# 一个云原生服务的爆炸半径治理

# 个人简介

**黄帅 / Henry Huang**

**亚马逊 资深技术专家**

在软件研发领域有十多年架构设计、分布式系统运维以及团队管理经验，近年来对混沌工程企业实践有深入的研究，力主推动亚马逊云科技全球混沌工程服务AWS Fault Injection Simulator (FIS) 的落地，于2021年3月成功发布。

经典混沌犹书《混沌工程：复杂系统韧性实现之道》的译者。

# 内容安排

- 一个十年前的生产事件
- 事件背后的核心问题剖析
- 十年后我们面临的新挑战
- 未来又将去向哪里

# 一个十年前的生产事件

## 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.

1. 从事件的发生到最终解决持续3天半
2. 受影响可用区中13%的存储卷卡住无法使用
3. 整个区域的存储服务API将近12小时不可用
4. 最终0.07%的存储卷无法彻底恢复

**Amazon EC2 Availability Event: April 21-22, 2011**

1:41 AM PDT We are currently investigating latency and error rates with EBS volumes and connectivity issues reaching EC2 instances in the US-EAST-1 region.
2:18 AM PDT We can confirm connectivity errors impacting EC2 instances and increased latencies impacting EBS volumes in multiple availability zones in the US-EAST-1 region. Increased error rates are affecting EBS CreateVolume API calls. We continue to work towards resolution.
2:49 AM PDT We are continuing to see connectivity errors impacting EC2 instances, increased latencies impacting EBS volumes in multiple availability zones in the US-EAST-1 region, and increased error rates affecting EBS CreateVolume API calls. We are also experiencing delayed launches for EBS backed EC2 instances in affected availability zones in the US-EAST-1 region. We continue to work towards resolution.
3:20 AM PDT Delayed EC2 instance launches and EBS API error rates are recovering. We're continuing to work towards full resolution.
4:09 AM PDT EBS volume latency and API errors have recovered in one of the two impacted Availability Zones in US-EAST-1. We are continuing to work to resolve the issues in the second impacted Availability Zone. The errors, which started at 12:55AM PDT, began recovering at 2:55am PDT.
5:02 AM PDT Latency has recovered for a portion of the impacted EBS volumes. We are continuing to work to resolve the remaining issues with EBS volume latency and error rates in a single Availability Zone.
6:09 AM PDT EBS API errors and volume latencies in the affected availability zone remain. We are continuing to work towards resolution.
6:59 AM PDT There has been a moderate increase in error rates for CreateVolume. This may impact the launch of new EBS-backed EC2 instances in multiple availability zones in the US-EAST-1 region. Launches of instance store AMIs are currently unaffected. We are continuing to work on resolving this issue.
7:40 AM PDT In addition to the EBS volume latencies, EBS-backed instances in the US-EAST-1 region are failing at a high rate. This is due to a high error rate for creating new volumes in this region.
8:54 AM PDT We'd like to provide additional color on what were working on right now (please note that we always know more and understand issues better after we fully recover and dive deep into the post mortem). A networking event early this morning triggered a large amount of re-mirroring of EBS volumes in US-EAST-1. This re-mirroring created a shortage of capacity in one of the US-EAST-1 Availability Zones, which impacted new EBS volume creation as well as the pace with which we could re-mirror and recover affected EBS volumes. Additionally, one of our internal control planes for EBS has become inundated such that it's difficult to create new EBS volumes and EBS backed instances. We are working as quickly as possible to add capacity to that one Availability Zone to speed up the re-mirroring, and working to restore the control plane issue. We're starting to see progress on these efforts, but are not there yet. We will continue to provide updates when we have them.
10:26 AM PDT We have made significant progress in stabilizing the affected EBS control plane service. EC2 API calls that do not involve EBS resources in the affected Availability Zone are now seeing significantly reduced failures and latency and are continuing to recover. We have also brought additional capacity online in the affected Availability Zone and stuck EBS volumes (those that were being remirrored) are beginning to recover. We cannot yet estimate when these volumes will be completely recovered, but we will provide an estimate as soon as we have sufficient data to estimate the recovery. We have all available resources working to restore full service functionality as soon as possible. We will continue to provide updates when we have them.
11:09 AM PDT A number of people have asked us for an ETA on when we'll be fully recovered. We deeply understand why this is important and promise to share this information as soon as we have an estimate that we believe is close to accurate. Our high-level ballpark right now is that the ETA is a few hours. We can assure you that all-hands are on deck to recover as quickly as possible. We will update the community as we have more information.
12:30 PM PDT We have observed successful new launches of EBS backed instances for the past 15 minutes in all but one of the availability zones in the US-EAST-1 Region. The team is continuing to work to recover the unavailable EBS volumes as quickly as possible.
1:48 PM PDT A single Availability Zone in the US-EAST-1 Region continues to experience problems launching EBS backed instances or creating volumes. All other Availability Zones are operating normally. Customers with snapshots of their affected volumes can re-launch their volumes and instances in another zone. We recommend customers do not target a specific Availability Zone when launching instances. We have updated our service to avoid placing any instances in the impaired zone for untargeted requests.
6:18 PM PDT Earlier today we shared our high level ETA for a full recovery. At this point, all Availability Zones except one have been functioning normally for the past 5 hours. We have stabilized the remaining Availability Zone, but recovery is taking longer than we originally expected. We have been working hard to add the capacity that will enable us to safely re-mirror the stuck volumes. We expect to incrementally recover stuck volumes over the coming hours, but believe it will likely be several more hours until a significant number of volumes fully recover and customers are able to create new EBS-backed instances in the affected Availability Zone. We will be providing more information here as soon as we have it.

Here are a couple of things that customers can do in the short term to work around these problems. Customers having problems contacting EC2 instances or with instances stuck shutting down/stopping can launch a replacement instance without targeting a specific Availability Zone. If you have EBS volumes stuck detaching/attaching and have taken snapshots, you can create new volumes from snapshots in one of the other Availability Zones. Customers with instances and/or volumes that appear to be unavailable should not try to recover them by rebooting, stopping, or detaching, as these actions will not currently work on resources in the affected zone.
10:58 PM PDT Just a short note to let you know that the team continues to be all-hands on deck trying to add capacity to the affected Availability Zone to re-mirror stuck volumes. It's taking us longer than we anticipated to add capacity to this fleet. When we have an updated ETA or meaningful new update, we will make sure to post it here. But, we can assure you that the team is working this hard and will do so as long as it takes to get this resolved.

# 一个十年前的生产事件

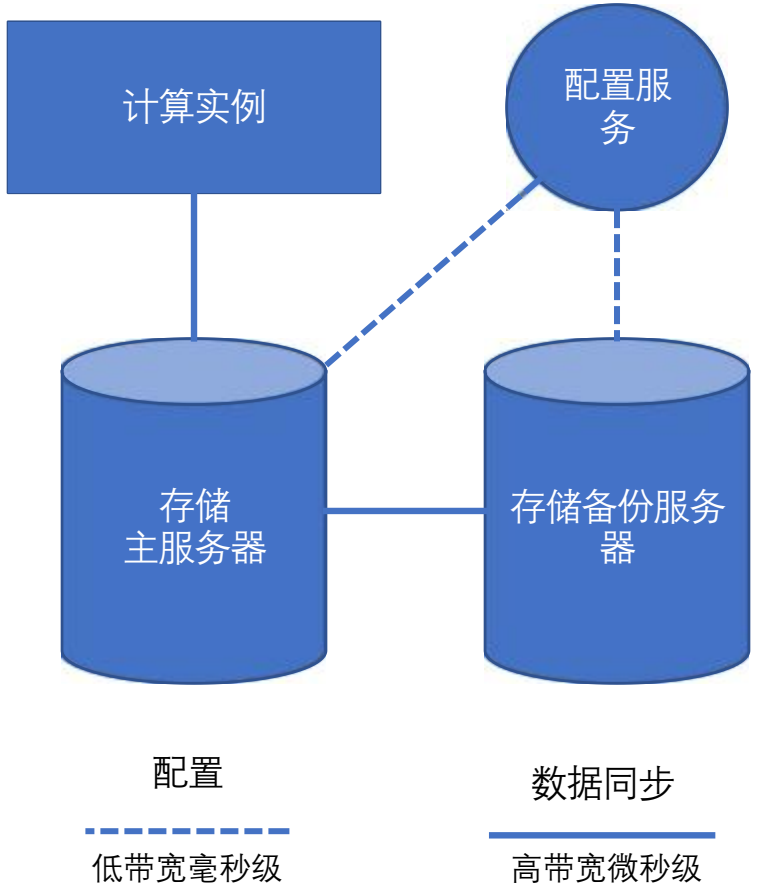
## Summary of the Amazon EC2 and Amazon RDS Service Disruption in the US East Region

**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.

1. 从事件的发生到最终解决持续3天半
2. 受影响可用区中13%的存储卷卡住无法使用
3. 整个区域的存储服务API将近12小时不可用
4. 最终0.07%的存储卷无法彻底恢复




