彎曲評論
科技 · 人物 · 潮流

# 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)**

**(第二部分)**

**陈怀临, 弯曲评论创办人**
**北极光创投投资顾问, 云基地中云网技术顾问**
**Email: huailin@gmail.com**

## 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SasS, 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:

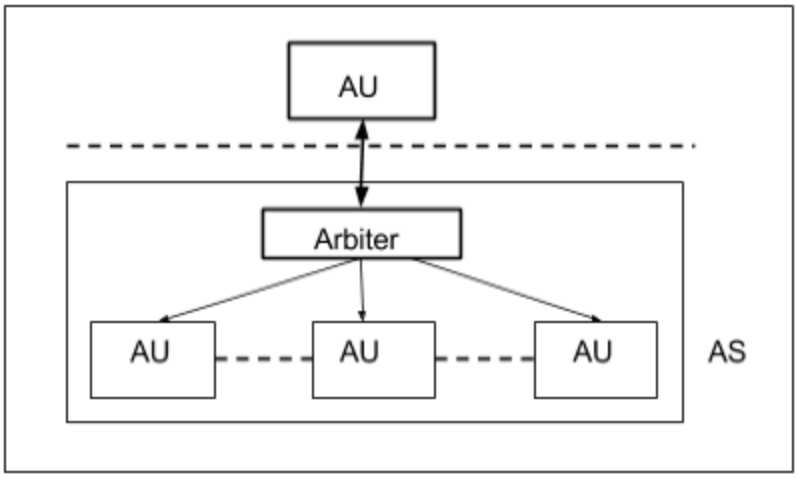



图7 可用单元和可用单元集

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 = $\{AU1, AU2, \ldots, AUm\}$ , 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:

$\forall AU_{i,j}$ (i, j = 1,2,...,m),

**公式 3** $\mathbf{SA_{AUi} = SA_{AUj} = A}$

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.

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

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.

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, $0 \le A \le 1$ , 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. 或者我们说, **1-** $(1-A)^m \ge A$ **.**

**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 ) $\le$ $(1-A)$ . 因此,
1- $(1-A)^2$ $\ge$ 1 - (1- A ) = A

从下图曲线, 我们也可用显然看出函数分布图.

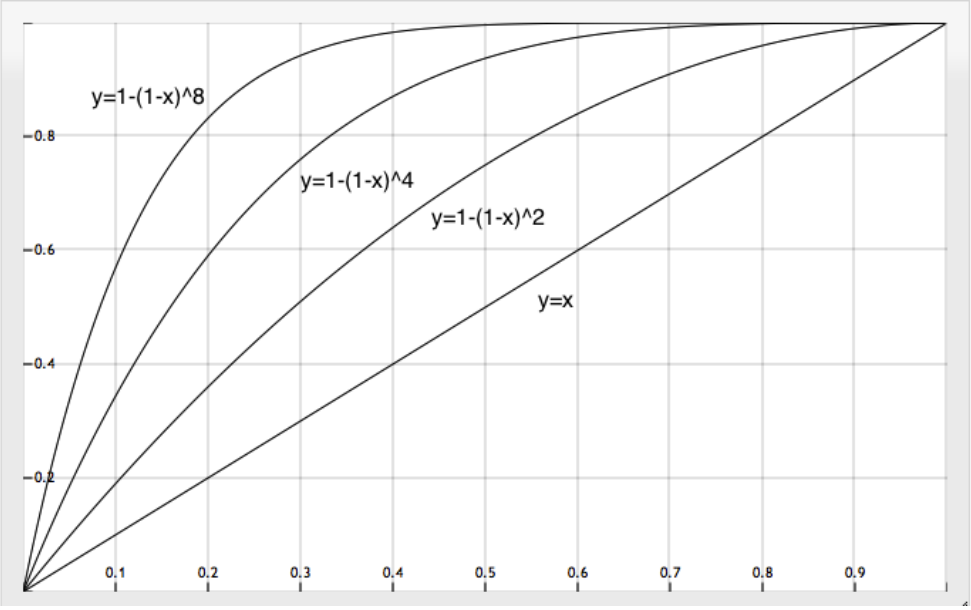
假设1- $(1-A)^m$ $\ge A$ . 下面来考虑1- $(1-A)^{m+1}$ .

