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)

(第三部分)

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

3 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14和图15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.

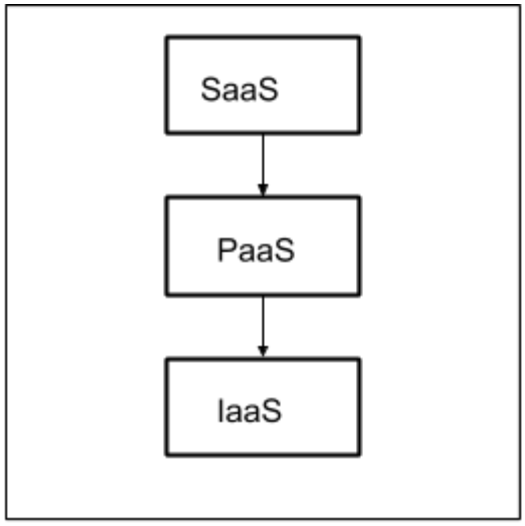

图14 参考设计模型一: 一字型结构 立体图

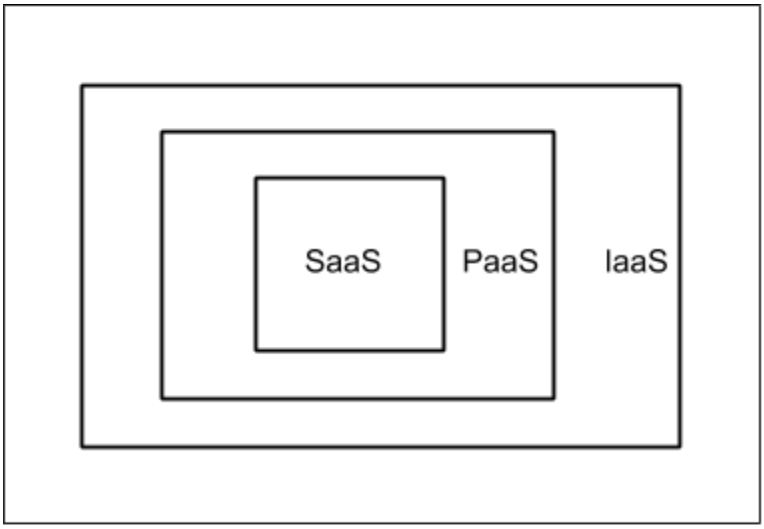

图15 参考设计模型一: 一字型结构 平面图

一字形结构的优点是简单并且成本低廉. 其缺点很明显, 存在者多个单点失效. 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: $DA_{IaaS} = SA_{IaaS}$ (我们假设数据中心的SA设计可用性已经考虑了存储,网络和服务器的失败率.)

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: $DA_{SaaS} = SA_{SaaS} * SA_{PaaS} * SA_{IaaS}$

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演变为含有两个AU的扁平AS. 如图16和图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.