存储服务简化模型

# 一个十年前的生产事件

4月21日
凌晨00:47

**1 升级主网络容量**
正常对一个可用区进行网络变更，将主网络的流量切到同为该网络的冗余服务器上。

**2 网络错误变更**
实际中将主网络的流量切到了备用网络中低容量的服务器上。

**3 低容量服务器开始崩溃**
备用网络中服务器容量有限，无法承受主网络中的流量压力，开始不响应，失去联系。

**4 主备网络失去连接**
受此影响的大量存储服务主节点无法联络到副本，副本复制服务受到严重影响。

**5 网络错误变更回滚**
主网络流量重新切回到原先的服务器上，存储服务的主节点开始搜寻合适的存储空间，重新镜像数据。

**6 重镜像风暴** 由于大量存储服务的主节点 受网络变更错误的影响，同 时并行搜寻存储空间重新镜 像数据，导致存储空间很快 耗尽，大量节点卡在搜寻中。

**7 控制平面无法提供服务**
核心的存储服务API无法正常响应和提供服务，大量API请求堆积，API响应的差错率和延迟都大幅度增加。

**8 禁用创建存储卷API**
API响应的差错率和延迟都开始大幅度减小，恢复到正常的状态。

# 一个十年前的生产事件

**控制平面再次故障** 重镜像引发竞争条件，增加了存储服务控制平面的协商工作量，导致控制平面再次故障，API响应的差错率和延迟又开始增加。

9

**禁止控制平面部分通信** 禁止受影响集群和控制平面的通信，阻止了受影响可用区中的API访问，差错率和延迟开始恢复正常。

10

**阻止降级服务器的搜寻** 开发了一种方法来防止降级的服务器徒劳地搜寻可用空间，集群停止进一步降级，不再有“卡住”的风险。

11

**添加新容量加速重镜像** 物理迁移多余的服务器容量，并将该容量安装到降级的集群中，通过添加大量新容量并处理重镜像积压的复制工作。

12

**控制平面API访问恢复** 构建一个单独的控制平面实例，可以在处理复制积压的同时将其分区到受影响的可用区以避免影响该区域中的其他可用区。

13

**受影响存储卷手动恢复** 受影响卷的恢复需要更多的手动过程来恢复，事件早期已经将快照备份，完成了从这些快照恢复卷的开发和测试代码，并开始批处理。

14

**0.07%存储卷无法恢复** 开始对遭受机器故障且无法为其备份快照的剩余卷进行处理。受影响的可用区中0.07%的卷，由于存在非一致状态，无法恢复。

15

**故障彻底恢复** 故障复盘和根因分析、完整信息披露以及改进措施的总结和下一步工作的推进。

16

4月24日
下午12:30

# 事件背后的核心问题剖析

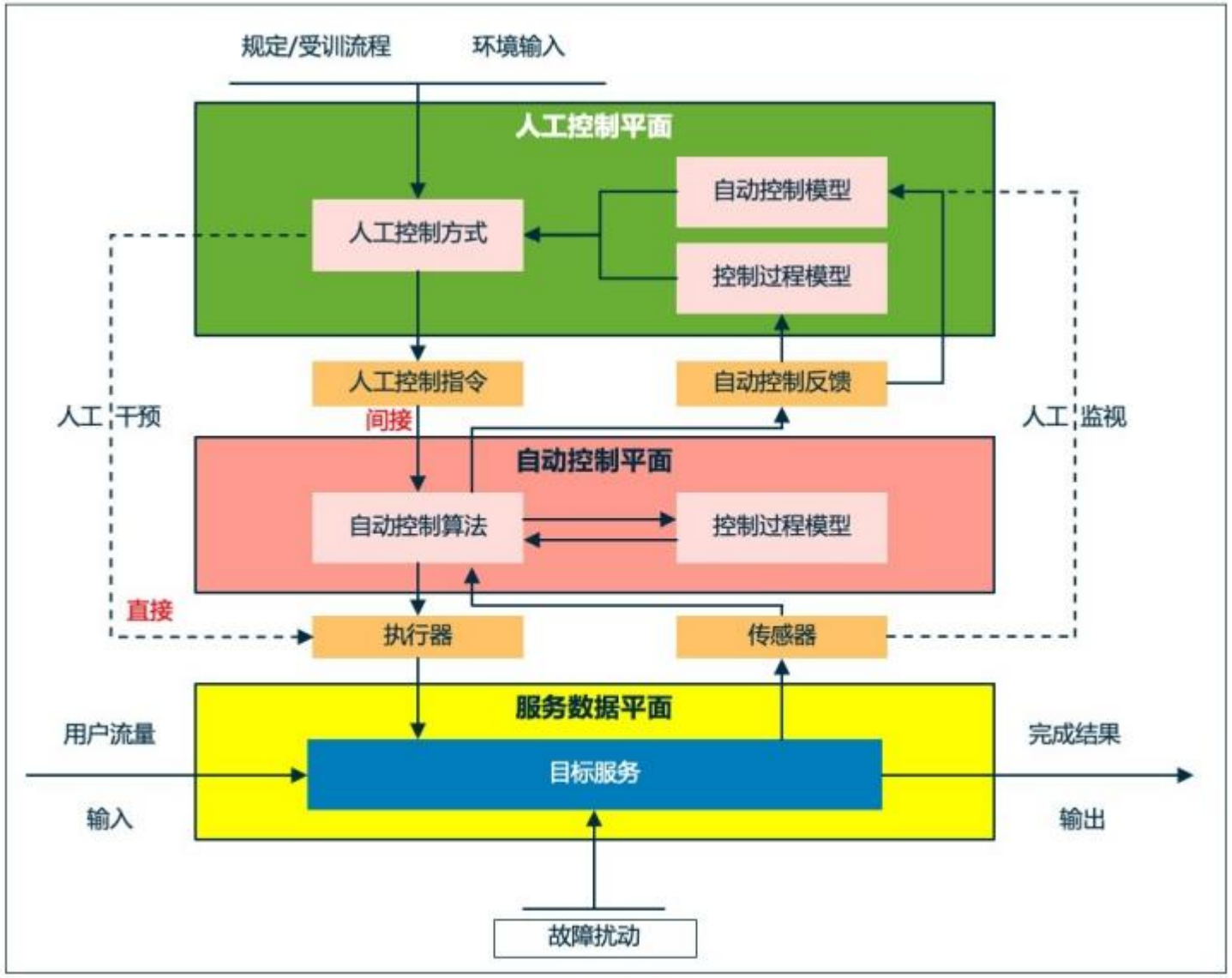

## *Engineering a Safer World*

**Systems Thinking Applied to Safety -2012**

**Professor Nancy G. Leveson -MIT**

**STPA -Systems Theoretic Process Analysis**

**STAMP -Systems Theoretic Accident Model & Processes**




STPA 分析模型

# 事件背后的核心问题剖析

4月21日
凌晨00:47

**升级主网络容量**
正常对一个可用区进行网络变更，将主网络的流量切到同为该网络的冗余服务器上。

**网络错误变更**
实际中将主网络的流量切到了备用网络中低容量的服务器上。

**低容量服务器开始崩溃**
备用网络中服务器容量有限，无法承受主网络中的流量压力，开始不响应，失去联系。

**主备网络失去连接**
受此影响的大量存储服务主节点无法联络到副本，副本复制服务受到严重影响。

1 2 3 4

**禁用创建存储卷API**
API响应的差错率和延迟都开始大幅度减小，恢复到正常的状态。

**控制平面无法提供服务**
核心的存储服务API无法正常响应和提供服务，大量API请求堆积，API响应的差错率和延迟都大幅度增加。

**重镜像风暴** 由于大量存储服务的主节点 受网络变更错误的影响，同 时并行搜寻存储空间重新镜 像数据，导致存储空间很快耗尽，大量节点卡在搜寻中。

**网络错误变更回滚** 主网络流量重新切回到原先 的服务器上，存储服务的主 节点开始搜寻合适的存储空 间，重新镜像数据。


8 7 6 5

# 事件背后的核心问题剖析

**控制平面再次故障** 重镜像引发竞争条件，增加了存储服务控制平面的协商工作量，导致控制平面再次故障，API响应的差错率和延迟又开始增加。

9

**禁止控制平面部分通信** 禁止受影响集群和控制平面的通信，阻止了受影响可用区中的API访问，差错率和延迟开始恢复正常。

10

**阻止降级服务器的搜寻** 开发了一种方法来防止降级的服务器徒劳地搜寻可用空间，集群停止进一步降级，不再有“卡住”的风险。

11

**添加新容量加速重镜像** 物理迁移多余的服务器容量，并将该容量安装到降级的集群中，通过添加大量新容量并处理重镜像积压的复制工作。

12

**控制平面API访问恢复** 构建一个单独的控制平面实例，可以在处理复制积压的同时将其分区到受影响的可用区以避免影响该区域中的其他可用区。

13

**受影响存储卷手动恢复** 受影响卷的恢复需要更多的手动过程来恢复，事件早期已经将快照备份，完成了从这些快照恢复卷的开发和测试代码，并开始批处理。

14

**0.07%存储卷无法恢复** 开始对遭受机器故障且无法为其备份快照的剩余卷进行处理。受影响的可用区中0.07%的卷，由于存在非一致状态，无法恢复。

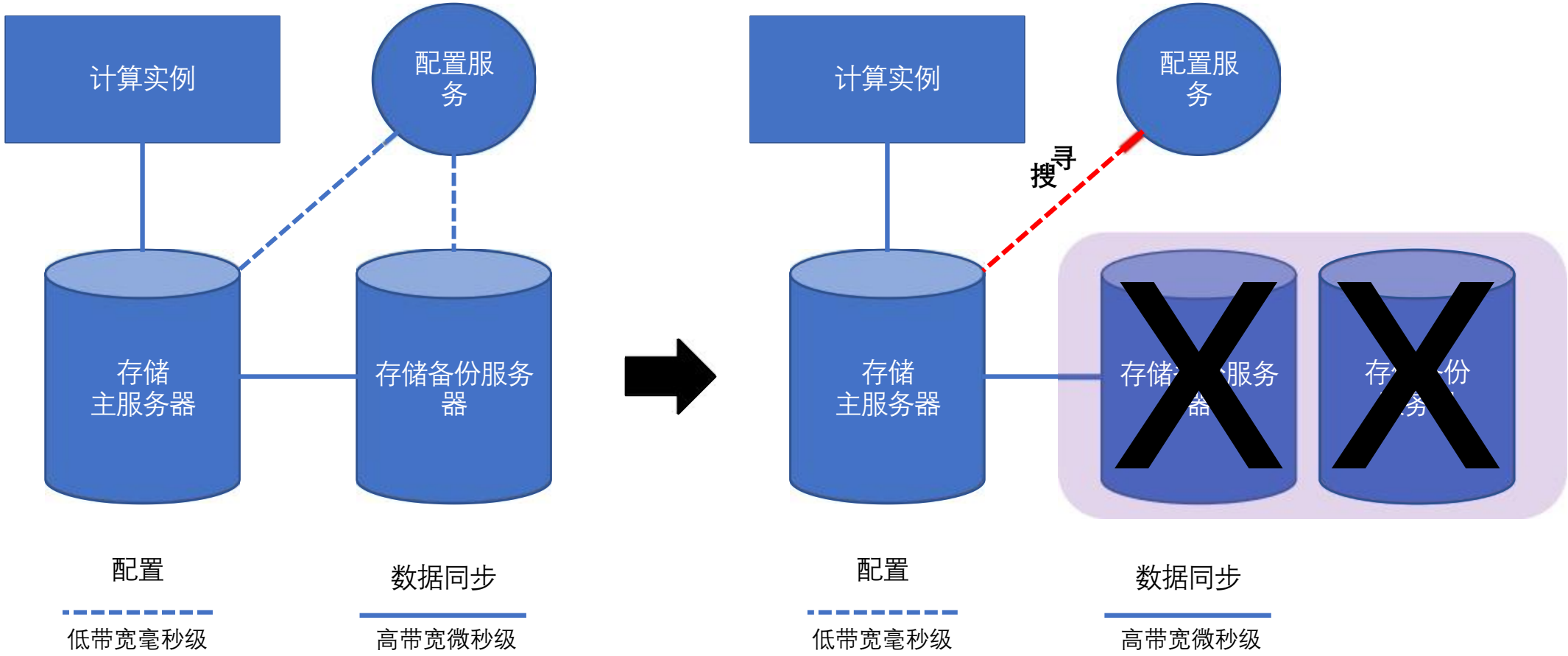
15

**故障彻底恢复** 故障复盘和根因分析、完整信息披露以及改进措施的总结和下一步工作的推进。

16

4月24日
下午12:30

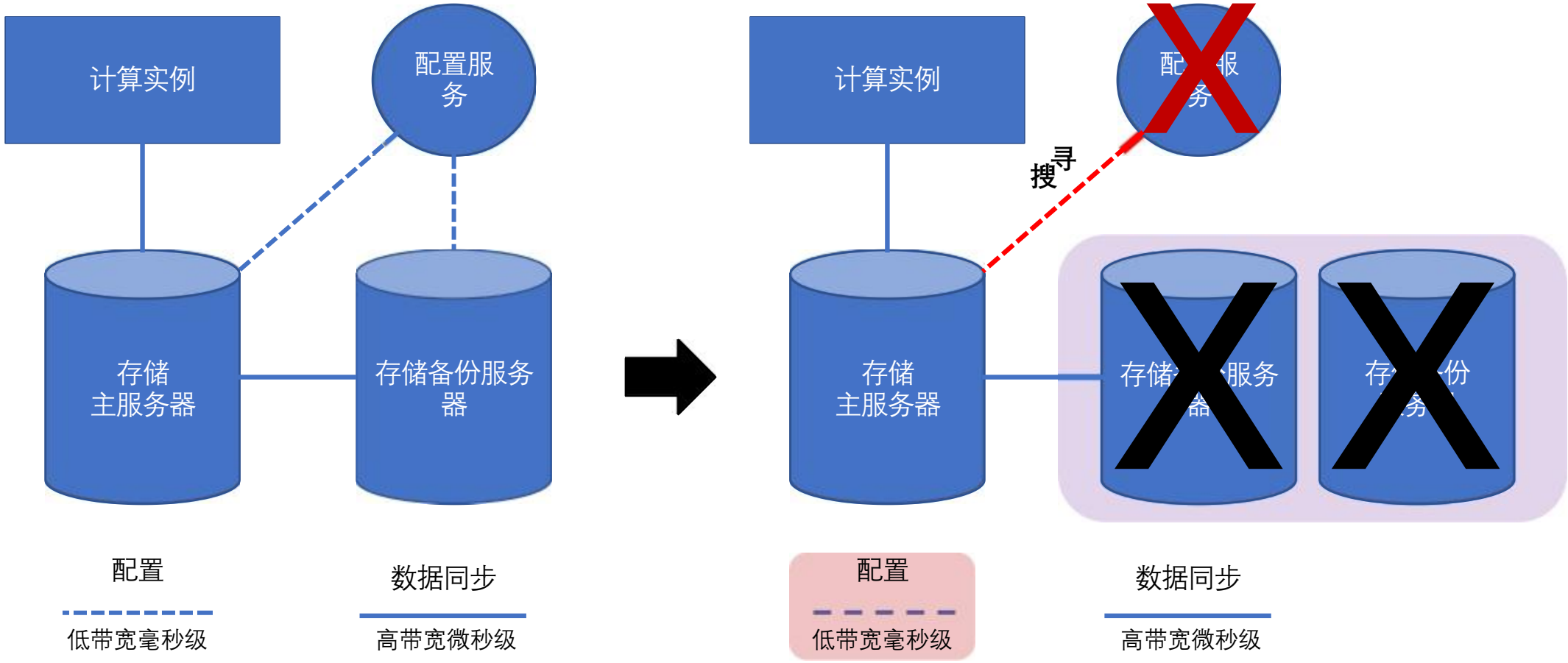
# 事件背后的核心问题剖析




存储服务简化模型

存储服务简化模型

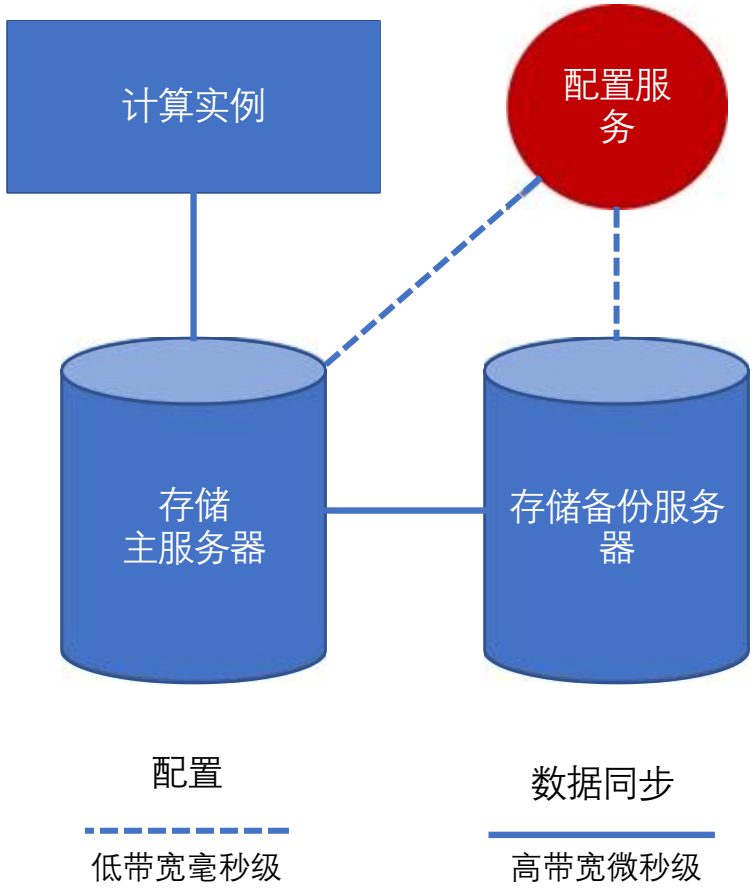
# 事件背后的核心问题剖析




存储服务简化模型

存储服务简化模型

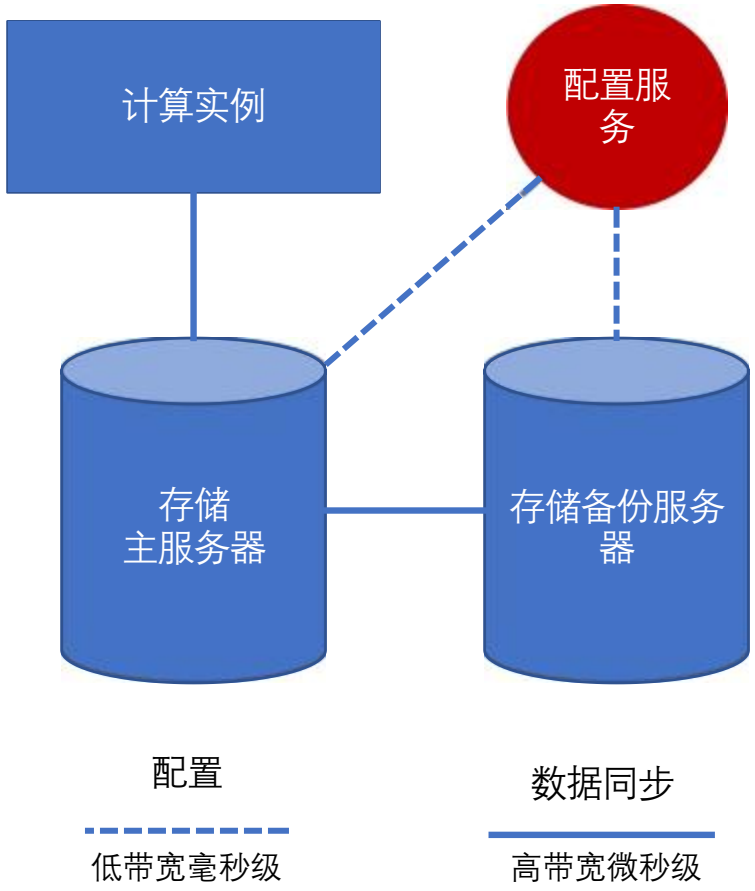
# 事件背后的核心问题剖析



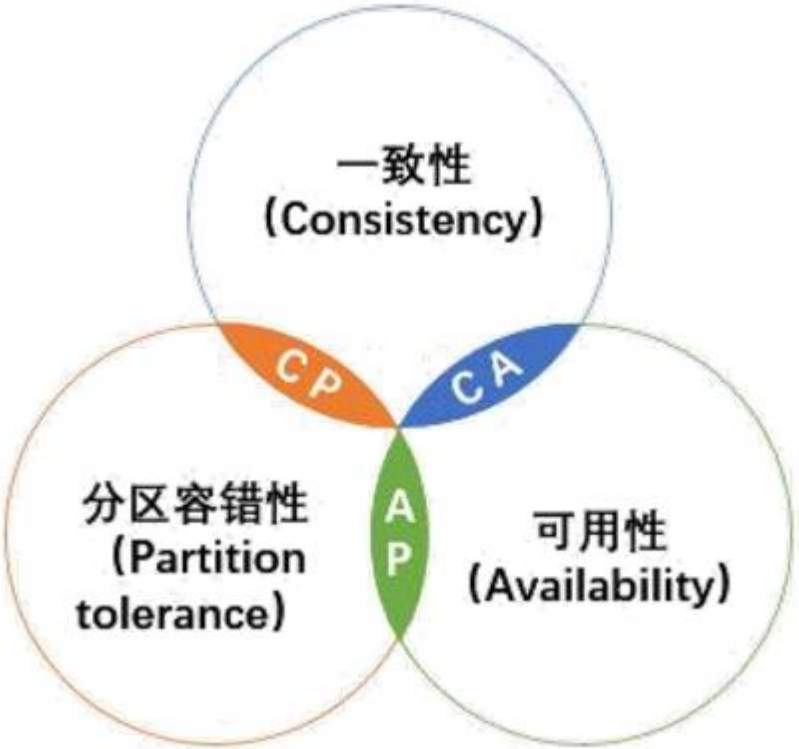

存储服务简化模型

1. 本身就是一个分布式服务
2. 背后由键值对数据库支撑
3. 要求强一致性，保障数据同步复制和正确性
4. 考虑到数据的正确性，完全无法接受最终一致性
5. 正常情况下，占用低带宽且保证毫秒级延迟
6. 存储卷IO是阻塞性，配置服务的调用未完成，就会卡在那里
7. 链式复制方法本身会需要配置更新，这就要求一定的可用性
8. 故障情况下，接收大量突发的搜寻工作，严重影响可用性
9. 发生不可用时，会逐步影响控制平面，最终毒化到API无法访问

