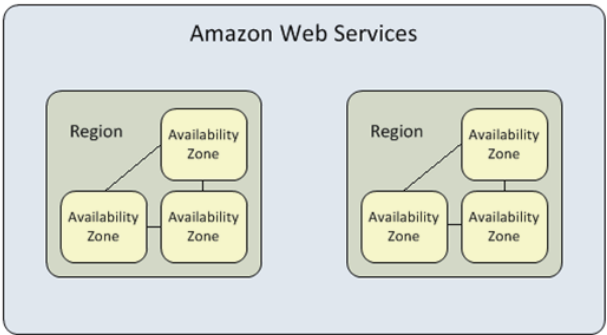


图 23 AWS 区域和可用区

不同的区域是完全独立的，不存在信息交互。从而去除一个区域失效故障可能带来对另外一个区域的影响。

同一个区域的可用区是分开的数据中心。一个可用区的失效不会影响到同一个区域里面的其他可用区。在同一个区域内，通过高速，低延时的网络将多个可用区互联。

用户可以根据自己业务的需求选择里客户比较接近的区域，例如美国的东部还是西部。

用户也可以通过在同一个区域内的不同的可用区里启动多个业务，从而避免可用区的单点失效

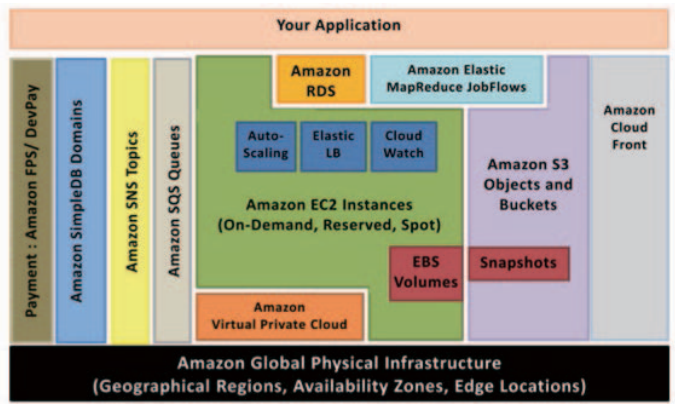


图 24 AWS 总体结构图

在一个可用区里，一个 AWS 的服务部件和服务模块实例 (Instance) 构成 [18]。如上图 24 所示，AWS 主要由 EC2(Elastic Compute Cloud),EBS(Elastic Block Store) 和 S3(Simple Storage Service) 等部件构成。其中 EC2 提供虚拟化的服务器主机资源；EBS 为 EC2 提供块设备的支持，类似一个大硬盘。然后通过格式化从而可以为 EC2 提供各种文件系统支持；S3 提供的是一个通过 WEB 可以访问的数据存储和备份资源，用户可以定期为其在 AWS EC2 产生的或者 EBS 里面的数据做实时或定期的备份，并可以通过 API 和各种 WEB 界面访问。

ELB(Elastic Load Balancer) 是可选的负载均衡的部件。通过 ELB，一个 AWS 的租户可以在 AWS 的一个可用区或多个可用区里部署多个 EC2 的实例。然后通过 ELB 实现负载均衡和容错的目的；RDS(Relational Database Service) 是一个基于 WEB 界面的关系数据库服务，AWS 的租户可以通过 RDS 的管理界面生成，管理自己的数据库业务。数据库可以是 MYSQL,ORACLE 或者微软的 SQL Server 等。CloudFront 是 AWS 的一个基于 WEB 的 CDN 加速服务，租户可以非常简单

的管理内容并决定对内容进行各种加速选择，也可以是和 AWS 的其他服务模块配合，例如，S3 等做数据存取加速。AWS 的 VPC(Virtual Private Cloud) 服务是一个提供给租户在 AWS 的云中建立自己的私有的网络的一种方案。类似于 Hosted Private Clouds 的范畴。一个租户可以在 AWS 里面创建一个 VPC, 然后自己完全控制在这个 VPC 域里网络的二层，三层配置，和各种安全掌控。

大体而言，一个最基本的 AWS 云服务由 EC2, EBS, RDS 和 S3 构成。其他的 服务是可选项，例如，用于负载均衡和容错的 ELB 等。

## 4.2 AWS 部署模式

AWS 的部署模式有多种，从最简单的 EC2+EBS+S3, 复杂的 ELB+EC2+ EBS+S3 的高可靠性容错模式和跨多个可用区的容错模式 [19]. 下面是几个相应布 网的示意图。

**最基本的 AWS 部署方案：**

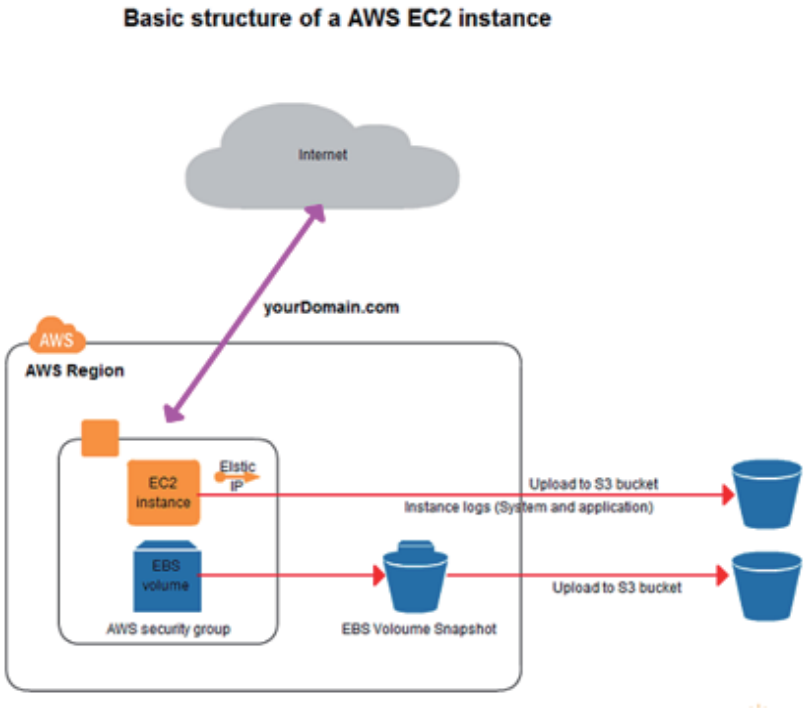


图 25 基本 AWS 部署方案

注：EC2 提供虚拟服务器；EBS 提供块设备存储；S3 提供基于 WEB 的数据访问和备份。

**基于 ELB 的高可靠性部署方案：**

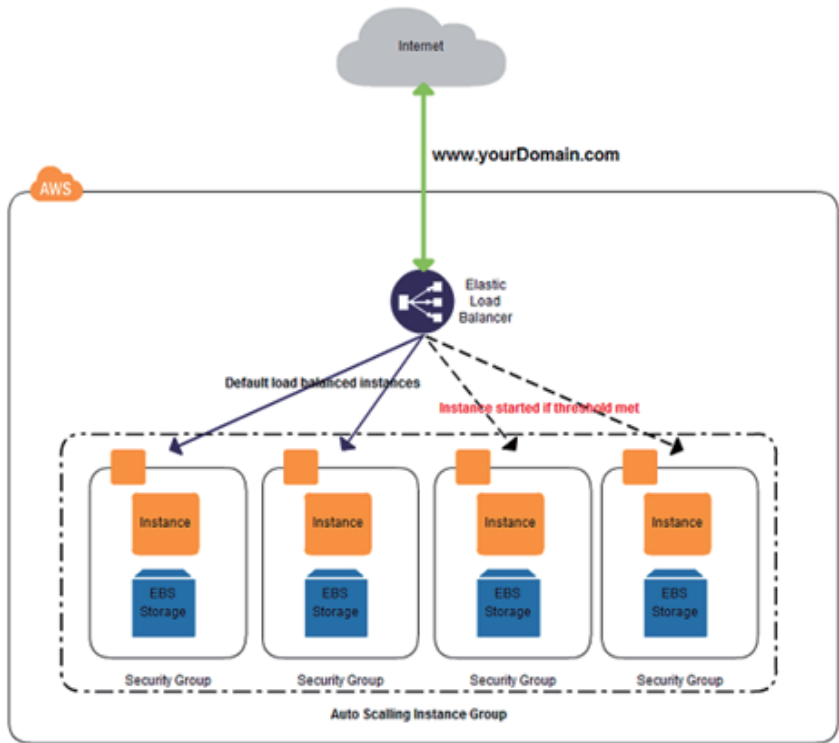


图 26 基于 ELB 的高可靠性部署方案

注：EC2 提供虚拟服务器；EBS 提供块设备存储；S3 提供基于 WEB 的数据访问和备份。ELB 提供负载均衡和容错，自动的把流量分布在不同的 EC2 实例上。

## 跨可用区的高可靠性部署方案：

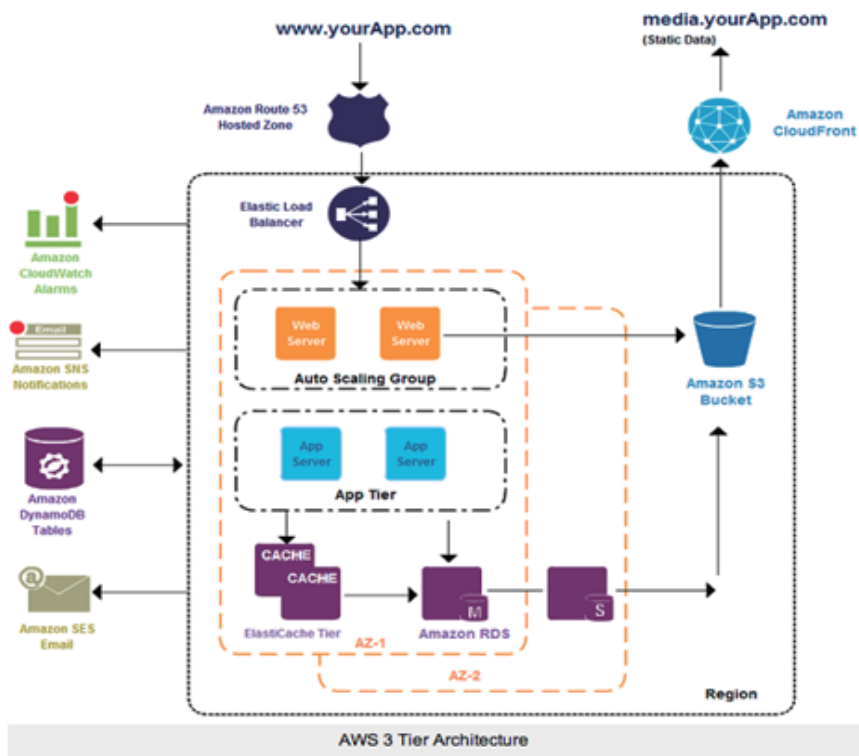


图 27 跨可用区的高可靠性部署方案

注：把业务分布在不同的可用区之间，每个可用区里，EC2 提供虚拟服务器；EBS 提供块设备存储；S3 提供基于 WEB 的数据访问和备份。ELB 提供负载均衡和容错，自动的把流量分布在不同的可用区的 EC2 实例上。S3 做总体的数据备份。CloudFront 作为 CDN 加速。