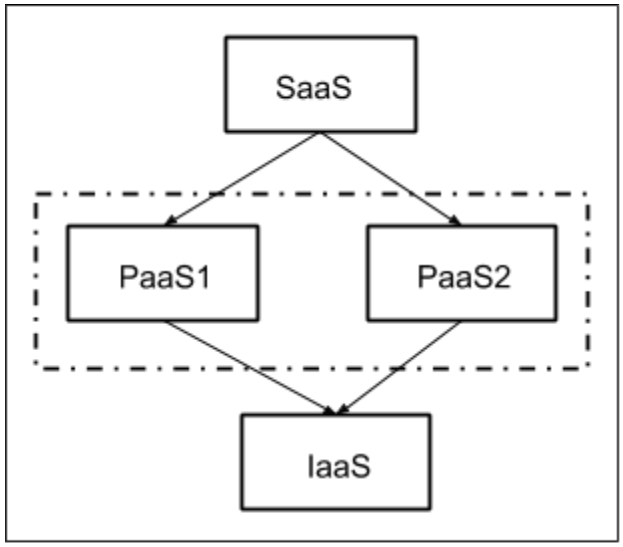

图16 参考设计模型二: 菱形结构 立体图

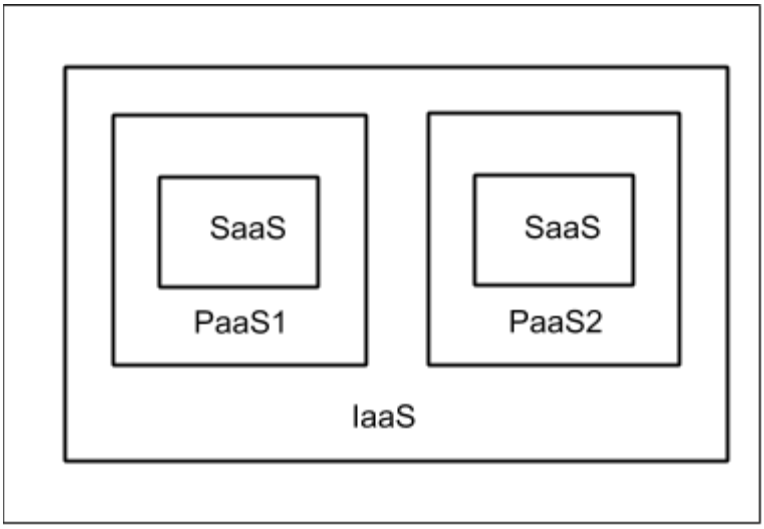

图17 参考设计模型二: 菱形结构 平面图

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层的布署, 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: $DA_{IaaS} = SA_{IaaS}$ (我们假设数据中心的SA设计可用性已经考虑了存储,网络和服务器的失败率.)

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.

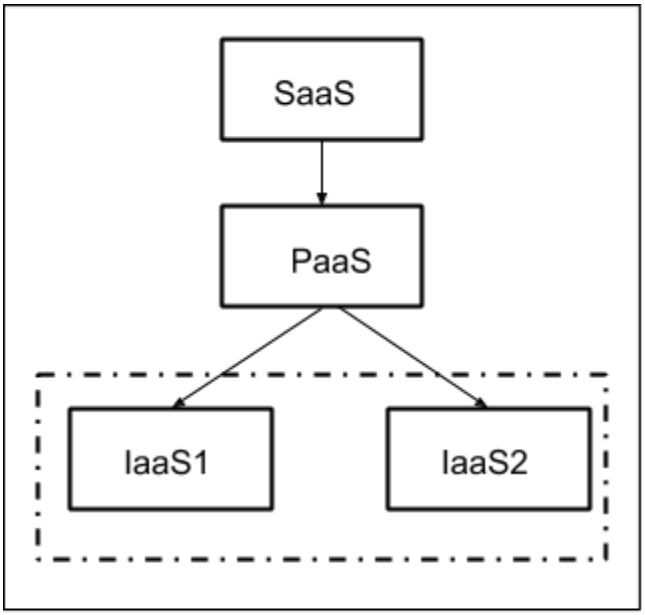

图18 参考设计模型三: 人形结构 立体图

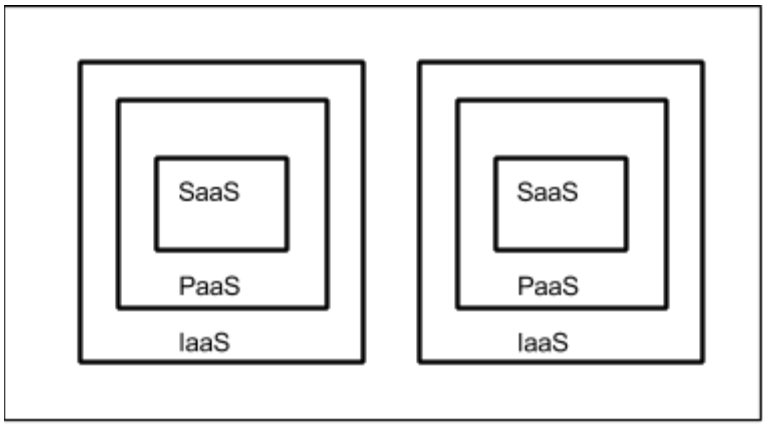

图19 参考设计模型三: 人形结构 平面图

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: $DA_{IaaS} = 1 - (1-DA_{Iass1}) * (1-DA_{Iass2}) = 1 - (1 - SA_{IaaS})^2$. (我们假设数据中心的SA设计可用性已经考虑了存储,网络和服务器的失败率.)

对于PaaS来说, 其最大部署可用性为: DA_{PaaS} =1- (1-DA_{Paas1}) * (1-DA_{Paas2}). 由于DPaaS

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18和图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.