# 事件背后的核心问题剖析

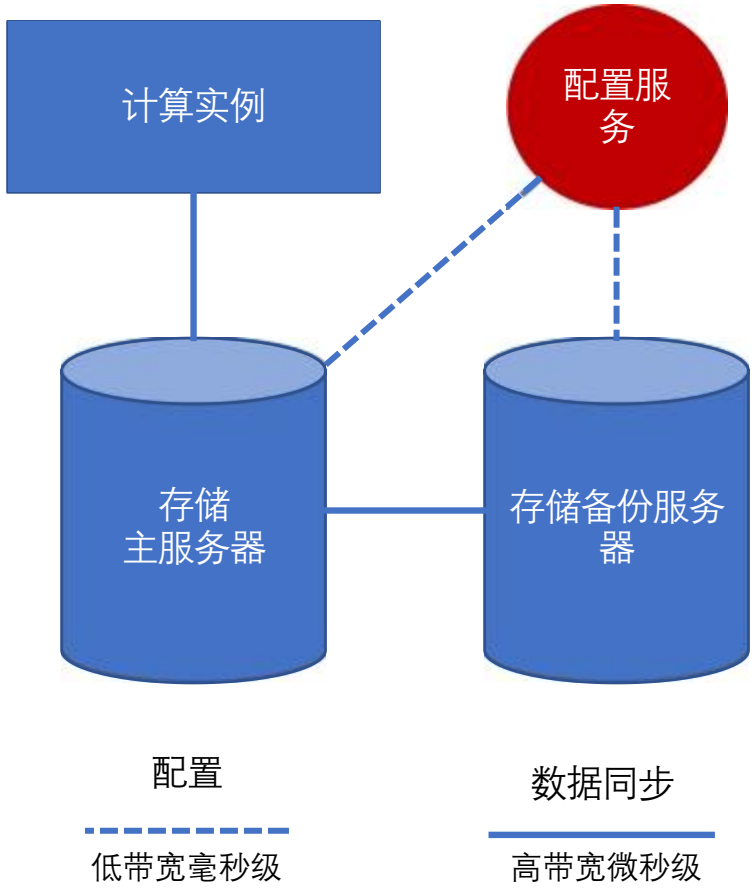



存储服务简化模型

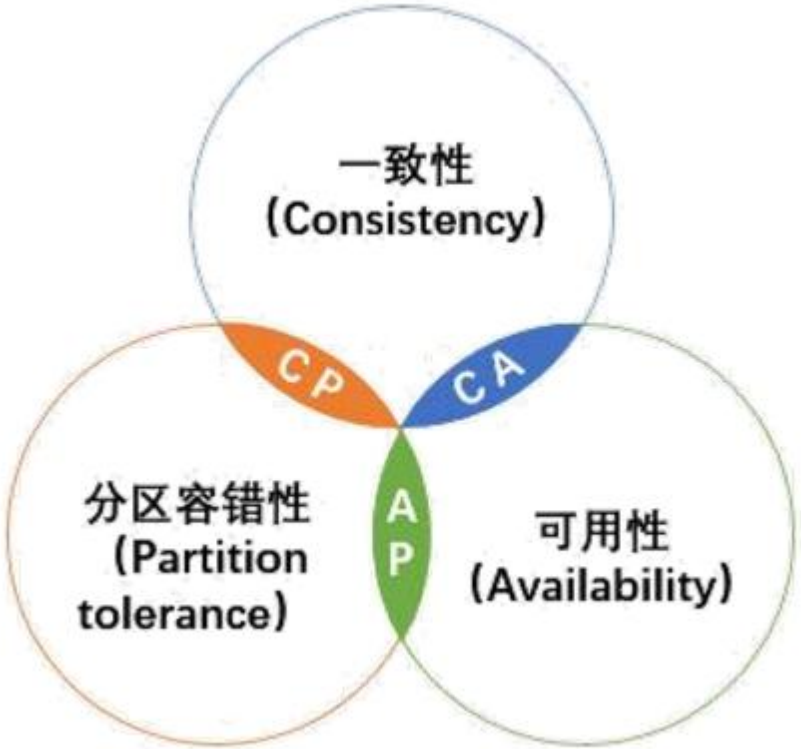



对于一个分布式计算系统来说，不可能同时满足这三方面的性能指标（一致性、可用性和分区容错性），最多只能同时满足其中的两项。

# 事件背后的核心问题剖析

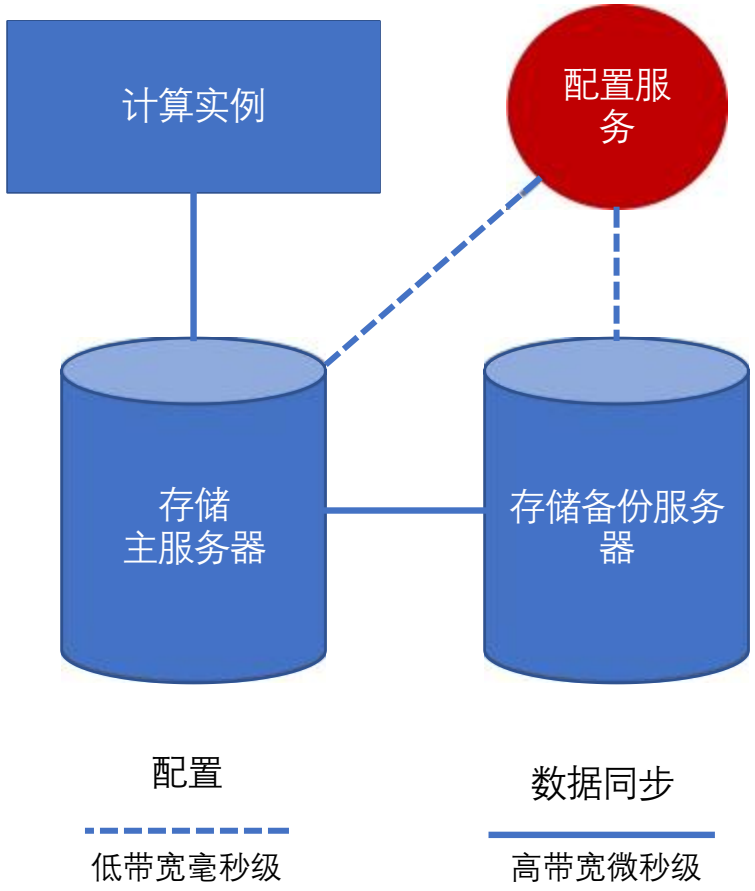



存储服务简化模型

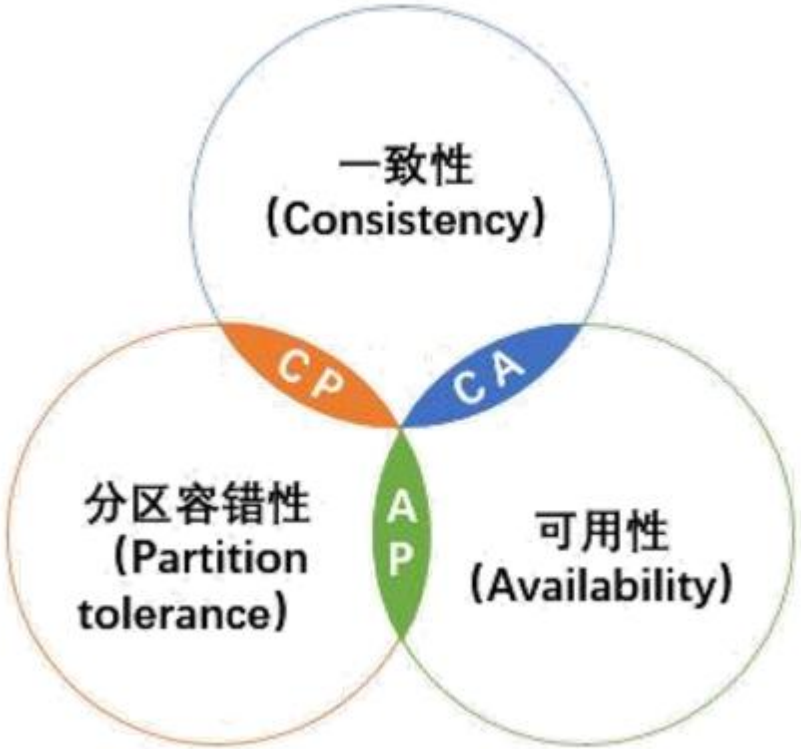



对于一个分布式计算系统来说，不可能同时满足这三方面的性能指标（一致性、可用性和分区容错性），最多只能同时满足其中的两项。

# 事件背后的核心问题剖析



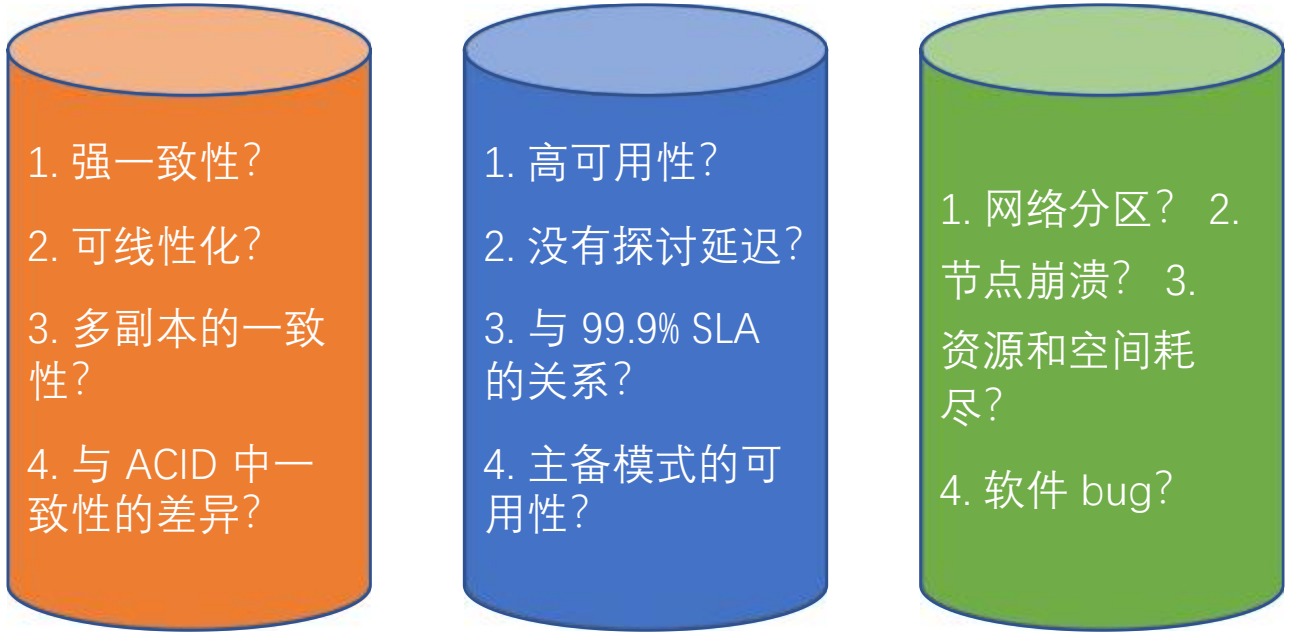

存储服务简化模型



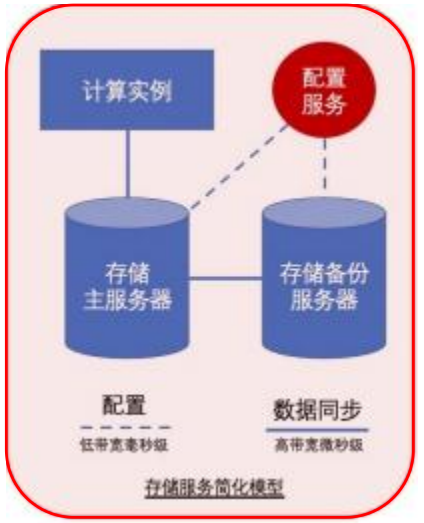

我们对配置服务的要求：**既要强一致性，又要高可用性?**

# 事件背后的核心问题剖析

- CAP 定理的常见疑惑




*一致性* **C**　　*可用性* **A**　　*分区容错性* **P**

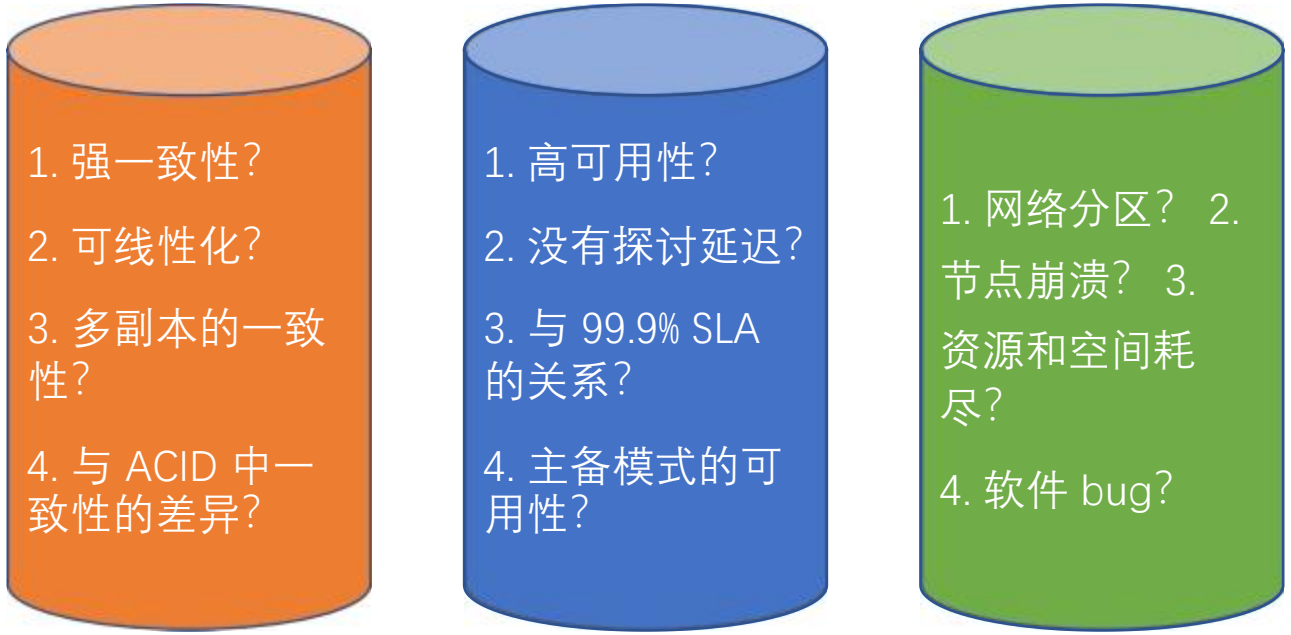



*分布式*
*多节点*
*并发性*
*扩展性*

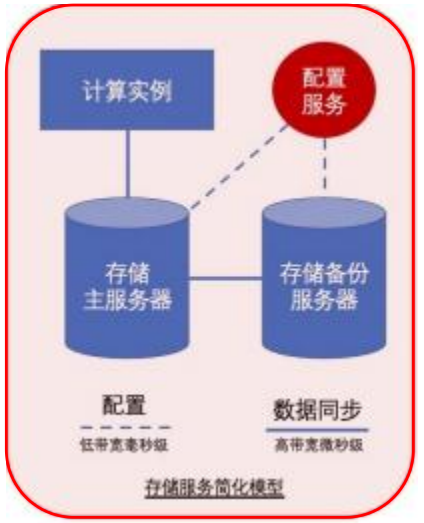

控制平面的核心部分之一

# 事件背后的核心问题剖析

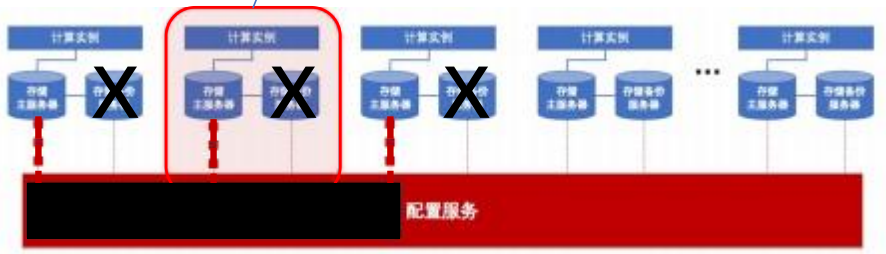
- CAP 定理的常见疑惑




*一致性* **C**　　*可用性* **A**　　*分区容错性* **P**



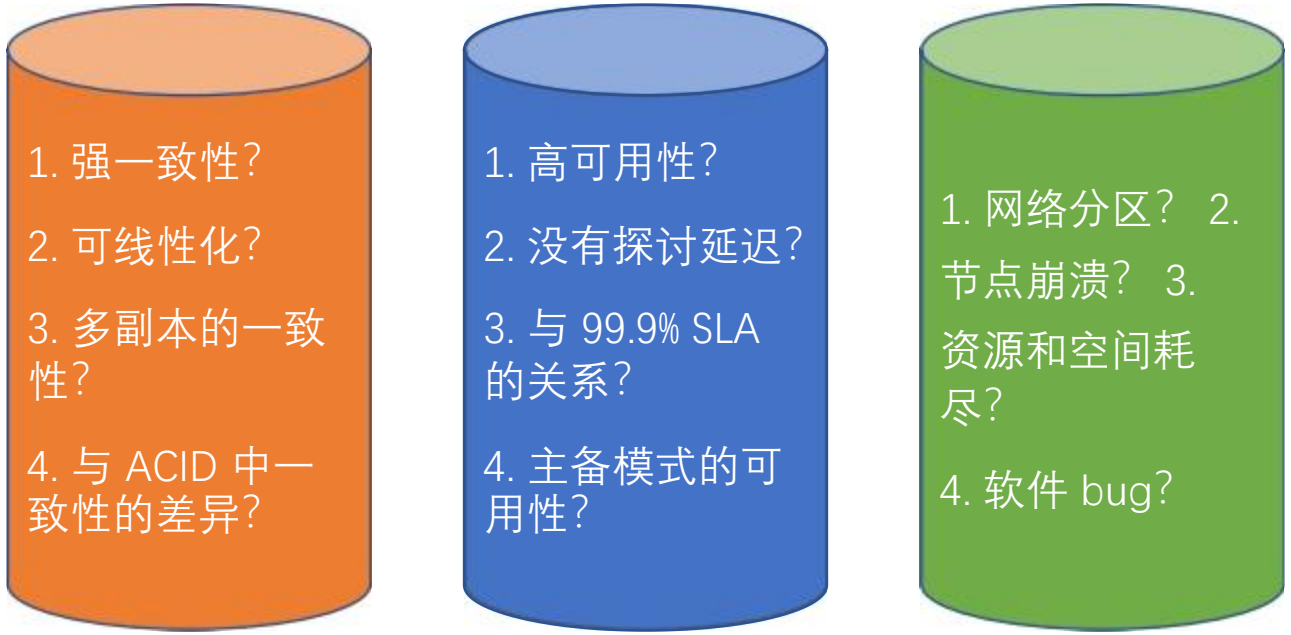

*分布式*
*多节点*
*并发性*
*扩展性*



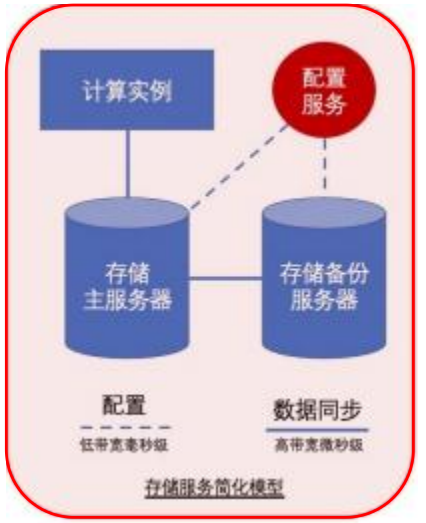

控制平面的核心部分之一

# 事件背后的核心问题剖析

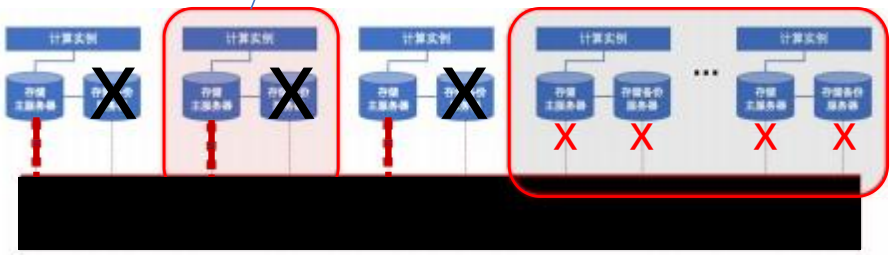
- CAP 定理的常见疑惑




*一致性* **C**　　*可用性* **A**　　*分区容错性* **P**

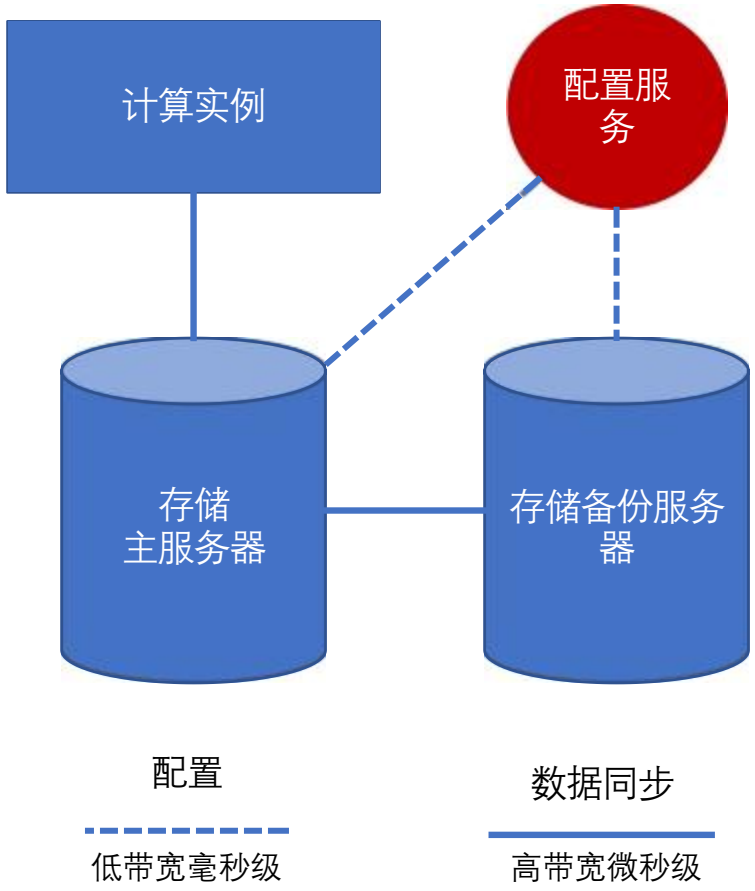



*分布式*
*多节点*
*并发性*
*扩展性*