## 4.3 AWS 的高可靠性

亚马逊深刻认识到 AWS 服务的高可靠性是 AWS 能够在竞争中胜出的最重要的手段之一。毕竟在云计算和大数据的年代，没有哪一个企业愿意承担网站宕机和数据丢失的灾难后果。

在 IaaS 基础架构方面，亚马逊主要是通过如下方法保证高可靠性的。

- \* 遍布全球的 AWS 区域，区域之间独立，避免单点失效。

- \* 区域内部署多个可用区，可用区之间对应物理独立的数据中心，最大限度的避免单点失效。

- \* 可用区之间互联通过高速，低延时网络互联，但不跨越区域，最大限度的避免单点失效。

- \* 通过 ELB 配合，租户可以通过部署其应用在不同的可用区内做负载均衡和容错。

- \* 通过 CDN CloudFront，做全球加速，提高用户体验和堵塞拥挤。

- \* 通过 Route 53 DNS 服务，最大程度的保证租户的 DDoS(Distributed Denial of Service) 攻击的安全。

- \* 通过 Security Group, 虚拟防火墙, ACL 和虚拟机隔离的方式，最大程度的保护租户的 VM 的安全。

- \* 通过虚拟硬盘的方式，保护租户的数据不被盗用和毁坏。

在上述的技术支撑下，亚马逊的 IaaS AWS 在商务上对个人和企业租户承诺了如干 SLA 保证。其分别为：EC2 SLA, S3 SLA, RDS SLA 和 Route53 SLA。其中，RDS SLA 是 2013 年 6 月才开始承诺的。

SLA 是租户和提供 AWS 服务的亚马逊公司的一个合约。合约提供亚马逊 AWS 提供的服务级别和如果达不到这个标准亚马逊公司付出的赔偿。

## **EC2 SLA 的承诺 [20]:**

承诺日期：October 23, 2008

承诺定义：

“AWS will use commercially reasonable efforts to make Amazon EC2 available with an Annual Uptime Percentage (defined below) of at least

99.95% during the Service Year. In the event Amazon EC2 does not meet the Annual Uptime Percentage commitment, you will be eligible to receive a Service Credit as described below.”

注：EC2 99.95%的 SLA 承诺是基于年为单位的 99.9%的服务保证，一年不能多于 4 小时 22 分钟 58.5 秒的不可用机时间。如果换算为星期和月的话，是一个星期不能多于 5 分钟 2.4 秒；一个月不能多于 21 分钟 54.9 秒的不可用机时间。但亚马逊为了保护自己的利益，EC2 是基于年为单位的 SLA，意味着，基于星期和月的统计不做考虑。

亚马逊对 EC2 不可用 (Unavailable) 的定义如下：

“Region Unavailable” and “Region Unavailability” means that more than one Availability Zone in which you are running an instance, within the same Region, is “Unavailable” to you.

注：“区域不可用”意味着你的虚拟机在同一个区域内有多于一个的可用区”不可用”。

“Unavailable” means that all of your running instances have no external connectivity during a five minute period and you are unable to launch replacement instances.

注：“不可用”是指一个租户所有的虚拟机与外面 (Internet) 失去连接长达 5 分钟，而且不能通过其他的虚拟机来替换。

### **S3 的 SLA 承诺 [21]:**

承诺日期：October 1, 2007

服务承诺：

“AWS will use commercially reasonable efforts to make Amazon S3 available with a Monthly Uptime Percentage (defined below) of at least 99.9% during any monthly billing cycle (the “Service Commitment”). In the event Amazon S3 does not meet the Service Commitment, you will be eligible to receive a Service Credit as described below.”

注：S3 的承诺是基于月为单位的 99.9%的服务保证，意味着一个星期不能多于 10 分钟 4.8 秒。一个月不能多于 43 分钟 49.7 秒。一年不能多于 8 小时 45 分钟 57 秒。基于月为单位的 SLA，意味着，每个新月，统计清零。所以基于星期和年的统计不做考虑。

亚马逊对 S3 错误 (ErrorRate) 的定义如下：

“Error Rate” means: (i) the total number of internal server errors returned by Amazon S3 as error status “InternalError” or “ServiceUnavailable” divided by (ii) the total number of requests during that five minute period. We will calculate the Error Rate for each Amazon S3 account as a percentage for each five minute period in the monthly billing cycle. The calculation of the number of internal server errors will not include errors that arise directly or indirectly as a result of any of the Amazon S3 SLA Exclusions (as defined below).

注：错误率是在 5 分钟内对 S3 所有的服务请求中，返回“内部错误”或者“服务无法可用”的比例。AWS 会对每个 S3 的租户进行 5 分钟为周期的取样。

### **RDS 的 SLA 承诺 [22]:**

承诺日期：June 1, 2013

服务承诺：

“AWS will use commercially reasonable efforts to make Multi-AZ instances available with a Monthly Uptime Percentage (defined below) of at least 99.95% during any monthly billing cycle (the "Service Commitment"). In the event Amazon RDS does not meet the Monthly Uptime Percentage commitment, you will be eligible to receive a Service Credit as described below.”

注：RDS 的承诺是基于月为单位的 99.95% 的服务保证。

亚马逊对 RDS 不可用的定义如下：

"Unavailable" means that all connection requests to the running Multi-AZ instance fail during a 1 minute period.

### **Route 53 的 SLA[23]:**

承诺日期：May 24, 2011

服务承诺：

“AWS will use commercially reasonable efforts to make Amazon Route



53 100% Available (defined below). In the event Amazon Route 53 does not meet the foregoing commitment, you will be eligible to receive a Service Credit as described below.”

注：DNS 服务 Route 53 的承诺是基于 100%的服务保证。

亚马逊对 Route 53 100% 可用性的定义如下：

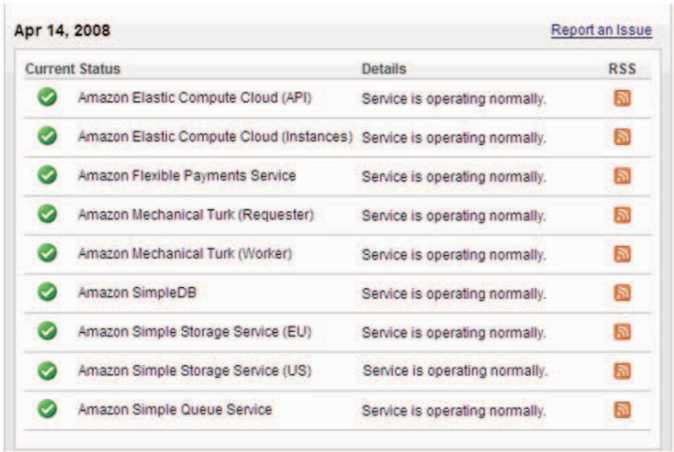
“100% Available” means that Amazon Route 53 did not fail to respond to your DNS queries during a monthly billing cycle.”










注：Route 53 在每个月的基础上在任何时间都能响应 DNS 的请求

# 4.4 Amazon AWS 服务宕机调查 (2006–2013)

Amazon AWS 自 2006 年 3 月 14 开放 S3 文件存储服务和 2006 年 8 月 25 日的 EC2 服务，2008 年 8 月的 EBS 服务以来，经历过许多服务崩溃下线。其中包括 EC2, S3 和 EBS 等。其影响面涉及到租用其服务的许多重要的互联网公司。

AWS 在 2008 年 2 月 16 日，AWS 的 S3 发生严重的服务宕机并导致许多 AWS 的用户的服务中断。Amazon 的 AWS 团队进行了深刻的反思，并在 4 月 8 日，开始提供 AWS Service Health Dashboard，每天跟踪发布各种服务的可靠性。



Apr 14, 2008			<a href="#">Report an Issue</a>
Current Status	Details	RSS	
✓ Amazon Elastic Compute Cloud (API)	Service is operating normally.		
✓ Amazon Elastic Compute Cloud (Instances)	Service is operating normally.		
✓ Amazon Flexible Payments Service	Service is operating normally.		
✓ Amazon Mechanical Turk (Requester)	Service is operating normally.		
✓ Amazon Mechanical Turk (Worker)	Service is operating normally.		
✓ Amazon SimpleDB	Service is operating normally.		
✓ Amazon Simple Storage Service (EU)	Service is operating normally.		
✓ Amazon Simple Storage Service (US)	Service is operating normally.		
✓ Amazon Simple Queue Service	Service is operating normally.		

本节试图对 AWS 上线的重大宕机事件做一个整理列表，并做相应的讨论。

## 1. 2006 年 4 月 1 日

Amazon 在开放其 S3 存储服务不到一个月，在 2006 年 4 月 1 日，S3 发生宕机事件 [24]。

事故影响 :S3

事故恢复 :406 分钟


事故解释：AWS 团队做 S3 存储的负载均衡的管理调配。结果导致内部网络负载崩溃，从而使得 S3 子系统服务宕机。

### Re: Http/1.1 Service Unavailable - Outage?

 Reply

Posted by:  Dave Barth REAL NAME™

Posted on: Apr 4, 2006 5:38 PM

 in response to: Dave Barth

Amazon S3 Customers,

Below is an update on the issues causing the Amazon S3 outage last Saturday evening, which we have isolated and resolved.

We were taking the low-load Saturday as an opportunity to perform some maintenance on the storage system, specifically on some very large (> 100 million objects) buckets in order to obtain better load-balancing characteristics. Normally this procedure is entirely transparent to users and bucket owners. In this case, the re-balancing caused an internal transit link to become flooded, this cascaded into other network problems, and the system was made unavailable.

We are taking several steps to ensure that we don't run into this situation again. We are modifying our maintenance procedures, and are adding further monitoring to prevent the transit link from getting full. In addition we are modifying the way that our system makes use of the network to prevent the cascading effect we saw on Saturday.

Providing world-class reliability is our top priority for Amazon S3. We appreciate your patience, and hope to surpass your expectations going forward.

-dave

2. 2007 年 3 月 6 日

Amazon 的 EC2 发生宕机，有些客户丢失了数据。EC2 API 管理功能被短暂的停止使用 [25]。

事故影响：EC2

事故恢复：40 分钟

事故解释：N/A


数据中心：US-EAST-1

### Re: EC2 Outage?

 Reply

Posted by:  Attila@AWS

Posted on: Mar 6, 2007 10:08 AM

 in response to: Anupam Singhal

Today a number of EC2 instances experienced a partial network outage starting at 01:05 PST. At 01:45 PST, full network functionality was restored to the vast majority of impacted instances.

This partial network outage was caused by a system software upgrade which disabled network connectivity for some instances.

We have taken steps to ensure that this specific problem during upgrades will no longer happen, including modifications to the gated steps we use to promote software to full production.

We apologize for any inconvenience that this has caused.

Sincerely,  
EC2 Team

### 3. 2007 年 9 月 27 日

Amazon 的 EC2 发生宕机，有些客户丢失了数据。EC2 API 管理功能被短暂的停止使用 [26]。

事故影响：EC2

事故恢复：463 分钟

事故解释：Amazon 的 AWS 团队的解释是 AWS 的一些管理软件错误的设置导致了一些客户的虚拟机被误杀。当时为了确保整个 AWS 服务的安全，AWS 团队迅速暂时停止了 EC2 的管理 API 功能

数据中心：全部

#### Re: EC2 API outage

Posted by:  Peter@AWS

Posted on: Sep 29, 2007 12:04 PM

★ in response to: [mediawonder](#)

 Reply

This is an update on the EC2 issues experienced today. A software deployment caused our management software to erroneously terminate a small number of user's instances. When our monitoring detected this issue, the EC2 management software and APIs were disabled to prevent further terminations. Once we corrected the problem, we restored the management software.