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

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

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

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




### 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% * 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:

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

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= $\{AU1, AU2, \ldots, AU\ m\}$ . 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** $$\mathbf{DA_{AS} = 1-(1-DA_{AU\ 1})*(1-DA_{AU\ 2})...*(1-DA_{AU\ m})}$$

假设$DA_{AU\ i} =$ $\mathbf{Max(}DA_{AU\ 1}, DA_{AU\ 2}, ...DA_{AU\ i,}...DA_{AU\ m}\mathbf{)}$,

我们可以容易证明得到,

$$\mathbf{DA_{AU\ i}} \leq \mathbf{DA_{AS}} \leq \mathbf{1-(1-DA_{AU\ i})^m}$$

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 $1-(1-DA_{AU\ i})^m$, 其中$AU_i$是该AS中可用性最大的AU.

**证明:**

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

所以,
$\mathbf{(1-}DA_{AU\ 1}\mathbf{)*(1-}DA_{AU\ 2}\mathbf{)..*(1-}DA_{AU\ i}\mathbf{)...*(1-}DA_{AU\ m}\mathbf{)} \leq \mathbf{(1-}DA_{AU\ i}\mathbf{)}$

因此, $1-(1-DA_{AU\ 1})\mathbf{*(1-}DA_{AU\ 2}\mathbf{)...*(1-}DA_{AU\ m}\mathbf{)} \geq \mathbf{1-\ (1-}DA_{AU\ i}\mathbf{) = DA_{AU\ i}}$

即, $DA_{AU\ i} \leq DA_{AS}$

同理,

$\mathbf{(1-}DA_{AU\ 1}\mathbf{)*(1-}DA_{AU\ 2}\mathbf{)..*(1-}DA_{AU\ i}\mathbf{)...*(1-}DA_{AU\ m}\mathbf{)} \geq$

$(1\text{-}DA_{AU\,i})*(1\text{-}DA_{AU\,i})..*(1\text{-}DA_{AU\,i})...*(1\text{-}DA_{AU\,i}) = (1\text{-}DA_{AU\,i})^m$

因此, $1\text{-}(1\text{-}DA_{AU\,1})*(1\text{-}DA_{AU\,2})...*(1\text{-}DA_{AU\,m}) \leq 1\text{-}(1\text{-}DA_{AU\,i})^m$

即, $DA_{AS} \leq 1\text{-}(1\text{-}DA_{AU\,i})^m$

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8所示.

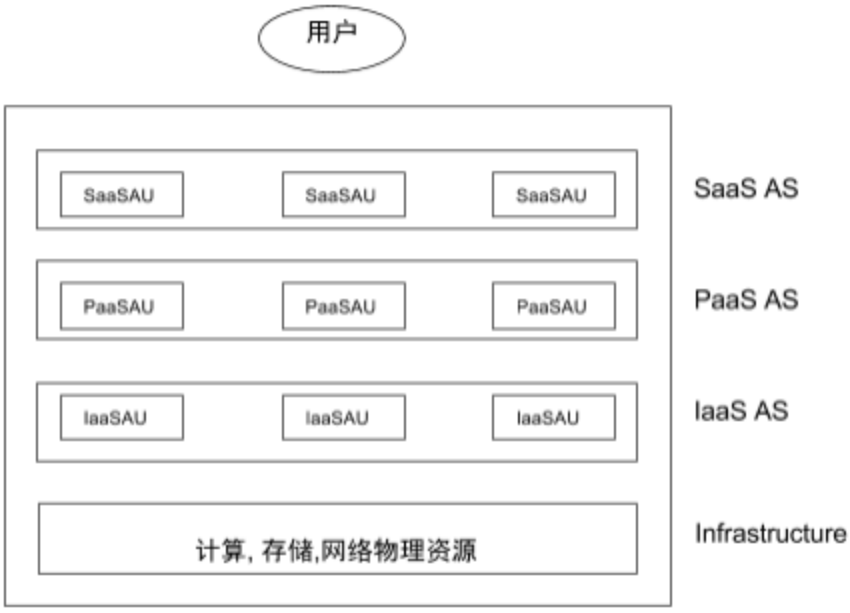



图8 基于AS的云计算模型

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

同理, PaaS AS和IaaS AS的关系是:

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

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.