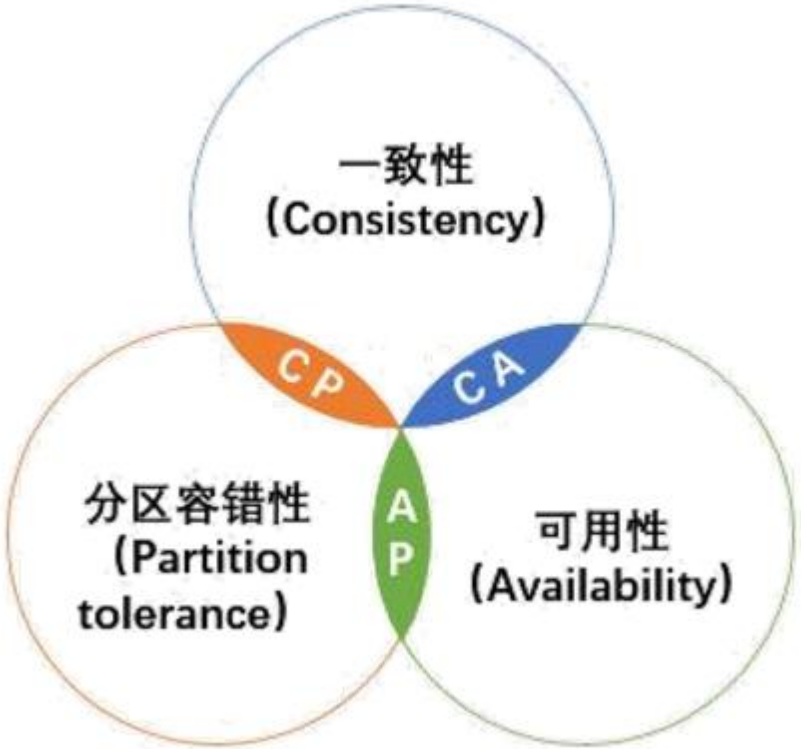


控制平面的核心部分之一

# 事件背后的核心问题剖析

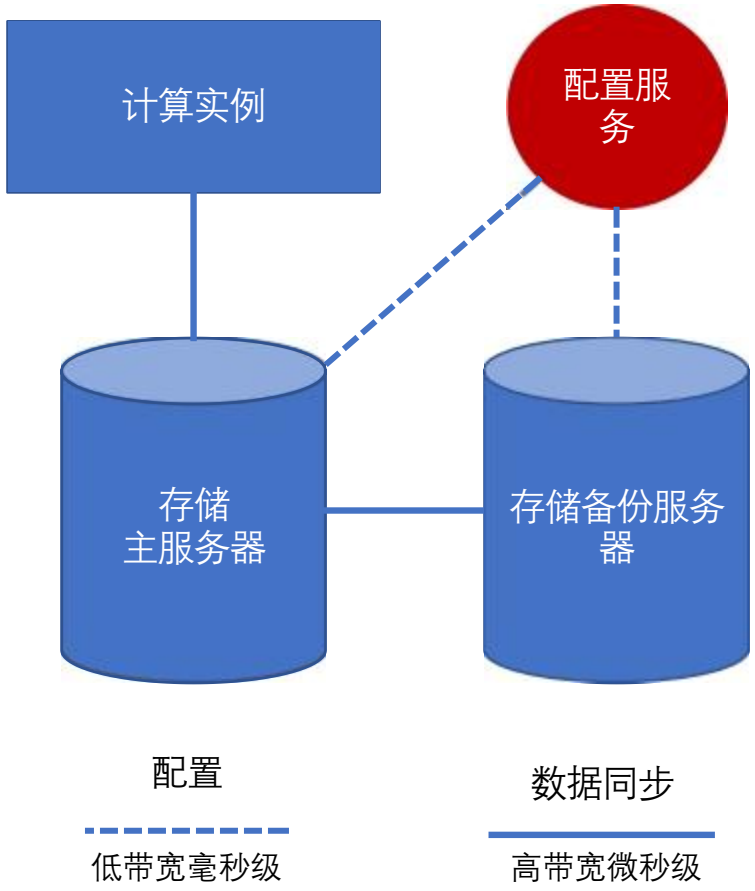



存储服务简化模型

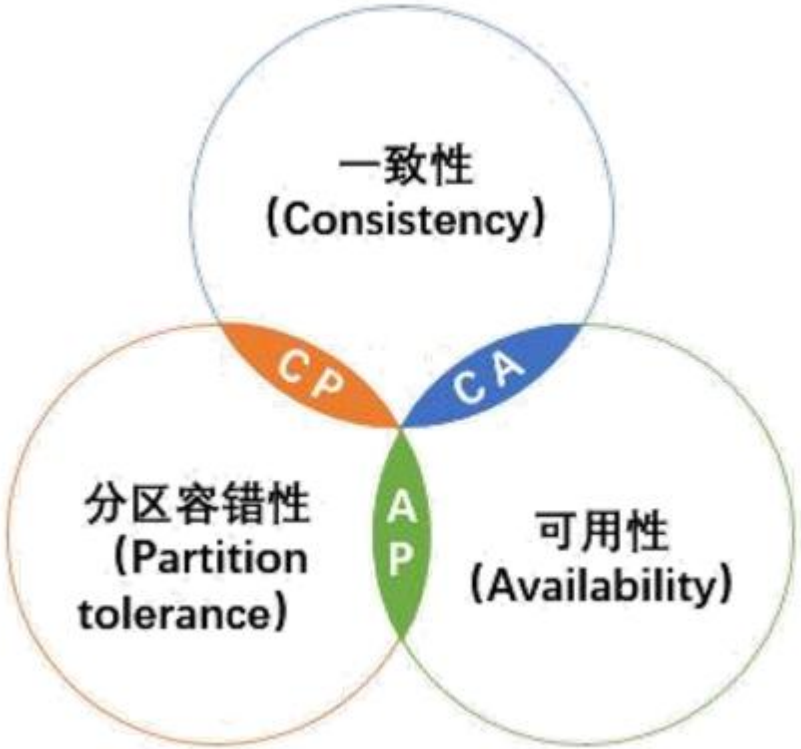



~~我们对配置服务的要求：**既要强一致性，又要高可用性？**~~

# 事件背后的核心问题剖析

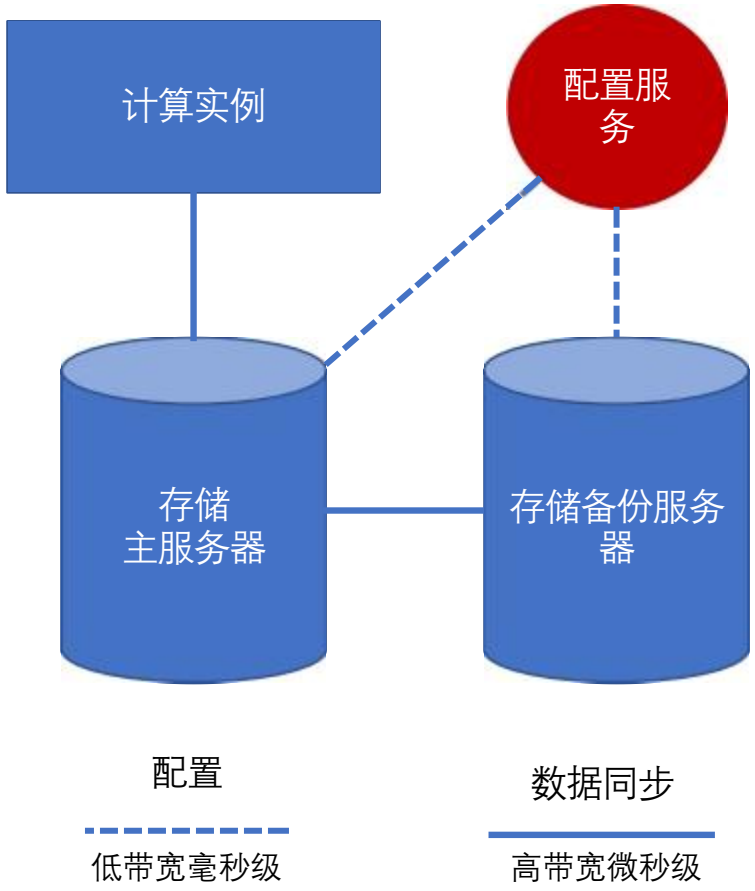



存储服务简化模型

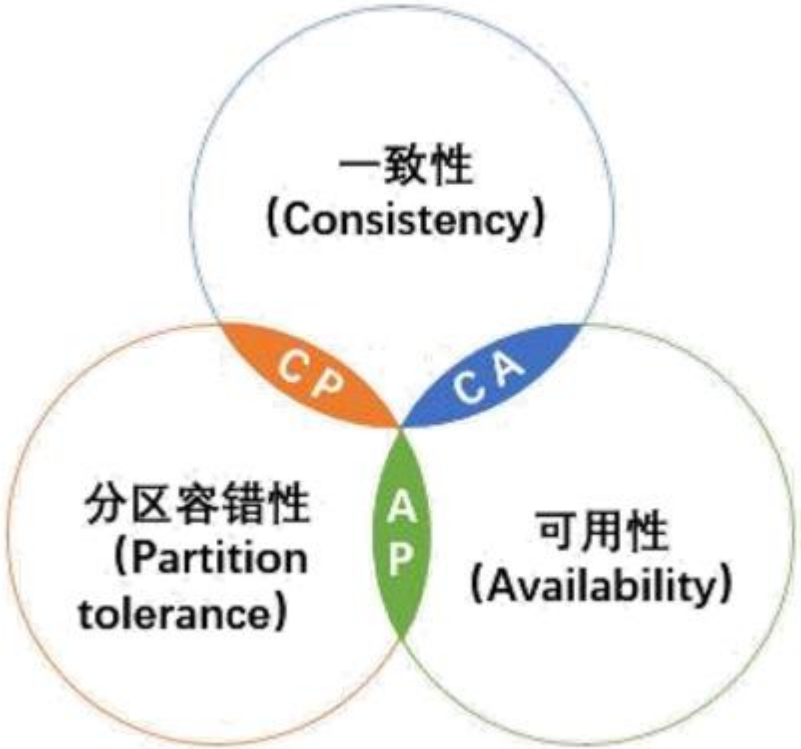



结论：不是所有的数据都需要对所有的用户可用。

# 事件背后的核心问题剖析




存储服务简化模型

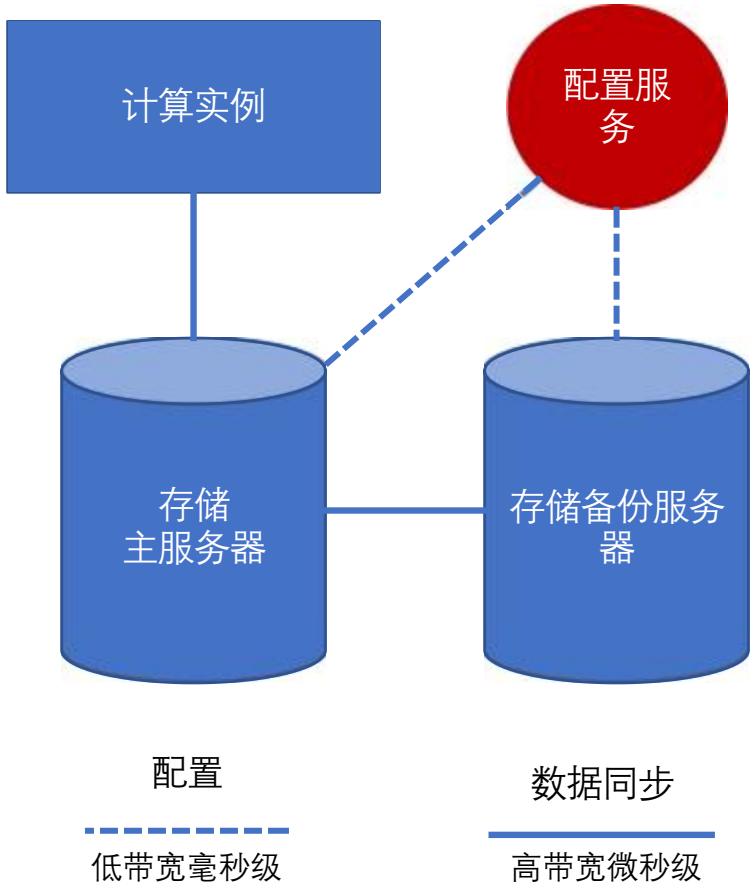



**结论：不是所有的数据都需要对所有的用户可用。**

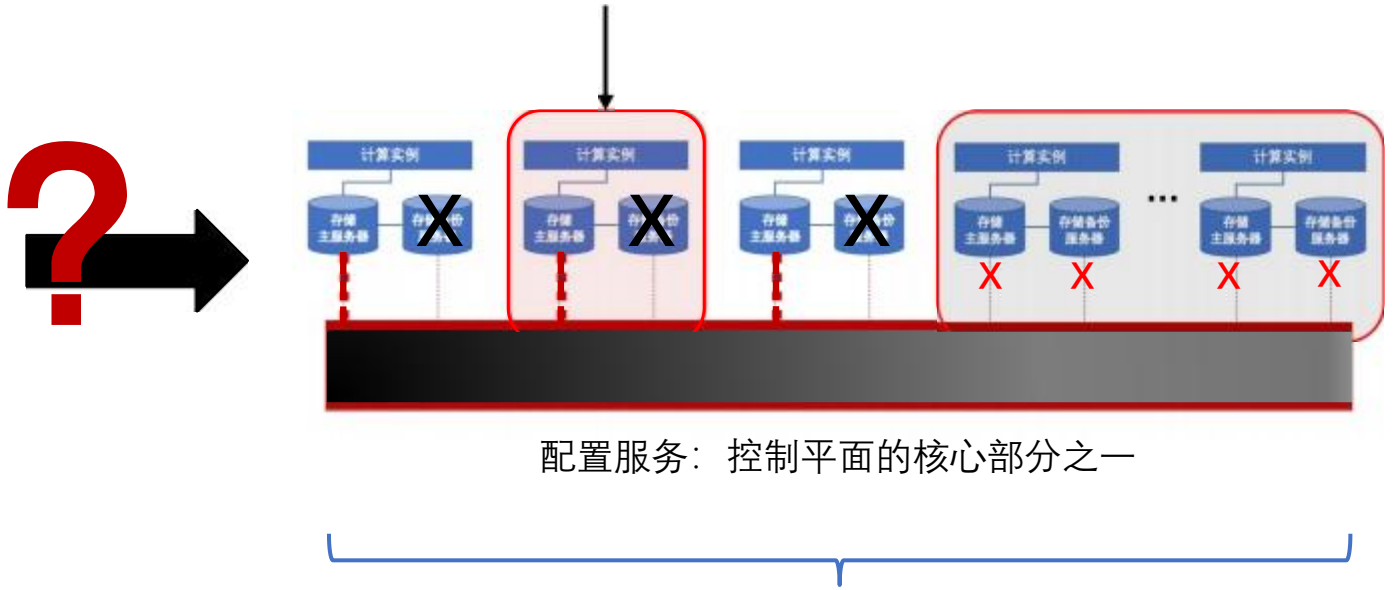
我们对配置服务的要求改成了：

**在强一致性的保证下，尽可能实现高可用性。**

# 事件背后的核心问题剖析




存储服务简化模型




配置服务：控制平面的核心部分之一

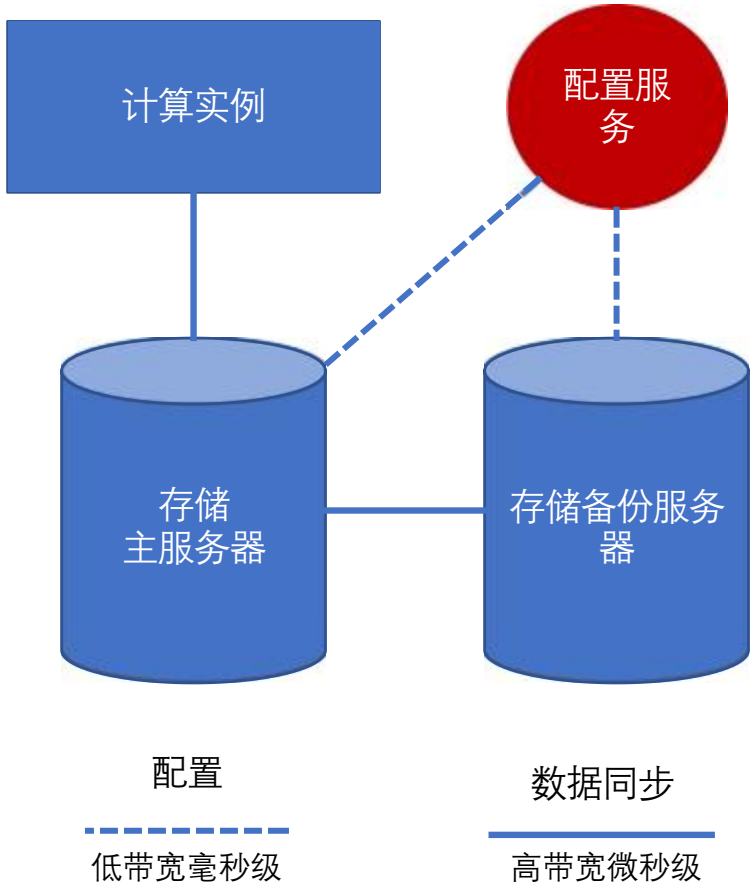
**配置服务的爆炸半径过大**

**结论：不是所有的数据都需要对所有的用户可用。**

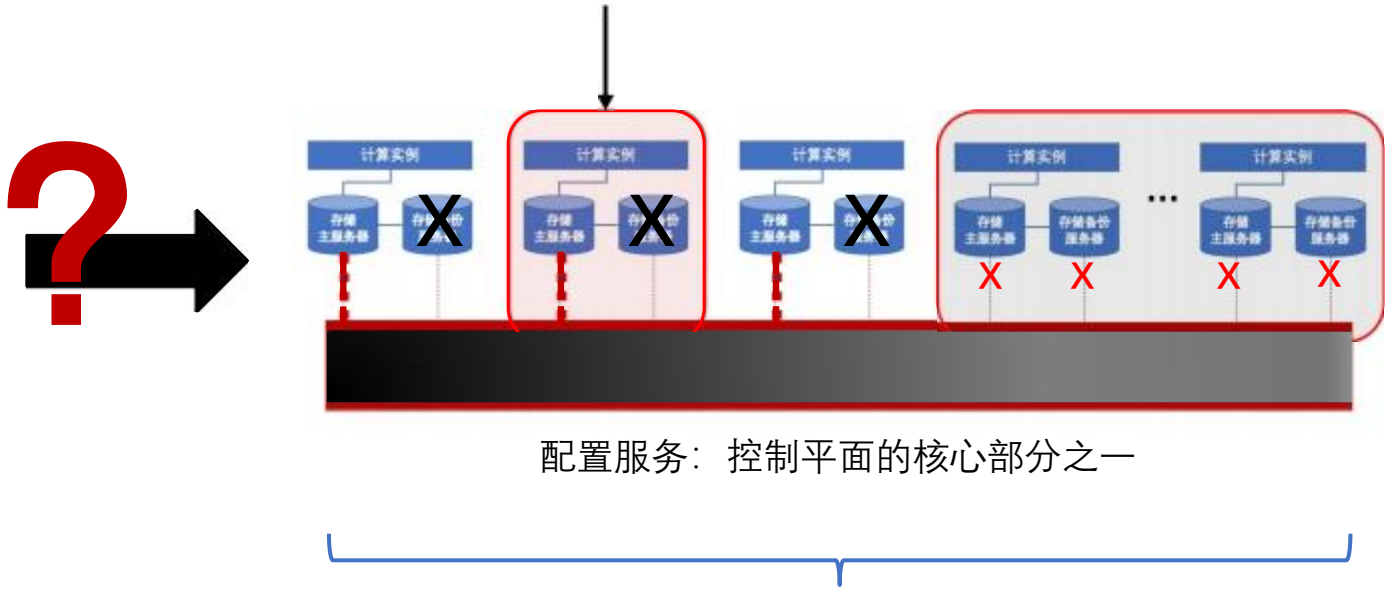
我们对配置服务的要求改成了：

**在强一致性的保证下，尽可能实现高可用性。**

# 事件背后的核心问题剖析




存储服务简化模型




配置服务：控制平面的核心部分之一

## 配置服务的爆炸半径过大

因迅速扩展引出的新问题

关联故障接连不断

**结论：不是所有的数据都需要对所有的用户可用。**

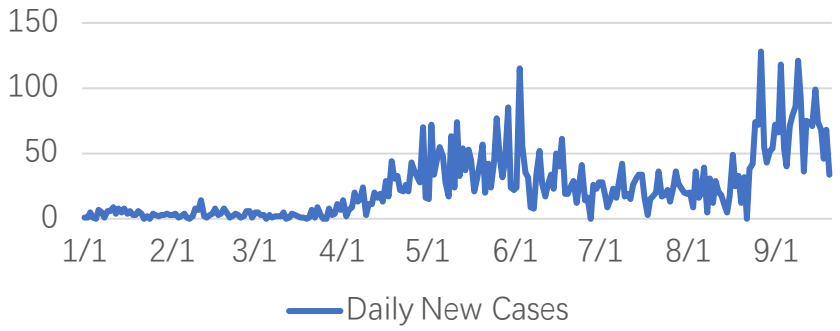
我们对配置服务的要求改成了：

**在强一致性的保证下，尽可能实现高可用性。**

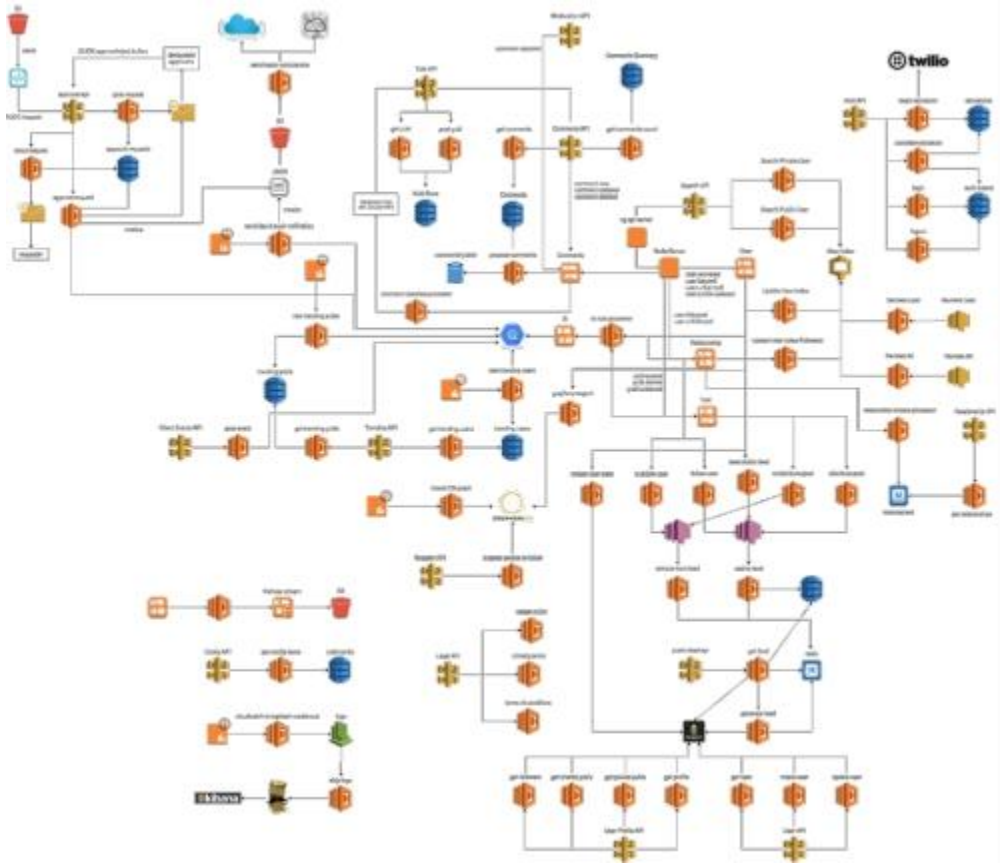
# 十年后我们面临的新挑战

## 在迅速扩展中，保持韧性的关键是什么？

- 业务快速扩充，组织增长迅猛
- 架构经多次迭代，全面实现微服务化
- 开发人员已经没有人能讲出当前整个架构设计背后的种种原因和考量，小修小补却不敢进行重构
- 用户反馈问题时，已经很难迅速定位故障点，根因分析甚至超过2周