We will contact users that lost instances directly by email. At this point, the service is fully functional, and you should be able to launch replacement instances immediately.

While we have corrected the immediate bug, we are also adding additional checks to prevent this sort of issue from recurring in the future.

We are aware of the following outstanding issues which we are working to resolve now:

1/ Some instances may get stuck in the "shutting down" state until we have completed our clean-up. These instances will not be billed and will be fully terminated shortly.

2/ Some instances will not show their launch indexes in describe-instances API.

We will keep you posted as we resolve these remaining issues.

To address a few of the questions posed on this thread:

The availability of the EC2 APIs is very important and it remains our goal to keep them highly available. We believe disabling the management software was the correct decision because of the risk to running instances. This is not a decision we take lightly, and we will work to avoid having to make this choice in the future.

There was no correlation between the instance terminations, so users with redundancy built into their instance deployments would have been better able to deal with the terminations. We also understand that failure isolation is very important and we are hard at work on additional functionality to help with this.

Please let us know if you experience any unexpected behavior.

The Amazon EC2 Team

### 4. 2008 年 2 月 15 日

这是 Amazon 官方对外承认和解释的第一次重大事故。也从根本的角度影响了产业界对公有云可靠性的认识和警惕。并直接导致了 Amazon 决定加强服务可用性的监管和透明化 [27]。

事故影响：S3

事故恢复：198 分钟

事故解释：S3 服务子系统的认证 (Authentication) 服务无法承受突然的大面积的服务请求，从而导致 S3 系统瘫痪。

数据中心：全部



***For one of our services, the Amazon Simple Storage Service, one of our three geographic locations was unreachable for approximately two hours and was back to operating at over 99% of normal performance before 7 a.m. pst. We've been operating this service for two years and we're proud of our uptime track record. Any amount of downtime is unacceptable and we won't be satisfied until it's perfect. We've been communicating with our customers all morning via our support forums and will be providing additional information as soon as we have it.***

Early this morning, at 3:30am PST, we started seeing elevated levels of authenticated requests from multiple users in one of our locations. While we carefully monitor our overall request volumes and these remained within normal ranges, we had not been monitoring the proportion of authenticated requests. Importantly, these cryptographic requests consume more resources per call than other request types.

Shortly before 4:00am PST, we began to see several other users significantly increase their volume of authenticated calls. The last of these pushed the authentication service over its maximum capacity before we could complete putting new capacity in place. In addition to processing authenticated requests, the authentication service also performs account validation on every request Amazon S3 handles. This caused Amazon S3 to be unable to process any requests in that location, beginning at 4:31am PST. By 6:48am PST, we had moved enough capacity online to resolve the issue.

As we said earlier today, though we're proud of our uptime track record over the past two years with this service, any amount of downtime is unacceptable. As part of the post mortem for this event, we have identified a set of short-term actions as well as longer term improvements. We are taking immediate action on the following: (a) improving our monitoring of the proportion of authenticated requests; (b) further increasing our authentication service capacity; and (c) adding additional defensive measures around the authenticated calls. Additionally, we've begun work on a service health dashboard, and expect to release that shortly.

Sincerely, The Amazon Web Services Team

5. 2008 年 6 月 5 日

Amazon 在东部弗吉尼亚的数据中心遭到雷电击，导致该区域的一些 EC2 服务宕机 [28].

事故影响：雷电

事故恢复：N / A

事故解释：雷电导致东部弗吉尼亚的数据中心失去电力，导致 EC2 宕机。  
数据中心：US-EAST



6. 2008 年 6 月 6 日

基于 Amazon 本身的网上零售业务突然宕机，主要是美国和英国的业务，但 AWS 本身没有出现异常 [29]。

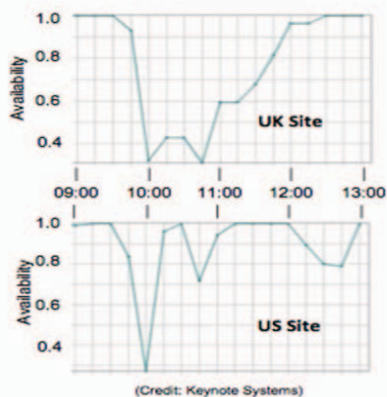
事故影响：Amazon 没有对这次事故作出任何官方解释，只是在非正式场合解释了“Amazon 的网络系统非常复杂，出了点小事情是很偶然和正常的...”。

事故恢复：159 分钟

事故解释：

由于 Amazon 没有对事故作出正式的解释，业界的猜测是 Amazon 的负载均衡业务，例如 DNS 服务出现了问题。另外一种说法是 Amazon 遭到了木马的恶意 DDoS 攻击。证据是，在 Amazon 主站点宕机的同时，Amazon 拥有的 IMDB 站点 (<http://www.imdb.com>) 被 DDoS 通过流量和层 7 放大攻击，其攻击流量大概是 3Mbits/sec。下图是当天 Amazon 美国和英国站点的宕机情况。

数据中心：未知



7. 2008 年 7 月 20 日

S3 再次发生重大宕机事故。许多重要的客户受到影响，例如 Twitter。Twitter 所有的图像基本上都是存放在 Amazon 的 S3 系统里 [30]。

事故影响：S3

事故恢复：498 分钟

事故解释：S3 服务器之间控制信息流不收敛，从而导致 S3 服务器无法处理任何用户的服务请求。同时 Amazon 也承认 EC2 的服务也受到了影响。有些客户的虚拟机无法运行。另外，Simple Queue Service (SQS) 的服务也受到了冲击和中断。

数据中心：全部

### Amazon S3 Availability Event: July 20, 2008

We wanted to provide some additional detail about the problem we experienced on Sunday, July 20th.

At 8:40am PDT, error rates in all Amazon S3 datacenters began to quickly climb and our alarms went off. By 8:55am PDT, error rates were significantly elevated and very few requests were completing successfully. By 8:55am PDT, we had multiple engineers engaged and investigating the issue. Our alarms pointed at problems processing customer requests in multiple places within the system and across multiple data centers. While we began investigating several possible causes, we tried to restore system health by taking several actions to reduce system load. We reduced system load in several stages, but it had no impact on restoring system health.

At 9:41am PDT, we determined that servers within Amazon S3 were having problems communicating with each other. As background information, Amazon S3 uses a gossip protocol to quickly spread server state information throughout the system. This allows Amazon S3 to quickly route around failed or unreachable servers, among other things. When one server connects to another as part of processing a customer's request, it starts by gossiping about the system state. Only after gossip is completed will the server send along the information related to the customer request. On Sunday, we saw a large number of servers that were spending almost all of their time gossiping and a disproportionate amount of servers that had failed while gossiping. With a large number of servers gossiping and failing while gossiping, Amazon S3 wasn't able to successfully process many customer requests.

At 10:32am PDT, after exploring several options, we determined that we needed to shut down all communication between Amazon S3 servers, shut down all components used for request processing, clear the system's state, and then reactivate the request processing components. By 11:05am PDT, all server-to-server communication was stopped, request processing components shut down, and the system's state cleared. By 2:20pm PDT, we'd restored internal communication between all Amazon S3 servers and began reactivating request processing components concurrently in both the US and EU.

At 2:57pm PDT, Amazon S3's EU location began successfully completing customer requests. The EU location came back online before the US because there are fewer servers in the EU. By 3:10pm PDT, request rates and error rates in the EU had returned to normal. At 4:02pm PDT, Amazon S3's US location began successfully completing customer requests, and request rates and error rates had returned to normal by 4:58pm PDT.



8. 2009 年 6 月 10 日

AWS 的 EC2 发生重大宕机事故。其原因是数据中心遭到雷电击,失去电力[31].

事故影响: EC2

事故恢复: 270 分钟

事故解释: 自然气候, 雷电使得数据中心失去电力.

数据中心: US-EAST



9. 2009 年 7 月 19 日

AWS 的 EC2 发生性能和宕机事故 [32].

事故影响: EC2

事故恢复: 89 分钟

事故解释: N/A

数据中心: US-EAST





The outage is the second in a month for Amazon Web Services, following a June 11 incident in which a [lightning strike](#) damaged power equipment at one of the company's data centers, disrupting service for some AWS customers. Today's problems come at a time of growing scrutiny of the reliability of cloud computing providers. EC2 previously experienced extended outages in [February 2008](#) and [October 2007](#).

10. 2009 年 10 月 5 日

Bitbucket 公司 ( 一个在线开源项目服务公司 ) 在 AWS 上的业务宕机 19 个小时 [33].

事故影响 : EC2, EBS

事故恢复 : 1140 分钟

事故解释 : Bitbucket 在 AWS 上的服务被黑客用流量攻击的方法打瘫服务 . 最开始使用的是 UDP Flooding. 然后转换为 TCP 的 Flooding. 服务停顿了 19 个小时 . AWS 的运维团队在处理过程中表现的缺乏经验 .

数据中心 : US-EAST

## DDoS attack against Bitbucket darkens Amazon cloud

At least 16 hours elapsed before Amazon acknowledged nature of attack, says Bitbucket

By Paul McNamara on Mon, 10/05/09 - 8:53am.

2 Comments Print in Share +1 1 Like 1 More

A crippling DDoS attack over the weekend against open-source hosting service [Bitbucket](#) and Amazon's [EC2 service](#) has questions being raised about the speed and effectiveness of Amazon's response to the emergency, as well as the general reliability of cloud services.

Bottom line for Bitbucket? They're considering switching to a different provider, several of whom were "concerned" enough to offer their services while Bitbucket and Amazon struggled to stem the tide. Bitbucket has almost 19,000 users.

I've requested comment from Amazon's public relations department. (Update: Amazon reply below.)

From a Bitbucket [blog post](#) yesterday by Jesper Noehr:

As many of you are well aware, we've been experiencing some serious downtime the past couple of days. Starting Friday evening, our network storage became virtually unavailable to us, and the site crawled to a halt.

We're hosting everything on Amazon EC2, aka. "the cloud", and we're also using their EBS service for storage of everything from our database, logfiles, and user data (repositories.)