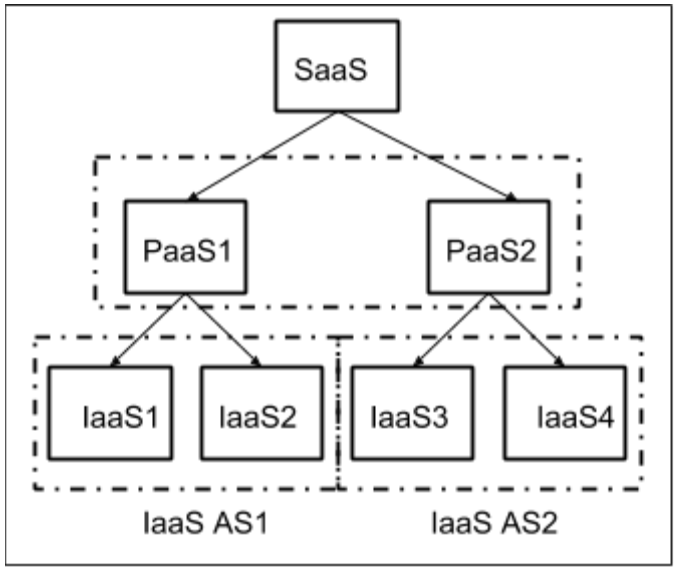

图18 参考设计模型四: 三角形结构 立体图

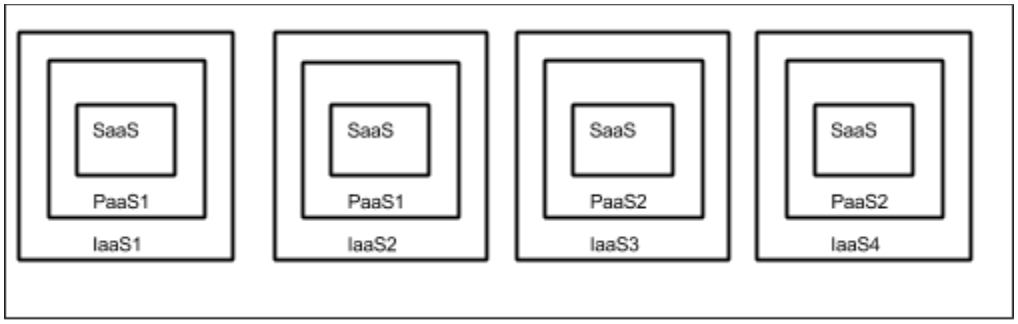

图19 参考设计模型四: 三角形结构 平面图

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: $DA_{IaaS\ AS1} = 1- (1-DA_{Iass1}) * (1-DA_{Iass2}) = 1- (1 - SA_{IaaS})^2$. $DA_{IaaS\ AS2} = 1-(1-DA_{Iass3}) * (1-DA_{Iass4}) = 1- (1 - SA_{IaaS})^2$. 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$DA_{PaaS} = 1-(1-DA_{PaaS1}) * (1-DA_{PaaS2}) = 1- [(1- SA_{PaaS} * [1-(1- SA_{IaaS})^2])^2]^2$

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

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和图21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.

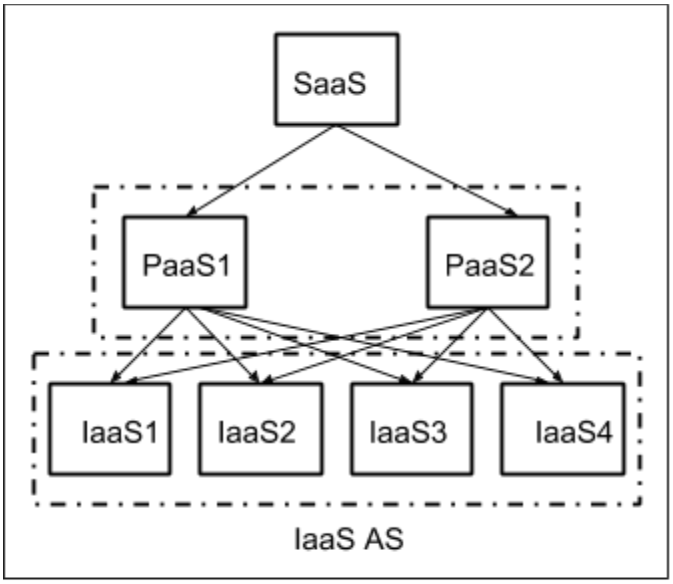

图20 参考设计模型四: 胖树结构 立体图

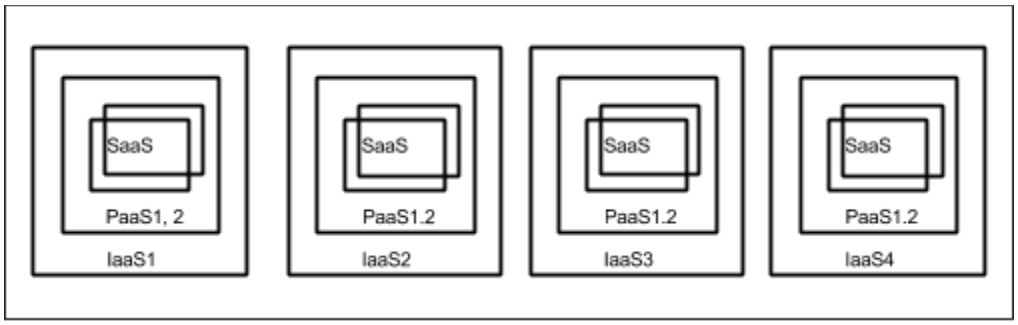

图21 参考设计模型五: 胖树结构 平面图

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.

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} =1- (1-DA_{Iass1}) * (1-DA_{Iass2}) * (1-DA_{Iass3}) * (1-DA_{Iass4}) =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.