- 运维人员不堪故障和用户问题侵扰，心理负荷特别重




**说好的微服务架构的优势都去了哪里？**




引用自 htt s://www.s ideshare. et/A az WebServices/a yi g- ri ci es- f-cha s-e gi eeri g-t -server ess-dvc305-aws-rei ve t-2018

# 十年后我们面临的新挑战

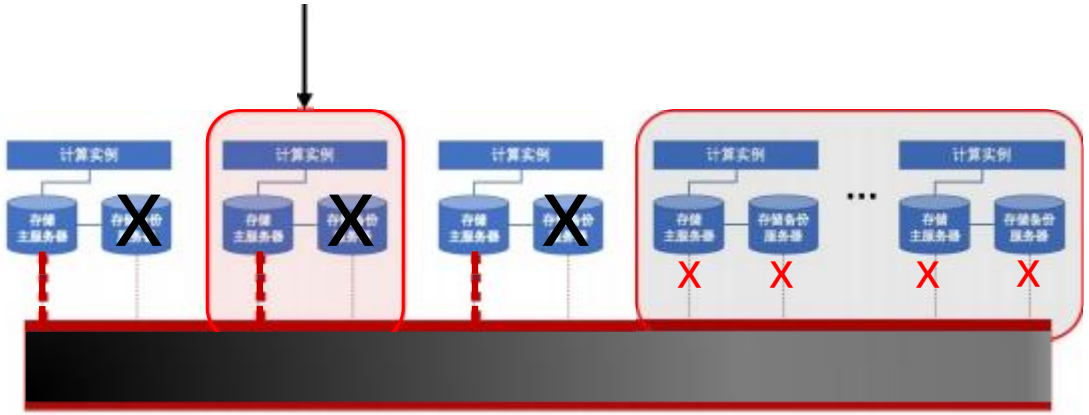
## 在迅速扩展中，保持韧性的关键是什么？

**如何减小爆炸半径？**

定义

- 哪些人会受到影响？
- 影响多少工作负载？
- 哪些功能受到影响？
- 多少个地点受影响？

亚马逊 CoE 文档模版中有一个题目：
**您将采用什么改进的方法来减少当前事件的爆炸半径？**




配置服务：控制平面的核心部分之一

**配置服务的爆炸半径过大**

因迅速扩展引出的新问题

关联故障接连不断

# 十年后我们面临的新挑战

4月21日
凌晨00:47

**人为错误引起**

**升级主网络容量**
正常对一个可用区进行网络变更，将主网络的流量切到同为该网络的冗余服务器上。

**网络错误变更**
实际中将主网络的流量切到了备用网络中低容量的服务器上。

**爆炸半径开始发酵**

**低容量服务器开始崩溃**
备用网络中服务器容量有限，无法承受主网络中的流量压力，开始不响应，失去联系。

**主备网络失去连接**
受此影响的大量存储服务主节点无法联络到副本，副本复制服务受到严重影响。

1 2 3 4

**禁用创建存储卷API**
API响应的差错率和延迟都开始大幅度减小，恢复到正常的状态。

**控制平面无法提供服务**
核心的存储服务API无法正常响应和提供服务，大量API请求堆积，API响应的差错率和延迟都大幅度增加。

**重镜像风暴** 由于大量存储服务的主节点 受网络变更错误的影响，同 时并行搜寻存储空间重新镜 像数据，导致存储空间很快耗尽，大量节点卡在搜寻中。

**网络错误变更回滚** 主网络流量重新切回到原先 的服务器上，存储服务的主 节点开始搜寻合适的存储空 间，重新镜像数据。

**人为限流-问题缓解**

**爆炸半径已经失控**

8 7 6 5

# 十年后我们面临的新挑战

关联故障-急转直下

**控制平面再次故障** 重镜像引发竞争条件，增加了存储服务控制平面的协商工作量，导致控制平面再次故障，API响应的差错率和延迟又开始增加。

9