11. 2009 年 12 月 9 日

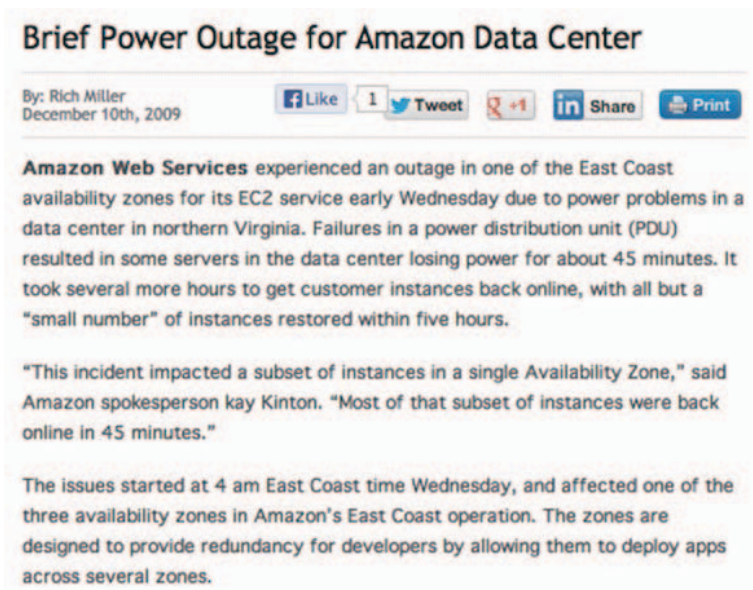
AWS 的 EC2 发生宕机事故。其原因是数据中心遭到雷电击，失去电力。地点发生在东部北弗吉尼亚的数据中心 [34].

事故影响：EC2

事故恢复：300 分钟

事故解释：自然气候，雷电使得数据中心失去电力。

数据中心：US-EAST



### Brief Power Outage for Amazon Data Center

By: Rich Miller  
December 10th, 2009

[Like](#) [1](#) [Tweet](#) [+1](#) [Share](#) [Print](#)

**Amazon Web Services** experienced an outage in one of the East Coast availability zones for its EC2 service early Wednesday due to power problems in a data center in northern Virginia. Failures in a power distribution unit (PDU) resulted in some servers in the data center losing power for about 45 minutes. It took several more hours to get customer instances back online, with all but a "small number" of instances restored within five hours.

"This incident impacted a subset of instances in a single Availability Zone," said Amazon spokesperson Kay Kinton. "Most of that subset of instances were back online in 45 minutes."

The issues started at 4 am East Coast time Wednesday, and affected one of the three availability zones in Amazon's East Coast operation. The zones are designed to provide redundancy for developers by allowing them to deploy apps across several zones.

12. 2009 年 12 月 23 日

AWS 的 DNS 服务出现问题，无法访问 [35].

事故影响：DNS

事故恢复：132 分钟

事故解释：N/A

数据中心：US-WEST

### Re: Amazon DNS outage?

 Reply

Posted by: KC

Posted on: Dec 23, 2009 5:28 PM

📌 in response to: KC

replying to myself...

a source says this is an ultrasdn problem affecting primarily west coast.

### Re: Amazon DNS outage?

 Reply

Posted by:  Jason@AWS

Posted on: Dec 23, 2009 6:46 PM

📌 in response to: KC

We have been tracking an issue and have posted regarding this on the Service Health Dashboard:

<http://status.aws.amazon.com>

13. 2010 年 5 月 4 日

AWS 的 EC2 遭受电源事故 . 影响到了一个数据中心的部分用户 . 有部分用户丢失了 EBS 数据 [36].

事故影响 : 电源

事故恢复 :240 分钟

事故解释 : 技术人员切换电源系统时操作失误 .

数据中心 :US-EAST

## Summary: Amazon EC2 experienced a power outage due to data center technicians shifting power to a new substation

Amazon EC2 experienced two power outages on May 4 and an extended power loss early on Saturday, May 8. In each case, a group of users in a single availability zone lost service, while the majority of EC2 users remained unaffected. The first outage on May 4 occurred as data center technicians were shifting power to a new substation from the local power utility. "During a power cut-over at 2:22 a.m. PDT, a single UPS failed to appropriately transfer power to the back-up generators," Amazon reported. The UPS unit failed to detect a drop in input power and shift the load to batteries, and subsequently the generator. Amazon was able to bypass the problematic UPS and provide generator power directly to the affected racks. Amazon of customer instances were recovered by 3:40 a.m. Pacific, and virtually all of them were recovered.

Number Affected: Unknown

Organization: Amazon Web Services

- Affected Service: Amazon Elastic Compute Cloud (Amazon EC2)

Reported Date: 2010-05-04

Incident Duration: 4 hours

14. 2010 年 5 月 4 日

AWS 的 EC2 遭受第 2 次电源事故 . 影响到了一个数据中心的部分用户 . 有部分用户丢失了 EBS 数据 [37].

事故影响 : 电源

事故恢复 :55 分钟

事故解释 : 在试图恢复第一个事故的时候 , 启动了备用电源 . 技术人员的错误操作使得备用电源也失去动力 .

数据中心 :US-EAST

**Summary:** Amazon EC2 experienced a power outage due to a human error causing the back-up generator to lose power

Amazon EC2 experienced two power outages on May 4 and an extended power loss early on Saturday, May 8. In each case, a group of users in a single availability zone lost service, while the majority of EC2 users remained unaffected. Later that day, while the affected racks were still being powered by the back-up generator, a human error caused the back-up generator to lose power, Amazon said. The generator was reset and power was again restored to the racks, with most customers regaining service between 6:45 p.m. and 7:40 p.m. Pacific time. The next day Amazon completed installation of a replacement UPS unit.

Number Affected: Unknown

Organization: [Amazon Web Services](#)

- Affected Service: [Amazon Elastic Compute Cloud \(Amazon EC2\)](#)

Reported Date: 2010-05-04

Incident Duration: 55 minutes

15. 2010 年 5 月 8 日

AWS 的 EC2 遭受电源事故 . 影响到了一个数据中心的部分用户的 EC2 和 EBS 服务 [38].

事故影响 : 电源

事故恢复 : 420 分钟

事故解释 : 一个数据中心的一些机架的电源故障 .

数据中心 :US-EAST

## Amazon Addresses EC2 Power Outages

By: Rich Miller

May 10th, 2010



Print

**Amazon Web Services** says it is making changes in its data centers to address a series of power outages last week that affected some users of its EC2 cloud computing service. Amazon EC2 experienced two power outages on May 4 and an extended power loss early on Saturday, May 8. In each case, a group of users in a single availability zone lost service, while the majority of EC2 users remained unaffected.

"As a result of the recent power events, we are working on non-intrusive changes to our power distribution architecture to significantly reduce the number of instances that can be affected by failures like we have seen in the last week," Amazon said Sunday in an incident report on its status dashboard. "This work is under way and will be carefully performed over the coming months throughout our datacenters."

### Seven-Hour Outage

Saturday's outage began at about 12:20 a.m. and lasted until 7:20 a.m., and affected a "set of racks," according to Amazon, which said the bulk of customers in its U.S. East availability zone remained unaffected.

16. 2010 年 5 月 13 日

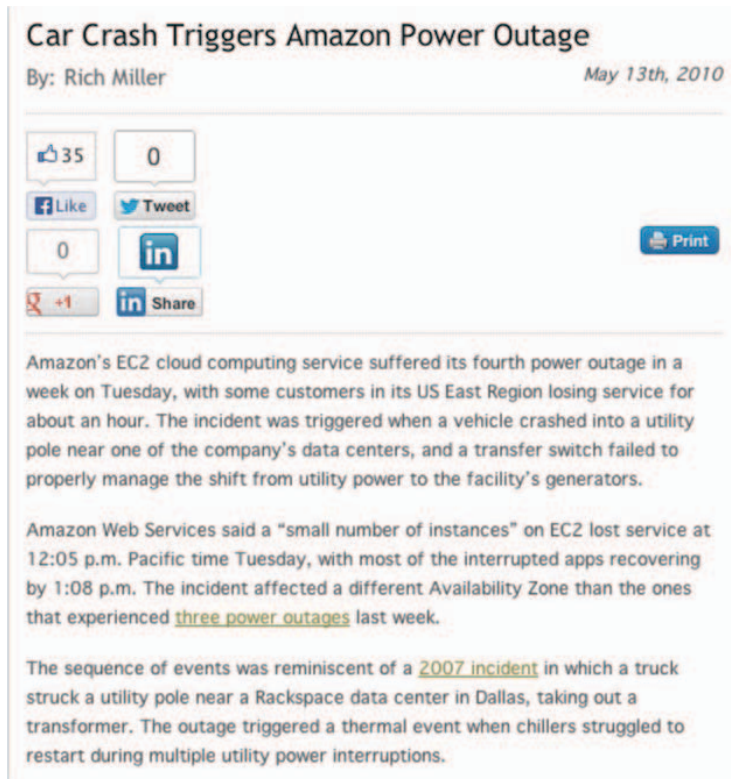
AWS 的 EC2 遭受电源事故。这使得 Amazon 在 5 月份 1 周内连续 4 次由于电源问题导致服务宕机 [39]。

事故影响：电源

事故恢复：63 分钟

事故解释：东部的一个数据中心由于汽车事故撞坏了电线杆，导致 EC2 服务宕机。

数据中心：US-EAST



17. 2010 年 6 月 27 日

Amazon 的主站零售业务服务宕机。事故影响面主要在 Amazon 的零售主业务上，没有对 AWS 的公有云服务造成影响。Amazon 当天的股票大跌 7.8%[40]。

事故影响：N / A

事故恢复：250 分钟

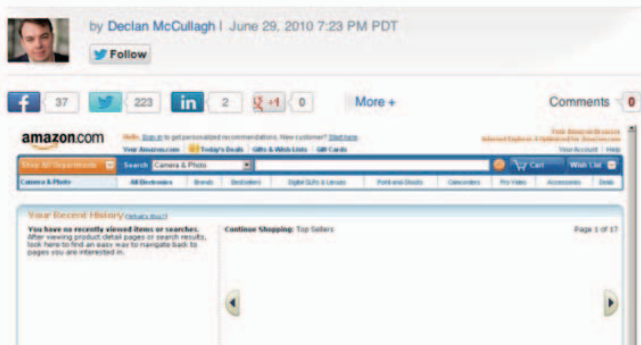
事故解释：Amazon 没有对事故作出相应的技术解释。

数据中心：未知



# Amazon.com experiences hours-long outage

Online retailer experiences an unusually long outage that frustrates customers by displaying blank product pages and empty shopping carts.



18. 2010 年 9 月 28 日

Amazon 位于东部的 EC2 USA-EAST-1 的数据中心宕机，EC2 服务受到影响 [41].

事故影响：EC2

事故恢复：60 分钟

事故解释：数据中心网络出现问题，导致 EC2 服务中断。

数据中心：US-EAST-1

EC2's USA-EAST-1 datacenter experienced network connectivity issues.	
<b>Number Affected:</b> Unknown <b>Organization:</b> Amazon Web Services <ul style="list-style-type: none"><li>Affected Service: <u>Amazon Elastic Compute Cloud (Amazon EC2)</u></li></ul>	<b>Reported Date:</b> 2010-09-28
	<b>Occurred Date:</b> 2010-09-28
	<b>Incident Duration:</b> 1 hour

19. 2010 年 10 月 1 日

CloudFront 服务宕机器 [42].

事故影响 : CloudFront 控制 API 连接超时  
事故恢复 : 45 分钟  
事故解释 : N/A  
数据中心 :US-EAST-1

Summary: CloudFront API Experiences Increased Latency	
Requests to CloudFront's control API experienced increased latency. According to Amazon, CloudFront's ability to serve content was not affected.	
Number Affected: Unknown Organization: <a href="#">Amazon Web Services</a> <ul style="list-style-type: none"><li>Affected Service: <a href="#">Amazon CloudFront</a></li></ul>	<div>Reported Date: 2010-10-01</div> <div>Occurred Date: 2010-10-01</div> <div>Incident Duration: 45 minutes</div>

20. 2010 年 10 月 8 日

EC2 服务宕机 [43] .