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

**[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,

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

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

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,通过小规模的抽样测试, 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9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.

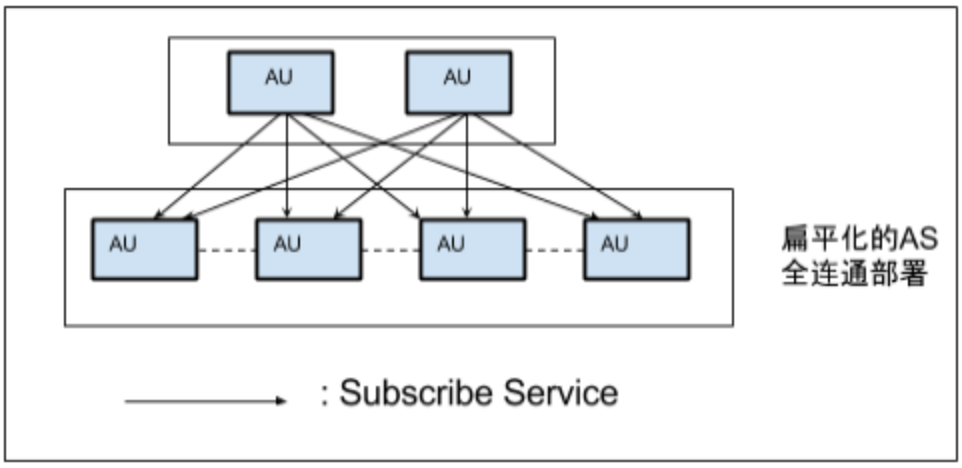



图9 扁平AS拓扑

**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10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.

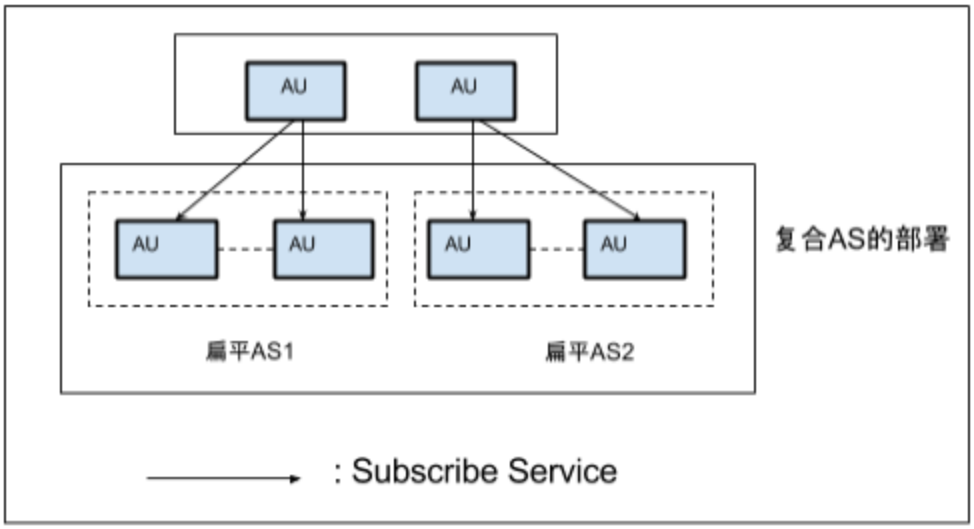



图10 复合AS拓扑

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11所示:

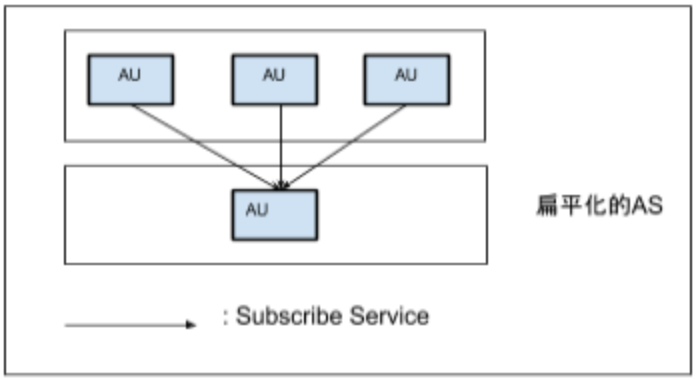



图11 扁平AS拓扑特例-M:1映射

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12所示:

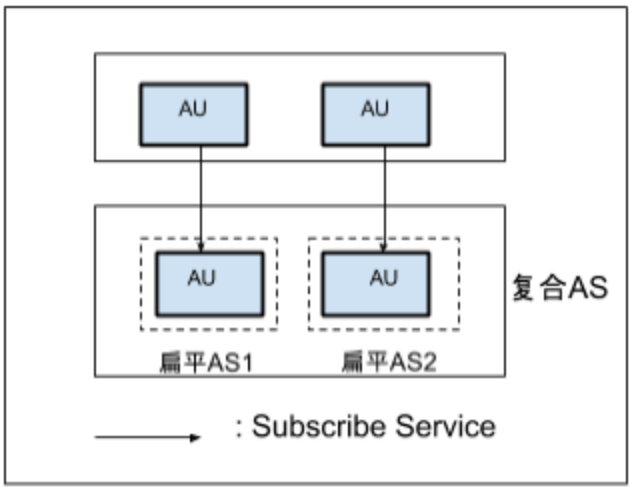



图12 复合AS拓扑特例-1:1映射

图11和图12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9到图12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** $$\mathbf{AS_{sub} = (\sum_{i=1}^{N} AU_{i\,sub})/N}$$

类似地, 一个复合AS的收敛比可以定义为:

**公式13** $$\mathbf{AS_{sub} = (\sum_{i=1}^{M} AS_{i\,sub})/M}$$ , 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9和图10的拓扑. 假设需要服务的AU位于一个PaaS层, 分别为PaaS $AU_1$, PaaS $AU_2$ , 并具有相同的$SA_{PaaS}$ A; 提供服务的IaaS层分别为IaaS $AU_1$, IaaS $AU_2$, IaaS $AU_3$和IaaS $AU_4$ , 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:

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

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

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

因此, 这个PaaS作为一个AS的可用性DA为:

**扁平拓扑可用性:**

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

由公式4, 可推出,

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

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:**

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:

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

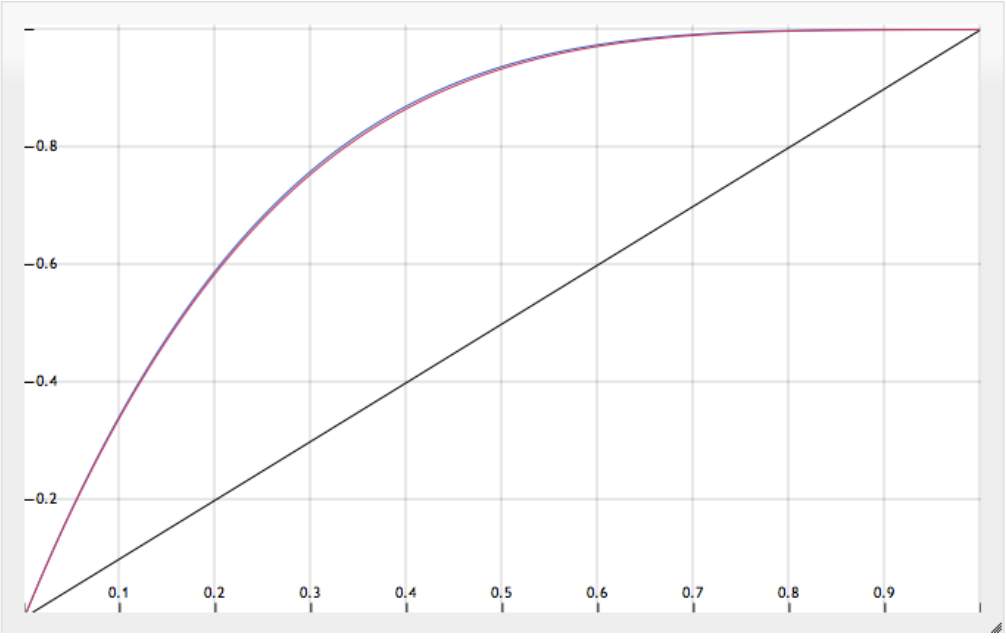
因此,

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

**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.**

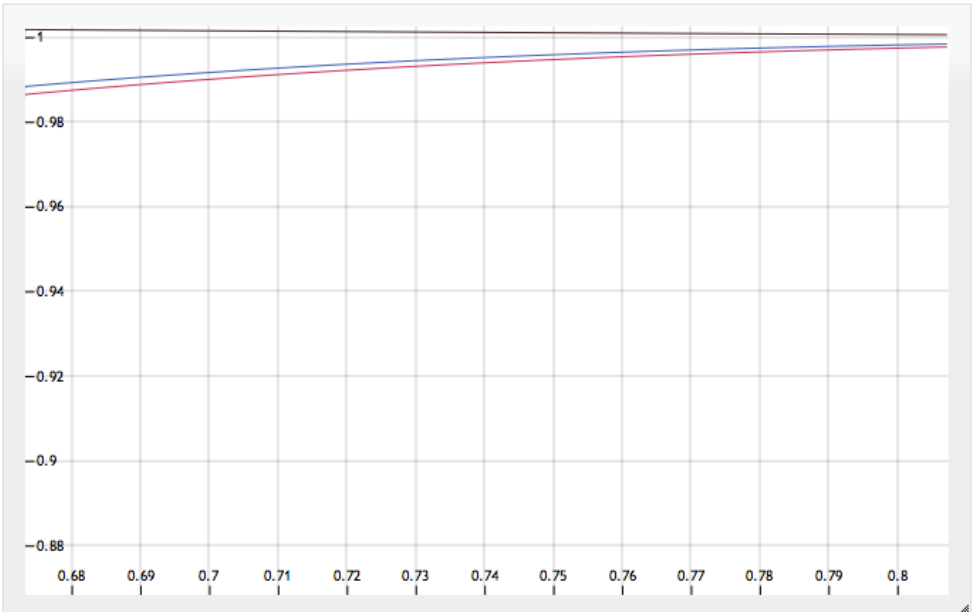
下图所示为假设 A = 0.99的情况下,扁平AS可用性函数为:
y = (1- (1-0.99)^2 ) * (1- (1-x)^4)
和
复合AS可用性函数为:
y = 1 - (1 - 0.99 * (1 - (1-x)^2) )^2
在区间(0, 10)的分布.

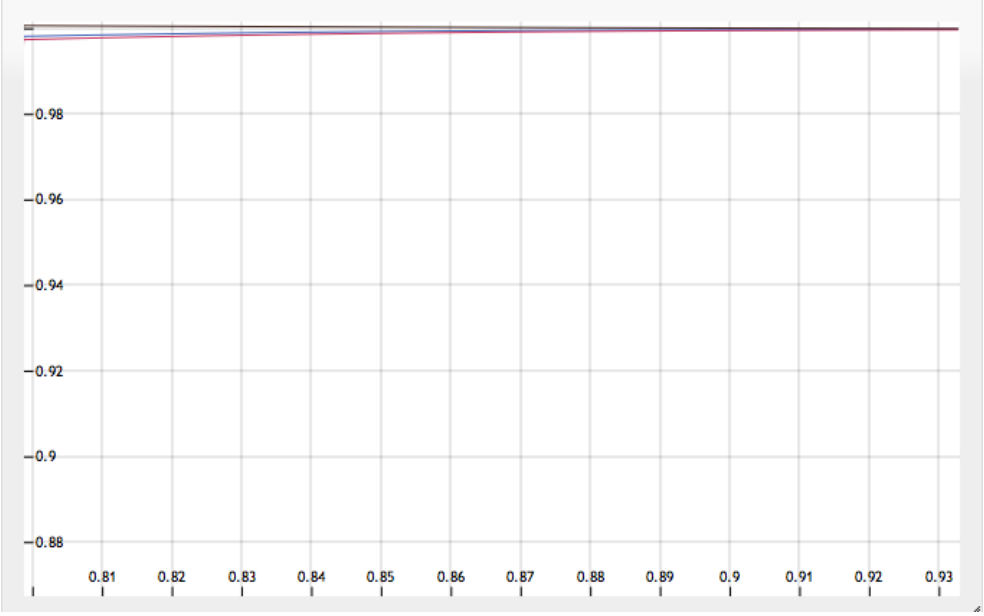
可以观察到, 其分布基本上一致.