**禁止控制平面部分通信** 禁止受影响集群和控制平面的通信，阻止了受影响可用区中的API访问，差错率和延迟开始恢复正常。

10

实施隔离、限流和打破瓶颈

**阻止降级服务器的搜寻** 开发了一种方法来防止降级的服务器徒劳地搜寻可用空间，集群停止进一步降级，不再有“卡住”的风险。

11 批量恢复和逐个清理

**添加新容量加速重镜像** 物理迁移多余的服务器容量，并将该容量安装到降级的集群中，通过添加大量新容量并处理重镜像积压的复制工作。

12

**控制平面API访问恢复** 构建一个单独的控制平面实例，可以在处理复制积压的同时将其分区到受影响的可用区以避免影响该区域中的其他可用区。

13

**受影响存储卷手动恢复** 受影响卷的恢复需要更多的手动过程来恢复，事件早期已经将快照备份，完成了从这些快照恢复卷的开发和测试代码，并开始批处理。

14

**0.07%存储卷无法恢复** 开始对遭受机器故障且无法为其备份快照的剩余卷进行处理。受影响的可用区中0.07%的卷，由于存在非一致状态，无法恢复。

15

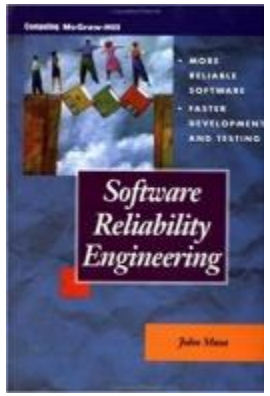
**故障彻底恢复** 故障复盘和根因分析、完整信息披露以及改进措施的总结和下一步工作的推进。

16

4月24日
下午12:30

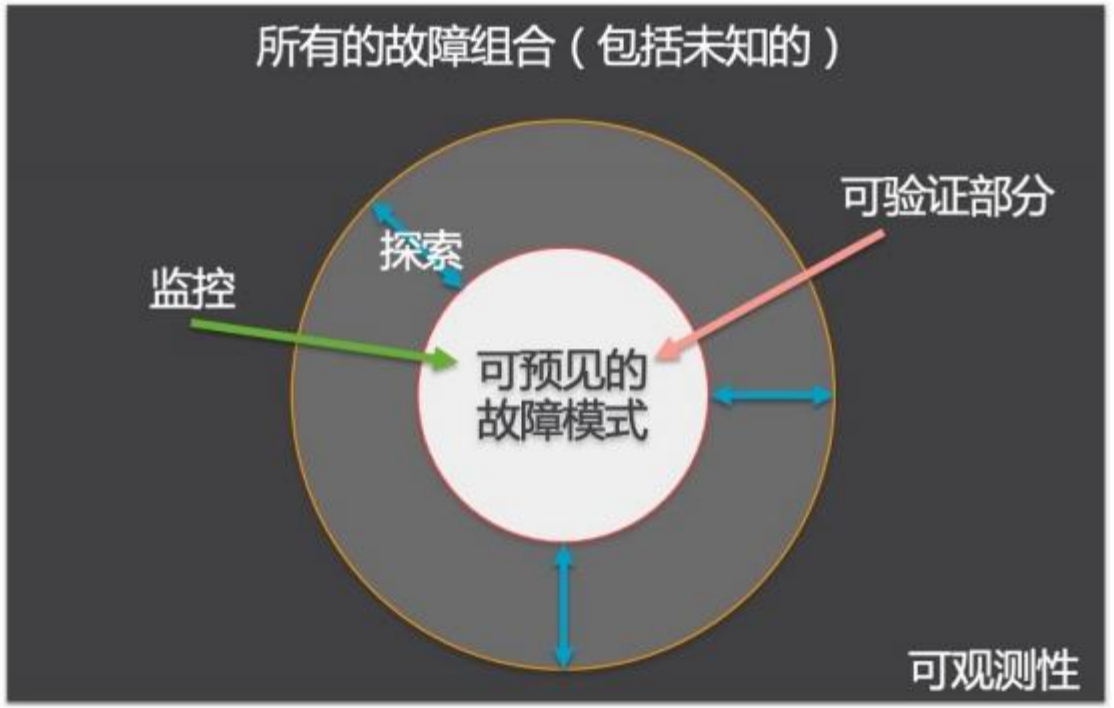
# 十年后我们面临的新挑战

**故障宿命论**：人为错误/网络异常/资源耗尽/软件bug等都不可避免


- 负载尖峰和突发过载
- 邻近的故障转移
- 崩溃查询
- 重试风暴
- 资源耗尽
- 资源限制

- CPU问题
- 内存问题
- 线程枯竭
- 发布和变更
- 启动时间过长
- 依赖失效

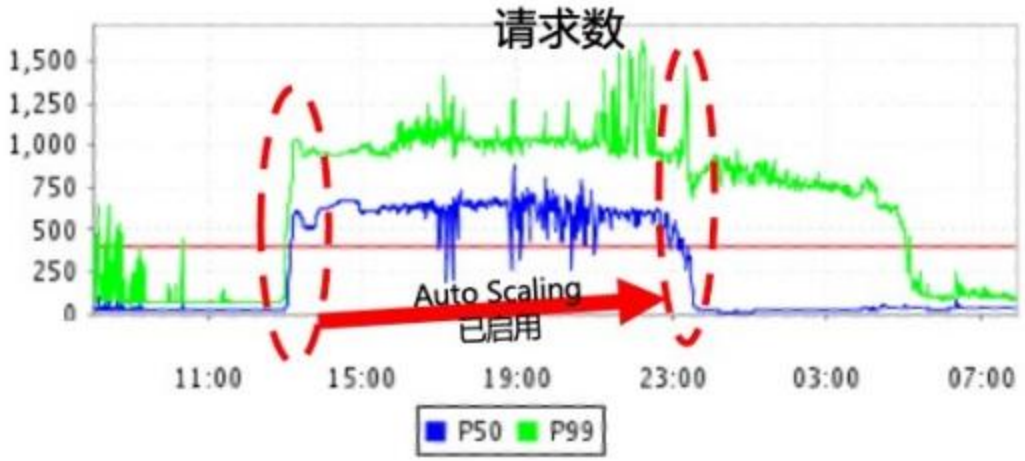


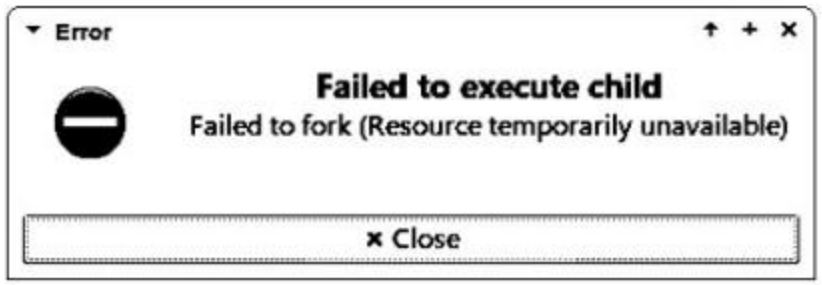

一切皆因人类设计的系统变得愈发复杂，超出了人类认知范围。

# 十年后我们面临的新挑战

**故障宿命论**：人为错误/网络异常/资源耗尽/软件bug等都不可避免

水平扩展的毒药效应





$$A=1-(1-Ax)^2$$

| 组件 | 可用性(A) | 宕机时长(按年计) |
|---|---|---|
| 一个组件 X 提供过服务 | 99% (2个9) | 3天15个小时 |
| 两个组件 X 并行提供服务 | 99.99% (4个9) | 52分钟 |
| 三个组件 X 并行提供服务 | 99.9999% (6个9) | 31秒 |