事故影响 : CloudFront Auto-Scaling Triggers Fail  
事故恢复 : N/A  
事故解释 : N/A  
数据中心 : 未知

Summary: CloudFront Auto-Scaling Triggers Fail	
Auto-scaling triggers from the US West-1 region were not executing.	
Number Affected: Unknown Organization: <a href="#">Amazon Web Services</a> <ul style="list-style-type: none"><li>Affected Service: <a href="#">Amazon Elastic Compute Cloud (Amazon EC2)</a></li></ul>	<div>Reported Date: 2010-10-08</div> <div>Occurred Date: 2010-10-08</div>

21. 2010 年 10 月 12 日

EBS 服务性能下降和宕机 [44].

事故影响 : 东部数据中心的 EBS 服务出现显著延时 .



事故恢复 : N/A

事故解释 : N/A

## Summary: Amazon ESB Experiences Performance Latencies

The us-east-1 region experienced increased latency on read/write operations for a single availability zone.

Number Affected: Unknown

Organization: [Amazon Web Services](#)

- Affected Service: [Amazon Elastic Block Store \(Amazon EBS\)](#)

**Reported Date:** 2010-10-12

**Occurred Date:** 2010-10-12

数据中心 : 未知

22. 2010 年 10 月 26 日

东部数据中心 EC2 出现连接问题 [45].

事故影响 : 东部数据中心的 EBS 服务出现显著延时 .

事故恢复 : 4500 分钟

事故解释 : N/A

数据中心 : US-EAST-1

## Summary: EC2 US East Region Outage

The US-EAST-1 region instances experienced connectivity issues.

Number Affected: Unknown

Organization: [Amazon Web Services](#)

- Affected Service: [Amazon Elastic Compute Cloud \(Amazon EC2\)](#)

**Reported Date:** 2010-10-26

**Occurred Date:** 2010-10-26

**Incident Duration:** 3 days and 3 hours

23. 2010 年 10 月 28 日

西部数据中心 EC2 API 出现连接问题 [46].

事故影响 : 西部数据中心的 EC2 出现连接错误 .

事故恢复 :N / A  
事故解释 : SSL 证书过期导致 EC2 API 错误 .  
数据中心 : US-WEST-1

**Summary: Expired SSL Cert Cause API Failures**

EC2 API calls made over HTTPS to the US-West-1 region experienced elevated error rates. This was the result of an expired SSL certificate.

Number Affected: Unknown Organization: Amazon Web Services <ul style="list-style-type: none"><li>Affected Service: Amazon Elastic Compute Cloud (Amazon EC2)</li></ul>	Reported Date:	2010-10-28
	Occurred Date:	2010-10-28

24. 2010 年 11 月 1 日

Amazon 在欧洲的数据中心的 EC2 和 EBS 服务宕机 [47].

事故影响 : EC2 和 EBS API 服务出现异常的高错误率 .  
事故恢复 :N/A  
事故解释 : N/A  
数据中心 : EU-WEST-1

**Summary: Amazon Web Services EC2 API Outage**

EC2 and EBS API calls in the EU-West-1 regions experienced elevated error rates. Amazon initial restored all API calls with the exception of CreateSnapshot. Eventually Amazon restored all the APIs.

Number Affected: Unknown Organization: Amazon Web Services <ul style="list-style-type: none"><li>Affected Service: Amazon Elastic Compute Cloud (Amazon EC2)</li><li>Affected Service: Amazon Elastic Block Store (Amazon EBS)</li></ul>	Reported Date:	2010-11-01
	Occurred Date:	2010-11-01

25. 2010 年 12 月 12 日

Amazon 在欧洲的零售主站点 (www DOT amazon DOT com) 宕机 [48].

事故影响 : EC2 和 EBS API 服务出现异常的高错误率 .  
事故恢复 :115 分钟

事故解释：欧洲的数据中心的 EC2 API 发生了高错误率。

数据中心：EU-WEST-1

**Summary:** Amazon.com websites in Europe suffered an outage for more than half an hour due to a hardware failure in its European data center network.

According to internet monitoring firm Netcraft, Amazon's service health dashboard reported "elevated error rates and latencies for the EC2 APIs in the EU-WEST-1 region". Netcraft said many other sites may have been affected such as Amazon accounts for more than a third of all web-facing computers in Ireland.

Number Affected: Unknown  
Organization: Amazon  
• Affected Service: Amazon.com

Reported Date: 2010-12-12

26. 2011 年 2 月 26 日

东部的一个数据中心的 EC2 无法提供服务 [49].

事故影响：EC2

事故恢复：484 分钟

事故解释：网络维护导致


数据中心：US-EAST-1

**Re: Partial Power/Network outage on us-east-1b ?**

 Reply

Posted by:  Luke@AWS

Posted on: Feb 26, 2011 8:24 AM

 in response to: cznuvol

cznuvol,

The event is resolved at this time and the hardware on which your instance resides is in an optimal state. I'm unable to make a connection to this instance, do you have a firewall running besides your current security group settings that may be permitting outside access?

27. 2011 年 4 月 21 日

AWS 遭受到了有史以来最大的一次宕机事故。东部的一个数据中心的 EC2 和 RDS 服务突然全面无法提供服务。导致大批客户在线服务停顿，例如，著名的 foursquare 等 [50].

事故影响：EC2

事故恢复：5170 分钟

事故解释：EBS 子系统崩溃导致 EC2 服务全面失效。

## Summary of the Amazon EC2 and Amazon RDS Service Disruption in the US East Region

April 29, 2011

Now that we have fully restored functionality to all affected services, we would like to share more details with our customers about the events that occurred with the Amazon Elastic Compute Cloud ("EC2") last week, our efforts to restore the services, and what we are doing to prevent this sort of issue from happening again. We are very aware that many of our customers were significantly impacted by this event, and as with any significant service issue, our intention is to share the details of what happened and how we will improve the service for our customers.

The issues affecting EC2 customers last week primarily involved a subset of the Amazon Elastic Block Store ("EBS") volumes in a single Availability Zone within the US East Region that became unable to service read and write operations. In this document, we will refer to these as "stuck" volumes. This caused instances trying to use these affected volumes to also get "stuck" when they attempted to read or write to them. In order to restore these volumes and stabilize the EBS cluster in that Availability Zone, we disabled all control APIs (e.g. Create Volume, Attach Volume, Detach Volume, and Create Snapshot) for EBS in the affected Availability Zone for much of the duration of the event. For two periods during the first day of the issue, the degraded EBS cluster affected the EBS APIs and caused high error rates and latencies for EBS calls to these APIs across the entire US East Region. As with any complicated operational issue, this one was caused by several root causes interacting with one another and therefore gives us many opportunities to protect the service against any similar event reoccurring.

数据中心 : US-EAST-1

注 :Amazon IaaS EBS 子系统在这次事故之后，又频频出事。EBS 为了可靠性加大了系统的复杂性，结果是 Amazon IaaS 里最不可靠的元素。教训极大。

28. 2011 年 8 月 7 日

AWS 在欧洲爱尔兰的数据中心遭遇雷电，变压器起火从而导致数据中心的电力系统崩溃 [51]。

事故影响 : EC2, EBS

事故恢复 : 1180 分钟

事故解释 : 电力失效。

数据中心 : EU-WEST

<ul style="list-style-type: none"> <li>Affected Service: EC2</li> <li>Organization: Amazon</li> <li>Number Affected: Unknown</li> </ul>	<p>Occurred Date: 2011-08-03</p> <p>Reported Date: 2011-08-03</p>
<p><b>Summary: Amazon EU datacenter in Dublin outage due to lightning</b></p>	

注 : 8 月 11 日，爱尔兰当地的电力公司声称事故影响不是变压器的问题。

29. 2011 年 8 月 8 日

AWS 在美国东北部弗吉尼亚的数据中心发生宕机事件，影响了许多著名的互联网公司 的正常服务，其中包括 Reddit, Quora, Netflix 和 FourSquare 等 [52].

事故影响：EC2, EBS

事故恢复：25 分钟

事故解释：网络故障

数据中心：US-EAST-1

# Down Goes The Internet... Again. Amazon EC2 Outage Takes Down Foursquare, Instagram, Quora, Reddit, Etc



Monday, August 8th, 2011

104 Comments

TC Tip	Amazon ec2 is down... Again
TC Tip	AWS outage
TC Tip	EngineYard is down.... AGAIN!
TC Tip	Amazon EC2 down again, taking the web with it
TC Tip	#AWS goes down along with @Fab @Quora,
TC Tip	AWS Down!

Are you trying to use the web right now? Just stop. It's largely broken.

As indicated by about 20 tips in the last few minutes and pretty much all of Twitter, Amazon's EC2 service appears to be down. That means services like Reddit, Heroku, Foursquare, Instagram, Fab, Quora, Turntable.fm, Netflix and many, many others are down.

If this sounds familiar, it's because **it just happened this past April**. So far, it looks like the outage has been **going on for about 30 minutes**.

**Update:** It looks like the outage may be **isolated to EAST-1**, so not all of EC2. Still, all of the companies above and hundreds of others are clearly affected right now.

**Update 2:** And after roughly 40 minutes of downtime, the Internet appears to be coming back online. Amazon's **status site** confirms that it's being resolved.

30. 2011 年 11 月 26 日

AWS 在欧洲的数据中心 DNS 服务发生故障，导致 ELB, Route 53 等服务受到影响 [53].

事故影响：ELB, Route 53

事故恢复：7 分钟

事故解释：DNS

数据中心 : EU-WEST  
31. 2012 年 2 月 16 日

AWS 在美国西部的数据中心发生宕机事件 [54].

事故影响 : EC2, EBS  
事故恢复 : 15 分钟  
事故解释 : 未知  
数据中心 : US-WEST-1

### AWS - Second Outage This Week - Why?

Posted by: endertech

Posted on: Feb 17, 2012 1:12 PM

ebs , ec2 , aws , outage

This question is **not answered**. Helpful answers available: **2**. Correct answers available: **1**.

This afternoon is the second occurrence of AWS North California region experiencing an outage.

First: Feb 16th @ 2:30am - about 10-15min down  
Second: Feb 17th @ 11:50am - about 10-15min down

This being the same exact kind of outage for a second time this week I think we're due for an explanation as to why exactly connectivity was impaired to EC2 instances and why EBS volumes became degraded. Why?

Replies: 9 | Pages: 1 - Last Post: Mar 14, 2012 2:38 PM by: endertech

32. 2012 年 2 月 17 日

AWS 在美国西部的数据中心发生宕机事件 [55].

事故影响 : EC2, EBS  
事故恢复 : 15 分钟  
事故解释 : 未知  
数据中心 : US-WEST-1

### AWS - Second Outage This Week - Why?

Posted by: endertech

Posted on: Feb 17, 2012 1:12 PM

ebs , ec2 , aws , outage

This question is **not answered**. Helpful answers available: **2**. Correct answers available: **1**.

This afternoon is the second occurrence of AWS North California region experiencing an outage.