下面是一张区间放大图. 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.




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13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.

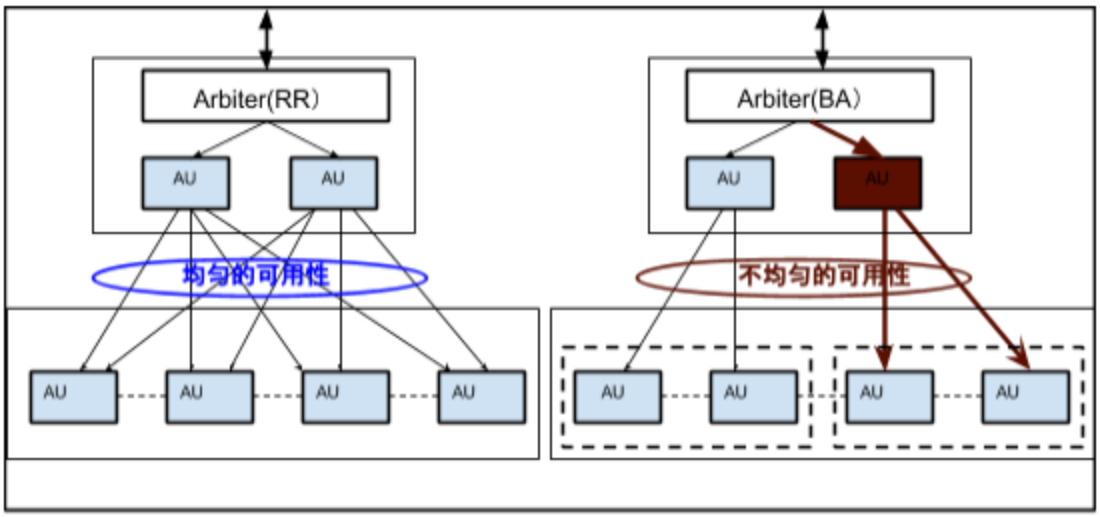



图13 扁平AS vs 复合AS

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

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:

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

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:

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

因为 $M \geq 1$, 而且 $AS_{扁平\ sub} = K/C$, 所以, $AS_{复合sub} \leq AS_{扁平\ sub}$

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:

$\mathbf{AU_{sub} = 2/C}$ **;**

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

而在复合AS的拓扑图10中, 其AU和AS的收敛比为:

$\mathbf{AU_{sub} = 1/C}$ **;**

$\mathbf{AS_{AS1\ sub} = (\sum_{i=1}^{2} AU_{i\ sub})/2 = 1/C}$ **;**

$\mathbf{AS_{AS2\ sub} = (\sum_{i=3}^{4} AU_{i\ sub})/2 = 1/C}$

因此, 复合AS的收敛比为:

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

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.

## [第二部分 参考文献]


9. Regions and Availability Zones - Amazon Elastic Compute Cloud

10.Windows Azure - Wikipedia, the free encyclopedia

11. LVS Introduction - Load Balancing Server Cluster

12. Elastic Load Balancing - Amazon Web Services