如果故障高度相关，则冗余不会增加可用性，还会产生雪崩效应。

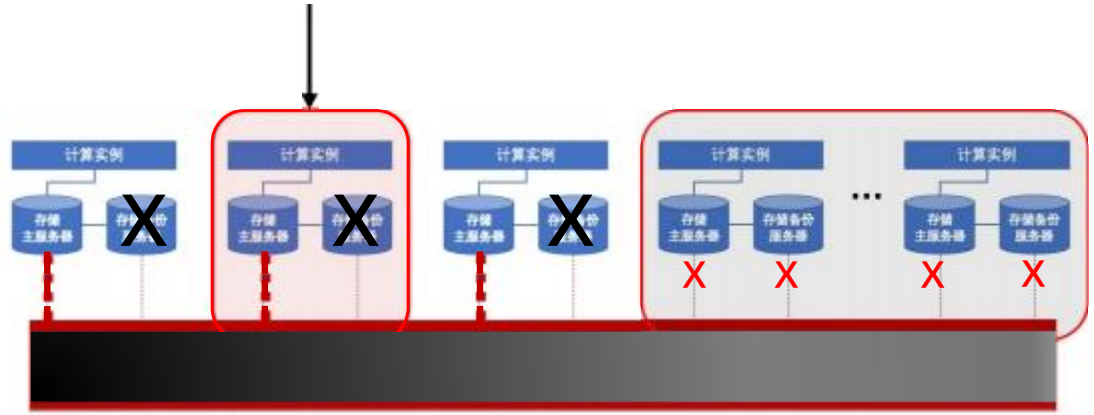
# 十年后我们面临的新挑战

**故障宿命论**：人为错误/网络异常/资源耗尽/软件bug等都不可避免

**如何减小爆炸半径？**

功能完整 | 固定资源 | 充分隔离 | 随机分区 | 自愈保障 | 按需付费

亟待新方法




配置服务：控制平面的核心部分之一

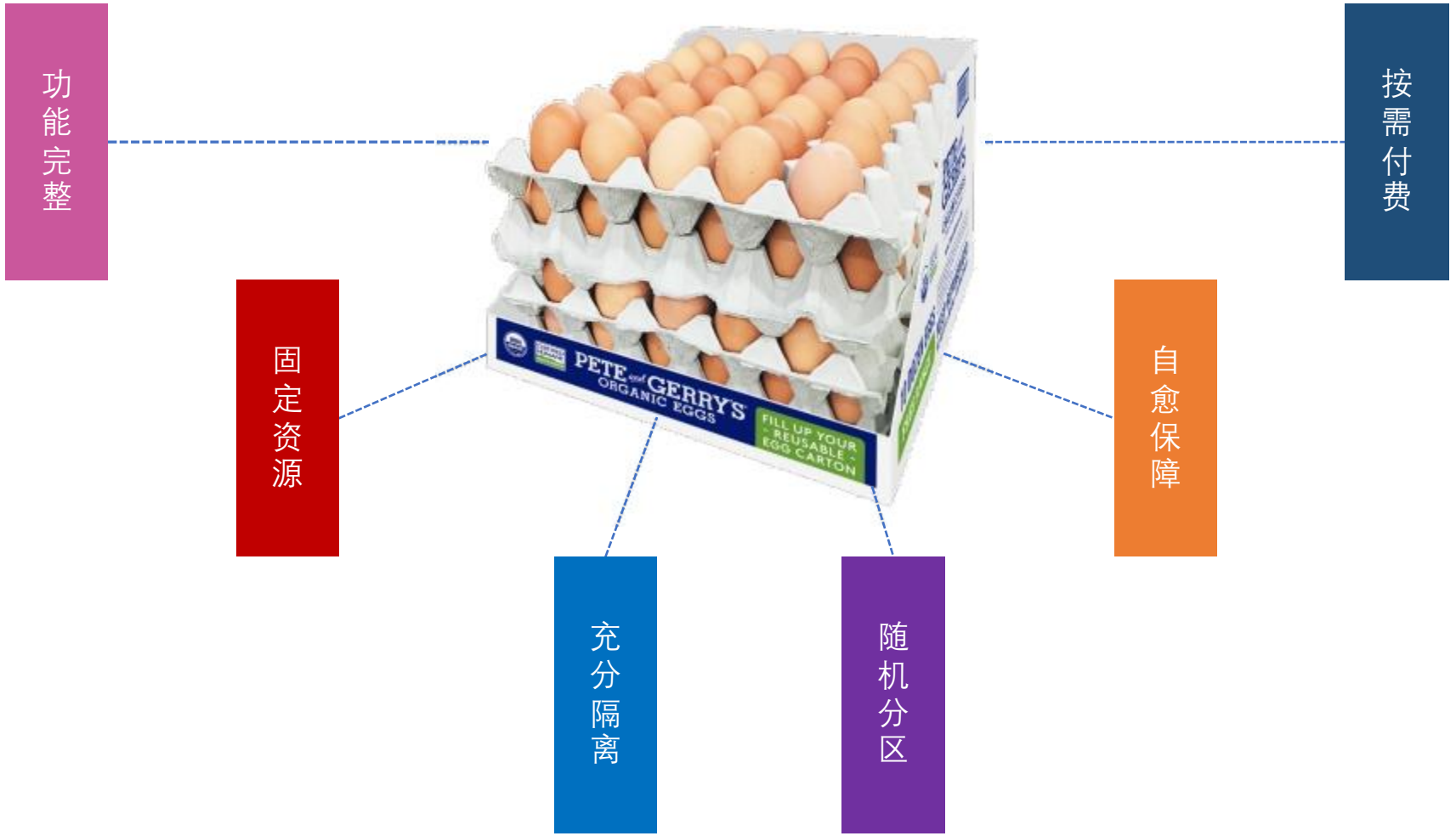
**配置服务的爆炸半径过大**

因迅速扩展引出的新问题

关联故障接连不断

# 未来又将去向哪里

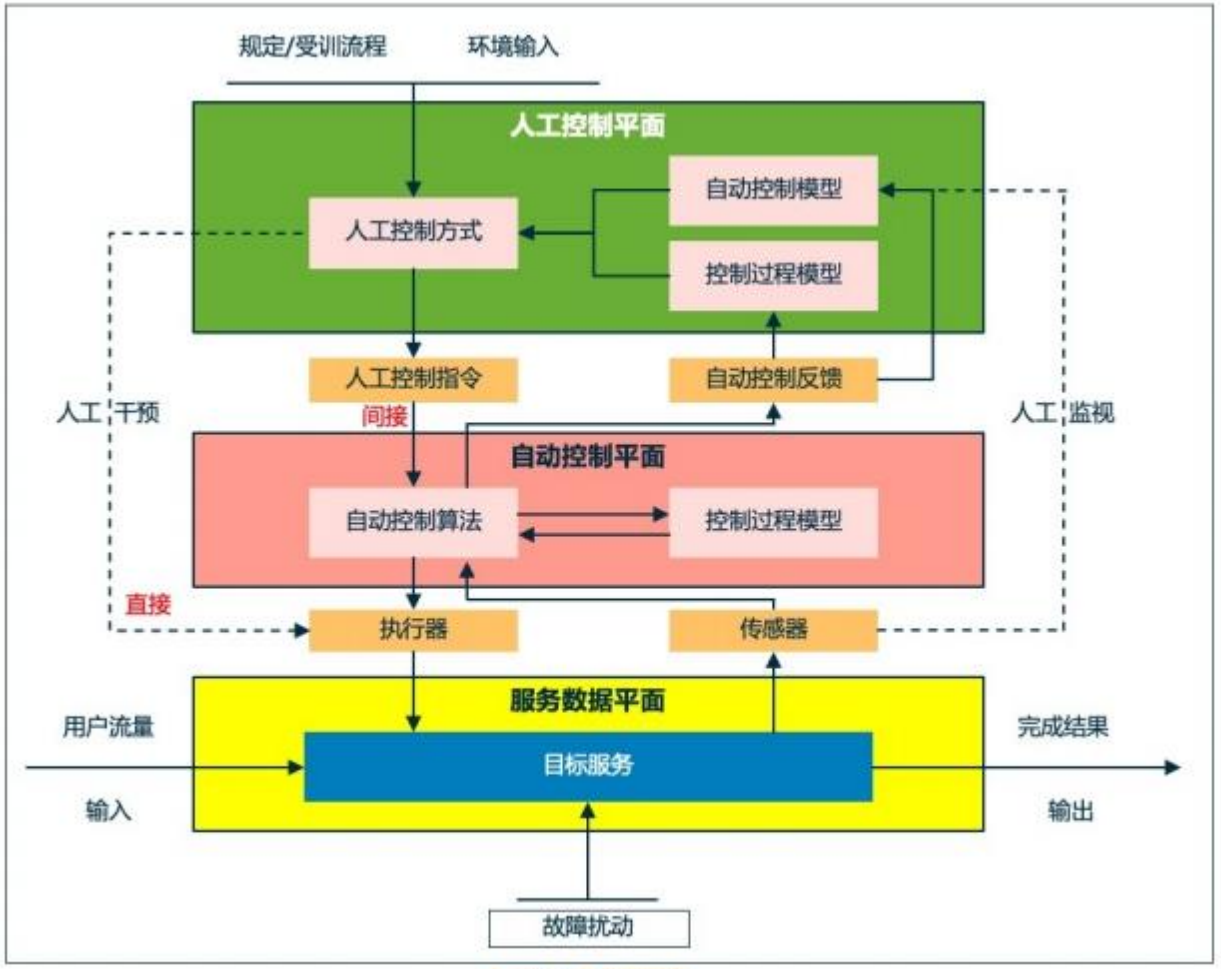
## 基于 Cell 的新架构模式



# 未来又将去向哪里

- 控制平面和数据平面在设计上的隔离

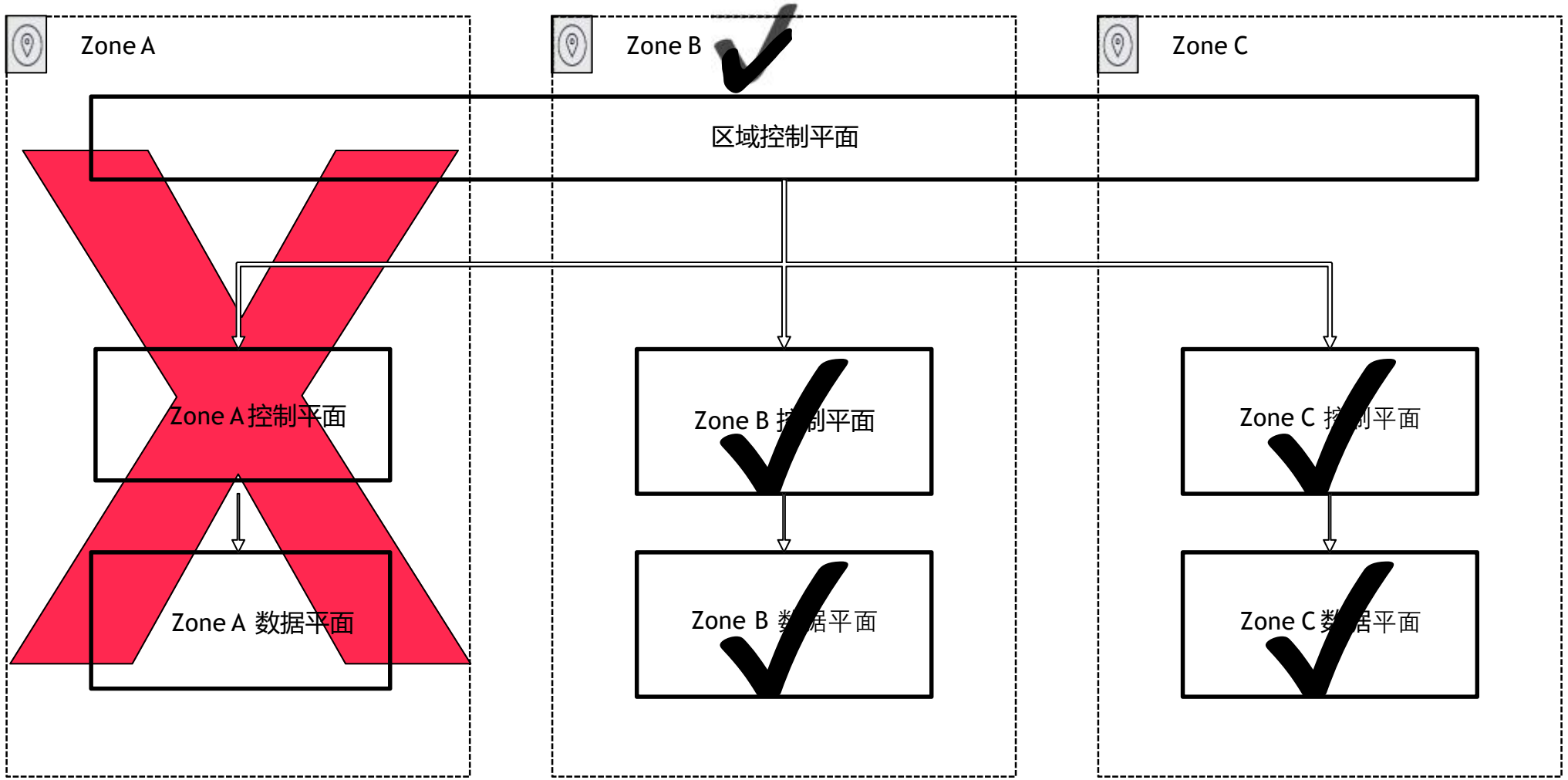
| AWS服务示例 | 控制平面 | 数据平面 |
| --- | --- | --- |
| Amazon DynamoDB | 读取表描述 API | 查询数据 API |
| Amazon EC2 | 创建实例 API | 运行时的EC2实例 |
| AWS Lambda | 创建函数 API | 调用函数 API |