First: Feb 16th @ 2:30am - about 10-15min down  
Second: Feb 17th @ 11:50am - about 10-15min down

This being the same exact kind of outage for a second time this week I think we're due for an explanation as to why exactly connectivity was impaired to EC2 instances and why EBS volumes became degraded. Why?

Replies: 9 | Pages: 1 - Last Post: Mar 14, 2012 2:38 PM by: endertech

33. 2012 年 3 月 8 日

AWS 在美国西部的数据中心发生宕机事件 [56].


事故影响 : EC2, EBS

事故恢复 : 38 分钟

事故解释 : 未知

数据中心 : US-WEST-1

**Re: AWS - Second Outage This Week - Why?**

Posted by:  [MattJ@AWS](#)

Posted on: Mar 8, 2012 4:59 PM

📧 in response to: [swong-aws](#)

Hello,

We are currently investigating issues in this region. We have updated the status on <http://status.aws.amazon.com/> with the following:

"4:51 PM PST We are investigating connectivity issues for EC2 in the US-WEST-1 Region."

Please keep an eye on the status for more information.

Best,

Matt J

34. 2012 年 6 月 13 日

AWS 在美国东部的数据中心发生宕机事件 [57].

事故影响 : EC2, EBS

事故恢复 : 974 分钟

事故解释 : SimpleDB 的握手控制信号超时 .

数据中心 : US-EAST

#### Summary of the Amazon SimpleDB Service Disruption

We wanted to share what we've learned from our investigation of the June 13 SimpleDB disruption in the US East Region. The service was unavailable to all API calls (except a fraction of the eventually consistent read calls) from 9:16 PM to 11:16 PM (PDT). From 11:16 PM to 1:30 AM, we continued to have elevated error rates for CreateDomain and DeleteDomain API calls.

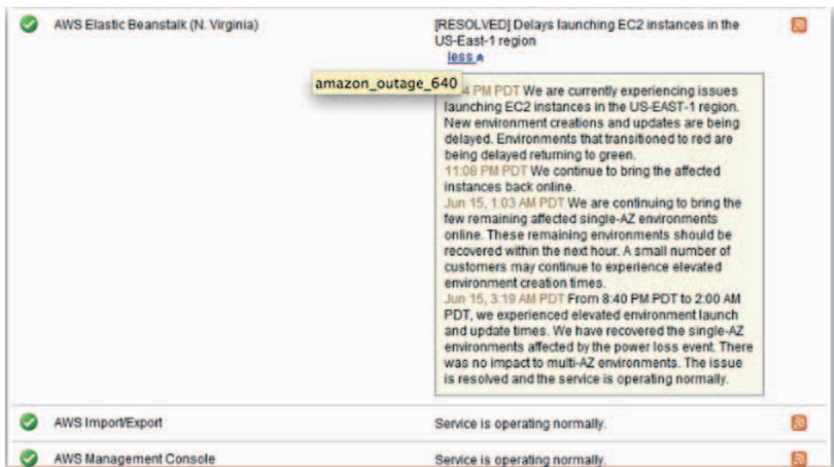
SimpleDB is a distributed datastore that replicates customer data across multiple data centers. The service employs servers in various roles. Some servers are responsible for the storage of user data ("storage nodes"), with each customer Domain replicated across a group of storage nodes. Other nodes store metadata about each customer Domain ("metadata nodes"), such as which storage nodes it is located on. SimpleDB uses an internal lock service to determine which set of nodes are responsible for a given Domain. This lock service itself is replicated across multiple data centers. Each node handshakes with the lock service periodically to verify it still has responsibility for the data or metadata it hosts.

In this event, multiple storage nodes became unavailable simultaneously in a single data center (after power was lost to the servers on which these nodes lived). While SimpleDB can handle multiple simultaneous node failures, and has successfully endured larger infrastructure failures in the past without incident, the server failure pattern in this event resulted in a sudden and significant increase in load on the lock service as it rapidly de-registered the failed storage nodes from their respective replication groups. This simultaneous volume resulted in elevated handshake latencies between healthy SimpleDB nodes and the lock service, and the nodes were not able to complete their handshakes prior to exceeding a set "handshake timeout" value. After several handshake retries and subsequent timeouts, SimpleDB storage and metadata nodes removed themselves from the SimpleDB production cluster, and SimpleDB API requests returned error messages (http response code 500 for server-side error). The affected storage nodes were not able to rejoin the SimpleDB cluster and serve API requests until receiving authorization to rejoin from metadata nodes. This process ensures that we do not allow a node with stale data to join the production cluster accidentally and start serving customer requests. However, in this case the metadata nodes were also down due to the same handshake timeout issue, and therefore could not authenticate the storage nodes.

35. 2012 年 6 月 15 日

AWS 在美国东部北弗吉尼亚的数据中心发生宕机事件 [58].

事故影响 : EC2, EBS  
事故恢复 : 240 分钟  
事故解释 : 电源故障  
数据中心 : US-EAST-1



36. 2012 年 6 月 29 日

AWS 在美国东部北弗吉尼亚的数据中心再次发生宕机事件 . NetFlix, Instagram 和 Pinterest 的服务都受到影响 [59].

事故影响 : EC2  
事故恢复 : 400 分钟  
事故解释 : 雷电  
数据中心 : US-EAST



# Friday Night, Taking Netflix, Instagram And Pinterest With It

43 comments, 16 called-out + Comment Now + Follow Comments



Netflix  
@Netflixhelps

Follow



We're aware that some members are experiencing issues streaming movies and TV shows. We're working to resolve the problem.

Reply Retweet Favorite

14

RETWEETS



11:50 PM - 29 Jun 12 via web · Embed this Tweet

As of 11:21 PM EST Amazon's [Elastic Compute Cloud](#) in North Virginia went down, due to severe thunder storms in the area. [The Washington Post](#) reports torrential rains, "scary winds," lightning and massive power outages in the D.C. area.

## Summary of the AWS Service Event in the US East Region

July 2, 2012

We'd like to share more about the service disruption which occurred last Friday night, June 29th, in one of our Availability Zones in the US East-1 Region. The event was triggered during a large scale electrical storm which swept through the Northern Virginia area. We regret the problems experienced by customers affected by the disruption and, in addition to giving more detail, also wanted to provide information on actions we'll be taking to mitigate these issues in the future.

Our US East-1 Region consists of more than 10 datacenters structured into multiple Availability Zones. These Availability Zones are in distinct physical locations and are engineered to isolate failure from each other. Last Friday, due to weather warnings of the approaching storm, all change activity in the US East-1 Region had been cancelled and extra personnel had been called into the datacenters for the evening.

On Friday night, as the storm progressed, several US East-1 datacenters in Availability Zones which would remain unaffected by events that evening saw utility power fluctuations. Backup systems in those datacenters responded as designed, resulting in no loss of power or customer impact. At 7:24pm PDT, a large voltage spike was experienced by the electrical switching equipment in two of the US East-1 datacenters supporting a single Availability Zone. All utility electrical switches in both datacenters initiated transfer to generator power. In one of the datacenters, the transfer completed without incident. In the other, the generators started successfully, but each generator independently failed to provide stable voltage as they were brought into service. As a result, the generators did not pick up the load and servers operated without interruption during this period on the Uninterruptable Power Supply ("UPS") units. Shortly thereafter, utility power was restored and our datacenter personnel transferred the datacenter back to utility power. The utility power in the Region failed a second time at 7:57pm PDT. Again, all rooms of this one facility failed to successfully transfer to generator power while all of our other datacenters in the Region continued to operate without customer impact.

37. Oct 22, 2012

AWS 在美国东北部弗吉尼亚的数据中心再次发生 EBS 子系统宕机事件 [60].

事故影响: EC2, EBS, RDS, ELB

事故恢复: 380 分钟

事故解释: EBS 子系统崩溃

# Amazon Web Services Outage Caused By Memory Leak And Failure In Monitoring Alarm



ALEX WILLIAMS

Saturday, October 27th, 2012

27 Comments



A memory leak and a failed monitoring system caused the **Amazon Web Services outage on Monday** that took out Reddit and other major services.

According to a **post** Friday night, AWS explained that the problem arose after a simple replacement of a data collection server. After installation, the server did not propagate its DNS address correctly and so a fraction of servers did not get the message. Those servers kept trying to reach the server, which led to a memory leak that then went out of control due to the failure of an internal monitoring alarm. Eventually the system ground to a virtual stop and millions of customers felt the pain.

AWS:



*By Monday morning, the rate of memory loss became quite high and consumed enough memory on the affected storage servers that they were unable to keep up with normal request handling processes.*

The failure in its North Virginia region eventually interrupted Reddit, Foursquare, Minecraft, Heroku, GitHub, Imgur, Pocket, HipChat, Coursera and a number of others.



Sign Up

My Account / Console

English

AWS Products & Solutions

AWS Product Information



Developers

Support

## Summary of the October 22, 2012 AWS Service Event in the US-East Region

We'd like to share more about the service event that occurred on Monday, October 22nd in the US-East Region. We have now completed the analysis of the events that affected AWS customers, and we want to describe what happened, our understanding of how customers were affected, and what we are doing to prevent a similar issue from occurring in the future.

### The Primary Event and the Impact to Amazon Elastic Block Store (EBS) and Amazon Elastic Compute Cloud (EC2)

At 10:00AM PDT Monday, a small number of Amazon Elastic Block Store (EBS) volumes in one of our five Availability Zones in the US-East Region began seeing degraded performance, and in some cases, became "stuck" (i.e. unable to process further I/O requests). The root cause of the problem was a latent bug in an operational data collection agent that runs on the EBS storage servers. Each EBS storage server has an agent that contacts a set of data collection servers and reports information that is used for fleet maintenance. The data collected with this system is important, but the collection is not time-sensitive and the system is designed to be tolerant of late or missing data. Last week, one of the data collection servers in the affected Availability Zone had a hardware failure and was replaced. As part of replacing that server, a DNS record was updated to remove the failed server and add the replacement server. While not noticed at the time, the DNS update did not successfully propagate to all of the internal DNS servers, and as a result, a fraction of the storage servers did not get the updated server address and continued to attempt to contact the failed data collection server. Because of the design of the data collection service (which is tolerant to missing data), this did not cause any immediate issues or set off any alarms. However, this inability to contact a data collection server triggered a latent memory leak bug in the reporting agent on the storage servers. Rather than gracefully deal with the failed connection, the reporting agent continued trying to contact the collection server in a way that slowly consumed system memory. While we monitor aggregate memory consumption on each EBS Server, our monitoring failed to alarm on this memory leak. EBS Servers generally make very dynamic use of all of their available memory for managing customer data, making it difficult to set accurate alarms on

38. Dec 24, 2012

圣诞节，AWS 在美国东北部弗吉尼亚的数据中心再次发生宕机事件，这次的起因是负载均衡子系统 ELB[61]。

事故影响：ELB

事故恢复：24 小时

事故解释：ELB 子系统

数据中心：US-EAST

## Updated: Netflix Crippled On Christmas Eve By AWS Outages



CATHERINE SHU ✓

Monday, December 24th, 2012

234 Comments



'Twas the night before Christmas, when all through the house, not a creature was stirring, not even those in the lighthearted holiday fare you were planning to watch on Netflix, which has been down since 1pm PST for some customers. Netflix confirmed the outage with a tweet on its official channel, though Netflix Cloud Architect Adrian Cockcroft [said on Twitter](#) that the service is still working on some

devices.



Sign Up

My Account / Console ▾

English ▾

AWS Products & Solutions ▾

AWS Product Information ▾



Developers ▾

Support ▾

### Summary of the December 24, 2012 Amazon ELB Service Event in the US-East Region

We would like to share more details with our customers about the event that occurred with the Amazon Elastic Load Balancing Service ("ELB") earlier this week in the US-East Region. While the service disruption only affected applications using the ELB service (and only a fraction of the ELB load balancers were affected), the impacted load balancers saw significant impact for a prolonged period of time.

The service disruption began at 12:24 PM PST on December 24th when a portion of the ELB state data was logically deleted. This data is used and maintained by the ELB control plane to manage the configuration of the ELB load balancers in the region (for example tracking all the backend hosts to which traffic should be routed by each load balancer). The data was deleted by a maintenance process that was inadvertently run against the production ELB state data. This process was run by one of a very small number of developers who have access to this production environment. Unfortunately, the developer did not realize the mistake at the time. After this data was deleted, the ELB control plane began experiencing high latency and error rates for API calls to manage ELB load balancers. In this initial part of the service disruption, there was no impact to the request handling functionality of running ELB load balancers because the missing ELB state data was not integral to the basic operation of running load balancers.

39. 2013 年 1 月 31 日

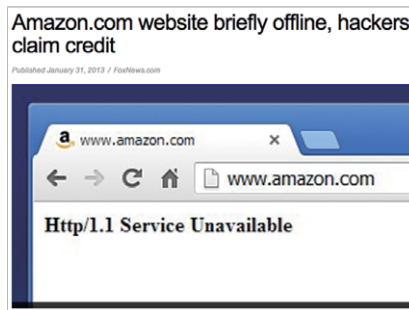
Amazon 主站点和 AWS 的 CloudFront 遭受事故。用户无法登录购买零售商品。

事故影响：Amazon 主站点前端 [62].

事故恢复：50 分钟

事故解释：黑客攻击

数据中心：未知



40. 2013 年 6 月 16 日

AWS 内部 DNS 解释错误 [63].

事故影响：DNS

事故恢复：13 分钟

事故解释：DNS

数据中心：US-EAST-1

Re: DNS outage in one AZ

Posted by:  MattJ@AWS

Posted on: Jun 16, 2013 5:54 PM

✉ In response to: nattahnaf

 Reply

Hello,

This was related to an event that was posted on the AWS Status Health page at [status.aws.amazon.com](http://status.aws.amazon.com) that read:

3:42 PM PDT We are investigating DNS resolution issues in the US-EAST-1 Region

3:56 PM PDT From 3:08PM to 3:39PM PDT in the US-EAST-1 Region some EC2 instances could not resolve internal EC2 instance DNS names. The issue has been resolved and the service is operating normally.

Regards,

Matt J

41. 2013 年 8 月 18 日

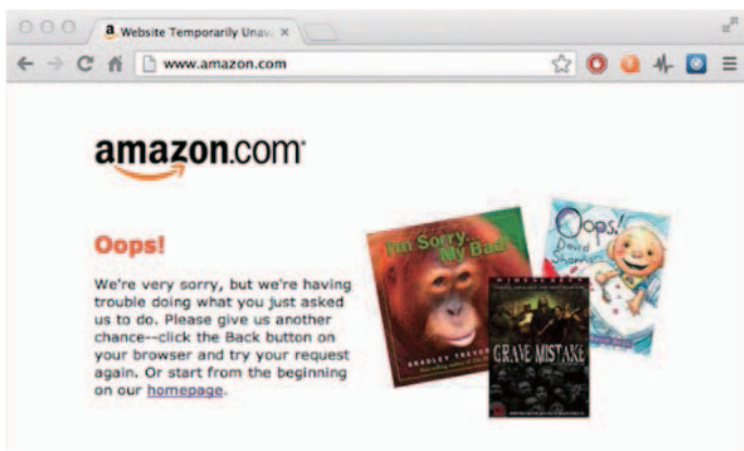
Amazon 主站点遭受事故． 用户无法登录购买零售商品 [64].

事故影响：Amazon 主站点前端

事故恢复：45 分钟

事故解释：黑客攻击

数据中心：未知



42. 2013 年 8 月 25 日

2013 年 8 月 25 日，AWS 遭受到了一次比较大的宕机事故． 东部的一个数据中心的 EC2 和 RDS 和负载均衡服务突然全无法提供服务． 导致大批客户在线服务停顿，例如，著名的 Instagram, Airbnb 和 Flipboard 等 [65].

事故影响：EC2

事故恢复 :51 分钟

事故解释 : 内部网络问题 , 导致 EBS 子系统性能缓慢 , 从而导致 EC2 服务失效 .  
数据中心 : US-EAST-1


1:22 PM PDT We are investigating degraded performance for some volumes in a single AZ in the US-EAST-1 Region  
1:29 PM PDT We are investigating degraded performance for some EBS volumes and elevated EBS-related API and EBS-backed instance launch errors in a single AZ in the US-EAST-1 Region.  
2:21 PM PDT We have identified and fixed the root cause of the performance issue. EBS backed instance launches are now operating normally. Most previously impacted volumes are now operating normally and we will continue to work on instances and volumes that are still experiencing degraded performance.  
3:23 PM PDT From approximately 12:51 PM PDT to 1:42 PM PDT network packet loss caused elevated EBS-related API error rates in a single AZ, a small number of EBS volumes in that AZ to experience degraded performance, and a small number of EC2 instances to become unreachable due to packet loss in a single AZ in the US-EAST-1 Region. The root cause was a "grey" partial failure with a networking device that caused a portion of the AZ to experience packet loss. The network issue was resolved and most volumes, instances, and API calls returned to normal. The networking device was removed from service and we are performing a forensic investigation to understand how it failed. We are continuing to work on a small number of instances and volumes that require additional maintenance before they return to normal performance.

Instagram alerted its users of a fault to its service almost as soon as it occurred:

We know many of you are having trouble loading Instagram. We identified the issue and are working to fix it ASAP. Thanks for holding on.

— Instagram (@instagram) August 25, 2013

Vine caught up, about half an hour later:

 **Vine**  
@vineapp

Follow

We're aware of some issues affecting our servers and are working to address them now. Thanks for your patience and hang tight!

2:20 PM - 25 Aug 2013

517 RETWEETS 415 FAVORITES

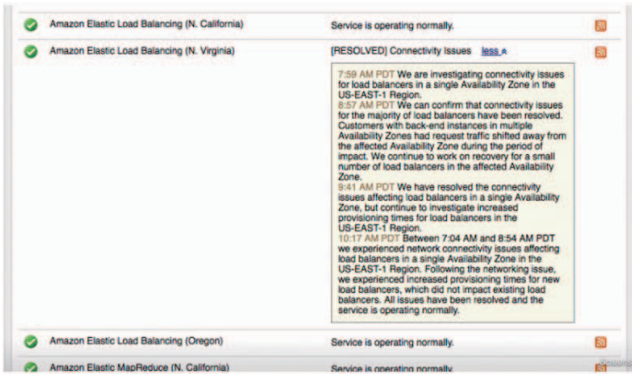
43. September 13, 2013

AWS 在东部的数据中心发生宕机事故 . 9 月 13 日是个黑色的星期五 . 受影响的站点包括著名的 Heroku, Github, CMSWire 等 [66].

事故影响 : EC2, EBS

事故恢复 : 115 分钟 .

事故解释 : 网络问题导致 EBS 和 EC2 失效 .

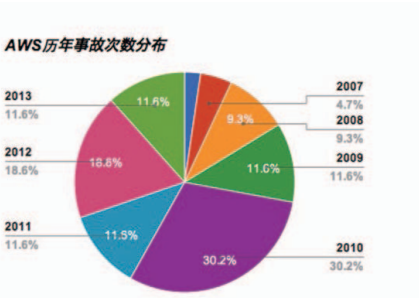


4.5 AWS 服务宕机数据统计分析

AWS 在 2006 年 3 月 14 日以来，大概有统计的宕机 (Outage) 事故为 43 次。每次事故发生的地点，原因，恢复的时间都不一样。这小节试图通过不同的维度来考察过去 8 年来的各种事故，并做一些相应的统计分析。

AWS 历年事故次数分布：

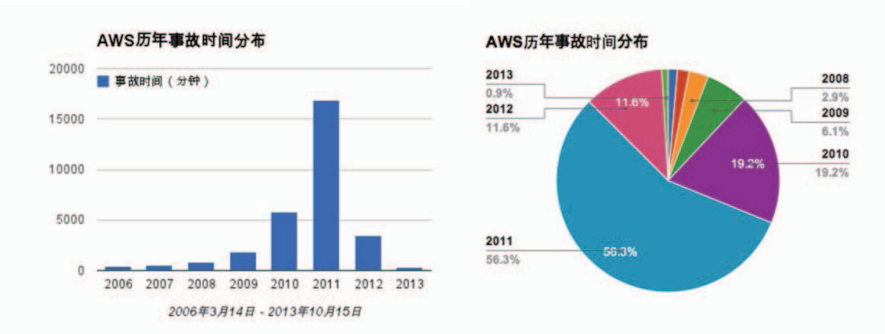
2006 年：1 次	2007 年：2 次	2008 年：4 次
2009 年：5 次	2010 年：13 次	2011 年：5 次
2012 年：8 次	2013 年：5 次	



从上图可见，2010 年是 AWS 过去 8 年来事故次数最多的一年，为 13 次，为 30.2%。

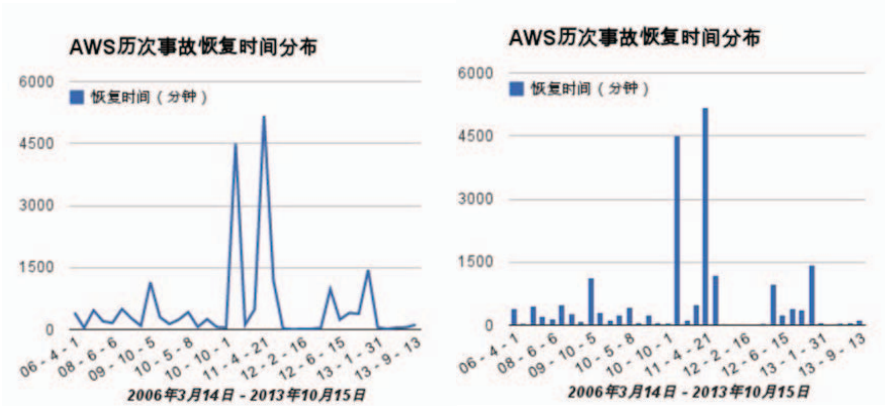
AWS 历年事故时间分布：

2006 年：406 分钟      2007 年：503 分钟      2008 年：855 分钟  
2009 年：1831 分钟      2010 年：5748 分钟      2011 年：6866 分钟  
2012 年：3472 分钟      2013 年：274 分钟



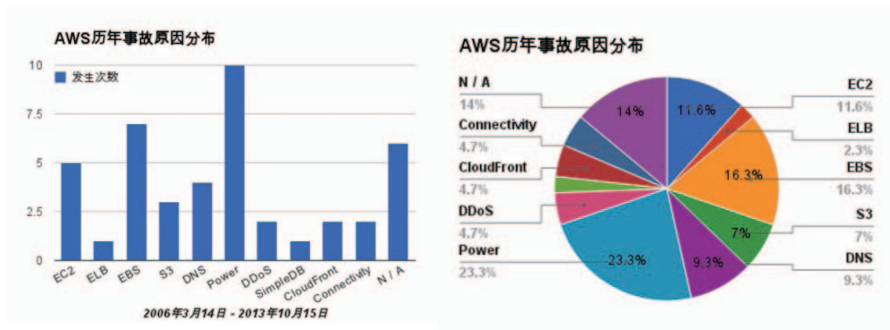
从上图可见，2011 年 AWS 虽然只宕机 5 次，但总分钟数是过去 8 年来时间最长的一年。

AWS 历年事故恢复时间分布：





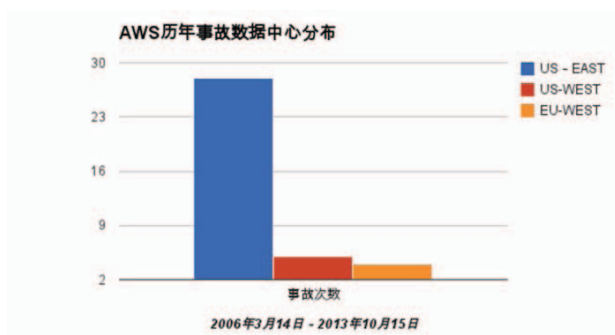
## AWS 历年事故原因分布：



EC2: 5 次 EBS: 7 次 ELB: 1 次 S3: 3 次 DNS: 4 次 Power / Lighting: 10 次 CloudFront: 2 次 DDoS: 2 次 SimpleDB: 1 次 Connectivity: 2 次 N / A: 6 次

从上图可见，电源和雷电对 AWS 数据中心事故目前还是第一主要原因，其次就是 AWS 的块存储服务 EBS. EBS 已经成为最近几年来 AWS 系统最不稳定的一个子系统。

## AWS 历年事故数据中心分布：



在 AWS 宕机事故地域分布方面，在统计到的 43 次事故里，在能确认的案例里，有 28 次是发生在 Amazon 在东部的数据中心 US-EAST, 占 65.1%. 5 次发生在 US-WEST, 占 11.9%, 4 次发生在欧洲的数据中心 EU-WEST, 占 9.3%.

US-EAST:28      US-WEST:5      EU-WEST:4

基本上 AWS 所有大的宕机事故都发生在东部北弗吉尼亚的数据中心。原因是该数据中心比较便宜，建设完成的时间也比较长和老旧。

## 5 总结和讨论

云计算服务在被越来越多的个人和企业所采用，但人们对于云计算服务在安全性，可靠性和服务响应确定性方面的担忧也与日俱增。虽然云服务提供商 (Clouds Service Provider) 通常都会承诺 SLA (Service Level Agreement) 的可用性 (Availability) 范围等，但许多云租户不太理解可用性的内在复杂性，因此在选择云平台时缺乏风险评估能力。

本文首次系统的定义和分析了云计算可用性 (Cloud Computing Availability) 的层次模型，提出了可用单元 AU(Availability Unit), 可用集合 AS(Availability Set), 单独可用性 SA(Standalone Availability) 和部署可用性 DA(Deployment Availability) 等概念，并对云计算 IaaS, PaaS 和 SaaS 各个层次 SA 和 DA 的算法关系展开了讨论。在此基础上，文章提出了扁平 AS(Flat AS) 和复合 AS(CompositeS) 为基本单元的参考设计模型，其中包括一字型架构，菱形架构，人字型架构，三角形架构和胖树架构。并对各种模型优缺点展开了相关的定性分析。

文章最后，对亚马逊的 AWS 公有云体系结构做了相应的讨论，并针对 2006 年到 2013 年以来 AWS 被外界所报道过的服务事故做了相应的统计调查和一些定量分析。从数据分析可以发现除了电力故障之外，许多故障是发生在 AWS 的块存储的 EBS 子系统上。

## 参考文献

- [1] <http://www.gartner.com/newsroom/id/2352816>
- [2] <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html>
- [3] <http://www.newvem.com/aws-cloud-best-practice-introduction-to-high-availability-deployment/>
- [4] <http://www.crn.com/slide-shows/cloud/240144284/the-10-biggest-cloud-outages-of-2012.htm>
- [5] <http://www.datacenterknowledge.com/archives/2012/12/17/the-year-in-downtime-/top-10-outages-of-2012/>
- [6] [http://en.wikipedia.org/wiki/Cloud\\_computing](http://en.wikipedia.org/wiki/Cloud_computing)
- [7] [http://en.wikipedia.org/wiki/Mean\\_time\\_between\\_failures](http://en.wikipedia.org/wiki/Mean_time_between_failures)
- [8] [http://en.wikipedia.org/wiki/Mean\\_time\\_to\\_repair](http://en.wikipedia.org/wiki/Mean_time_to_repair)
- [9] <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html>
- [10] [http://en.wikipedia.org/wiki/Windows\\_Azure](http://en.wikipedia.org/wiki/Windows_Azure)
- [11] <http://www.linuxvirtualserver.org/whatis.html>
- [12] <http://aws.amazon.com/elasticloadbalancing/>
- [13] Bill Highleyman, et. al., Breaking the Availability Barrier, 2004.
- [14] Bill Highleyman, Calculating the Availability, 2006
- [15] <http://aws.amazon.com/about-aws/globalinfrastructure/>
- [16] <http://www.newvem.com/cloudpedia/aws-availability-zones/>
- [17] <http://mytechblog.com/2013/01/aws-regions-availability-zones/>
- [18] Jinesh Varia, Architecting for the Cloud: Best Practices, 2011
- [19] <http://aws.amazon.com/architecture/>
- [20] <http://aws.amazon.com/ec2-sla/>
- [21] <http://aws.amazon.com/s3-sla/>
- [22] <http://aws.amazon.com/rds-sla/>
- [23] <http://aws.amazon.com/route53/sla/>
- [24] <https://forums.aws.amazon.com/thread.jspa?threadID=10185>
- [25] <https://forums.aws.amazon.com/thread.jspa?messageID=55028>
- [26] <https://forums.aws.amazon.com/thread.jspa?threadID=17211>

- [27] <http://www.zdnet.com/blog/btl/amazon-explains-its-s3-outage/8010>
- [28] <http://www.datacenterknowledge.com/archives/2008/06/05/brief-outage-for-amazon-web-services/>
- [29] [http://news.cnet.com/8301-10784\\_3-9962010-7.html](http://news.cnet.com/8301-10784_3-9962010-7.html)
- [30] <http://status.aws.amazon.com/s3-20080720.html>
- [31] <http://www.datacenterknowledge.com/archives/2009/06/11/lightning-strike-triggers-amazon-ec2-outage/>
- [32] <http://www.datacenterknowledge.com/archives/2009/07/19/outage-for-amazon-web-services/>
- [33] <http://www.networkworld.com/community/node/45891>
- [34] <http://www.datacenterknowledge.com/archives/2009/12/10/power-outage-for-amazon-data-center/>
- [35] <https://forums.aws.amazon.com/thread.jspa?messageID=159106&#159106>
- [36] <http://www.datacenterknowledge.com/archives/2010/05/10/amazon-addresses-ec2-power-outages/>
- [37] <http://www.datacenterknowledge.com/archives/2010/05/10/amazon-addresses-ec2-power-outages/>
- [38] <http://www.datacenterknowledge.com/archives/2010/05/10/amazon-addresses-ec2-power-outages/>
- [39] <http://www.datacenterknowledge.com/archives/2010/05/13/car-crash-triggers-amazon-power-outage/>
- [40] [http://news.cnet.com/8301-1023\\_3-20009241-93.html](http://news.cnet.com/8301-1023_3-20009241-93.html)
- [41] <http://cloutage.org/incidents/346-amazon-web-services-amazon-elastic-compute-cloud-amazon-ec2>
- [42] <http://cloutage.org/incidents/320-amazon-web-services-amazon-cloudfront>
- [43] <http://cloutage.org/incidents/299-amazon-web-services-amazon-elastic-compute-cloud-amazon-ec2>
- [44] <http://cloutage.org/incidents/286-amazon-web-services-amazon-elastic-block-store-amazon-ebs>
- [45] <http://cloutage.org/incidents/342-amazon-web-services-amazon-elastic-compute-cloud-amazon-ec2>
- [46] <http://cloutage.org/incidents/364-amazon-web-services-amazon-elastic-compute-cloud-amazon-ec2>

compute/

-cloud-amazon-ec2

[47] [http://cloutage.org/incidents/384-amazon-web-services-amazon-elastic-](http://cloutage.org/incidents/384-amazon-web-services-amazon-elastic-compute-)  
[compute-](http://cloutage.org/incidents/384-amazon-web-services-amazon-elastic-compute-)

[cloud-amazon-ec2-amazon-elastic-block-store-amazon-ebs](http://cloutage.org/incidents/384-amazon-web-services-amazon-elastic-compute-)

[48] <http://cloutage.org/incidents/429-amazon-amazon-com>

[49] <https://forums.aws.amazon.com/thread.jspa?messageID=226458#226458>

[50] <http://aws.amazon.com/message/65648/>

[51] <http://www.datacenterknowledge.com/archives/2011/08/07/lightning-in-dublin-/>  
[knocks-amazon-microsoft-data-centers-offline/](http://www.datacenterknowledge.com/archives/2011/08/07/lightning-in-dublin-/)

[52] <http://techcrunch.com/2011/08/08/amazon-ec2-outage/>

[53] <https://forums.aws.amazon.com/thread.jspa?messageID=295636&#295636>

[54] <https://forums.aws.amazon.com/thread.jspa?messageID=320089&#320089>

[55] <https://forums.aws.amazon.com/thread.jspa?messageID=320089&#320089>

[56] <https://forums.aws.amazon.com/thread.jspa?messageID=320089&#320089>

[57] <http://aws.amazon.com/cn/message/65649/>

[58] <http://www.wired.com/insights/2012/06/amazon-outage-pilot-error/>

[59] <http://aws.amazon.com/message/67457/>

[60] <https://aws.amazon.com/message/680342/>

[61] <http://aws.amazon.com/message/680587/>

[62] <http://www.foxnews.com/tech/2013/01/31/amazoncom-website-offline/>

[63] <https://forums.aws.amazon.com/thread.jspa?messageID=460122&#460122>

[64] <http://gigaom.com/2013/08/19/amazons-u-s-front-end-web-site-is-down/>

[65] [http://www.cruxialcio.com/outage-boots-amazon-web-services-customers-](http://www.cruxialcio.com/outage-boots-amazon-web-services-customers-offline-875)  
[offline-875](http://www.cruxialcio.com/outage-boots-amazon-web-services-customers-offline-875)

[66] [http://www.cloudpro.co.uk/iaas/3300/aws-customers-hit-by-friday-the-](http://www.cloudpro.co.uk/iaas/3300/aws-customers-hit-by-friday-the-thirteenth-outage)  
[thirteenth-outage](http://www.cloudpro.co.uk/iaas/3300/aws-customers-hit-by-friday-the-thirteenth-outage)