STPA 分析模型

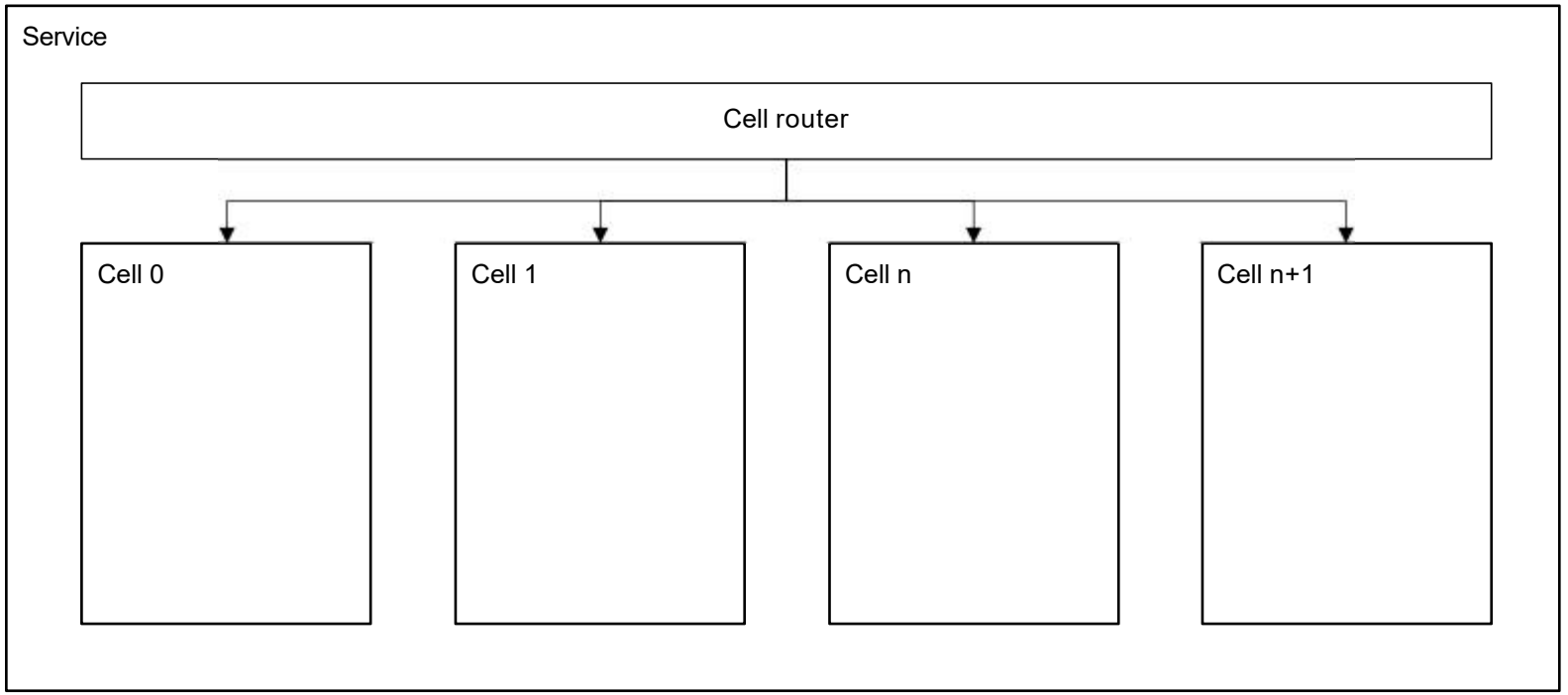
# 未来又将去向哪里

- 基础设施层面的故障隔离支撑



# 未来又将去向哪里

- Cell 架构的基本形式

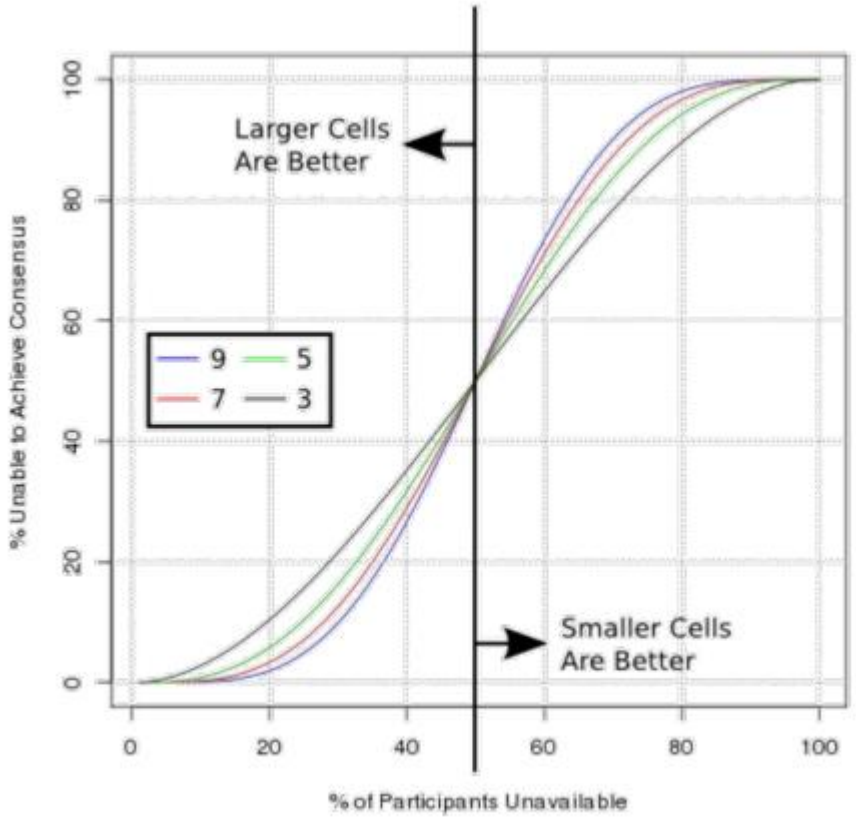



- 服务完整和独立
- 故障自留和隔离
- 路由策略 置放/
- 编排策略 扩展
- 策略
- 自我修复 可部
- 署测试单元 可
- 编排管理单元

# 未来又将去向哪里

- Cell 大小的权衡

VS




**小号 Cell**

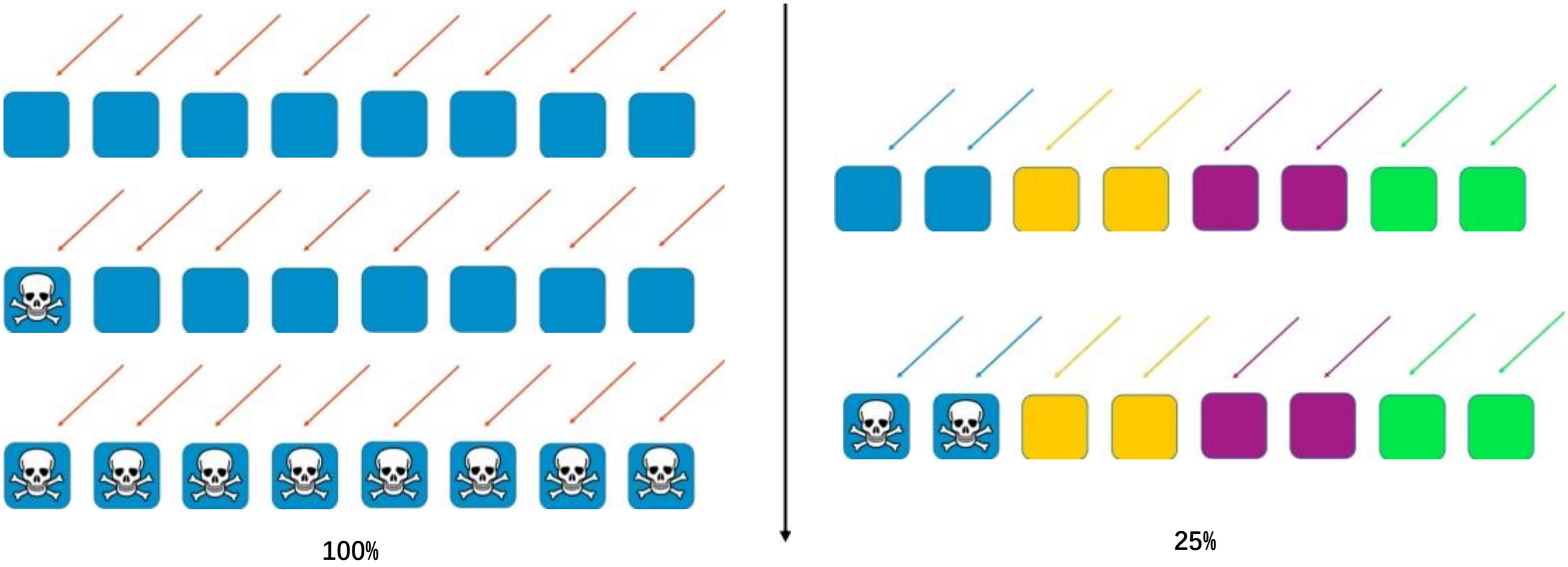
1. 更小的爆炸半径
2. 更容易部署测试
3. 更容易编排管理

**大号 Cell**

1. 成本上更优
2. 减少进一步的拆解
3. 更容易进行整体维护和管理

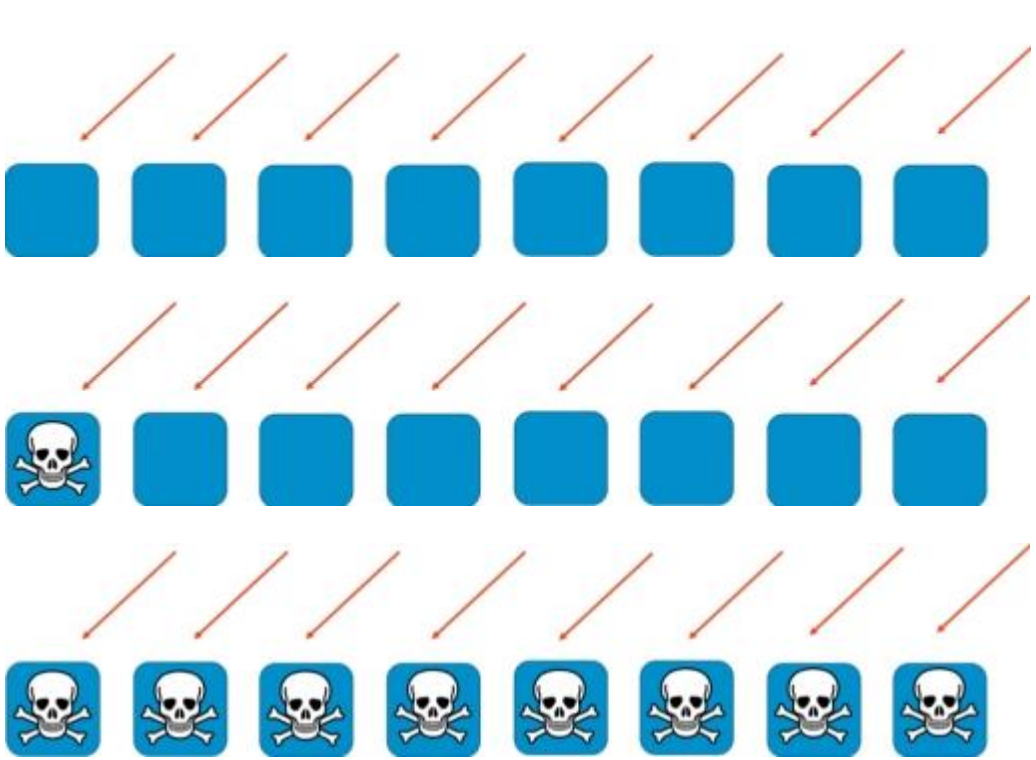
# 未来又将去向哪里

- 随机分区 – Cell 核心技术之一，应对水平扩展的毒药效应

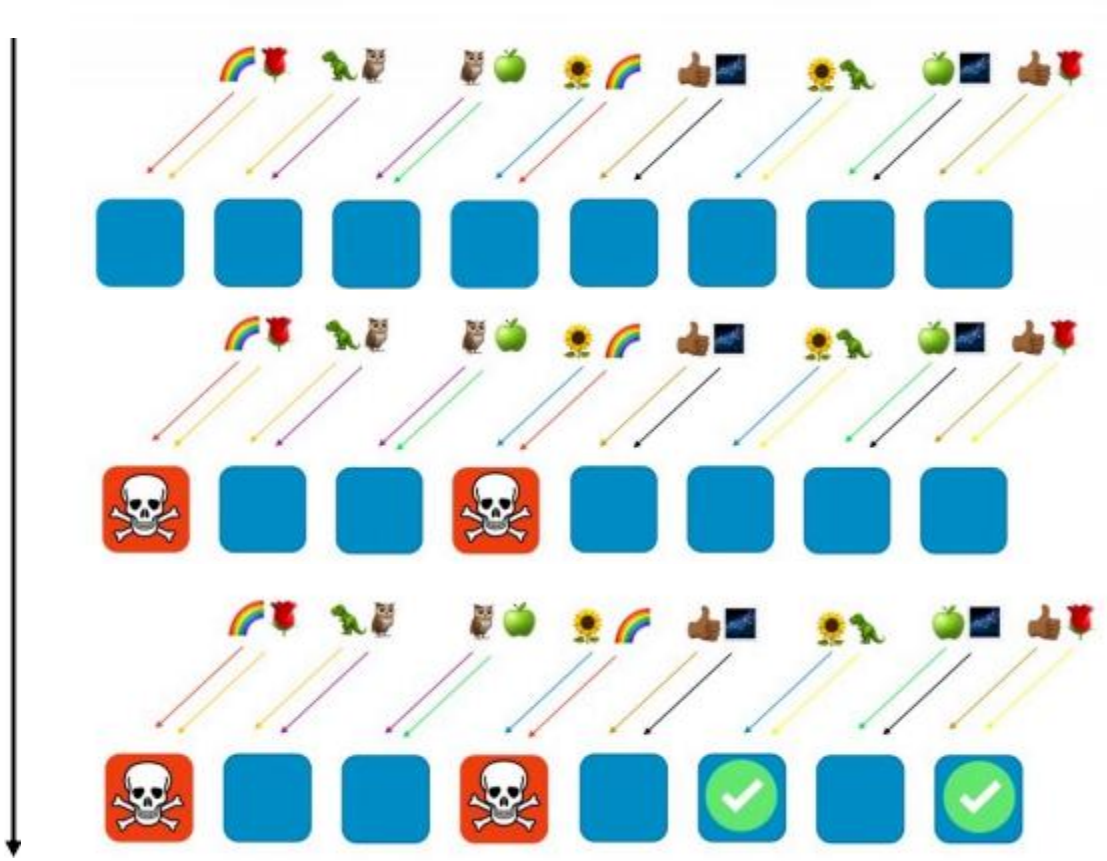


# 未来又将去向哪里

- 随机分区 – “Cell”海战术，应对水平扩展的毒药效应



100%



?? %

# 未来又将去向哪里

- 随机分区 – “Cell” 海战术，应对水平扩展的毒药效应

$$受影响率 = \frac{1}{C(|cells|,|shards|)}$$

8个cell - 每个客户2个分区

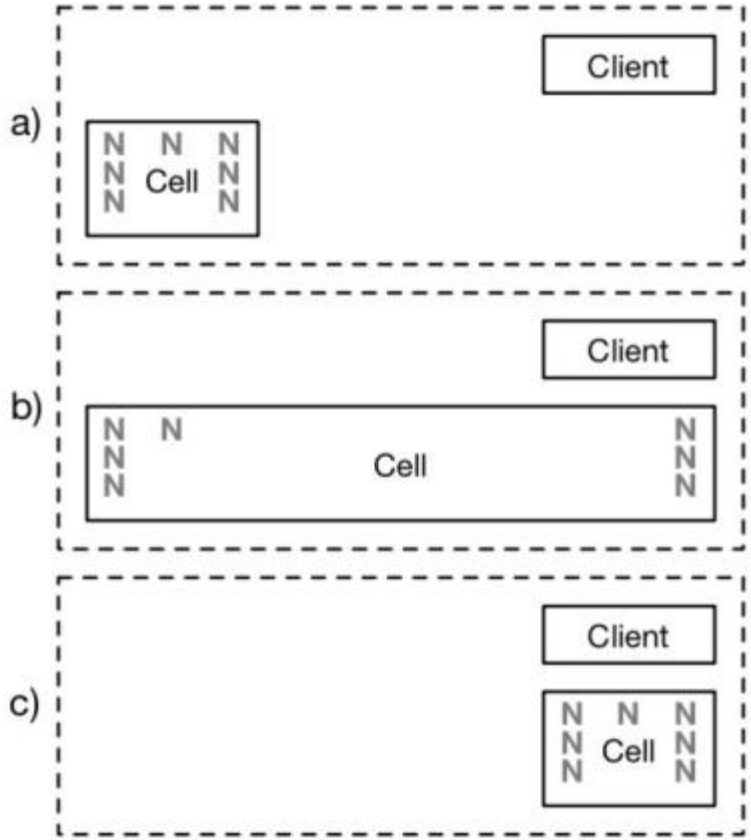
| 重叠数量 | 受影响的客户比例 |
| --- | --- |
| 0 | 53.6% |
| 1 | 42.8% |
| 2 | 3.6% |

100个cell - 每个客户5个分区

| 重叠数量 | 受影响的客户比例 |
| --- | --- |
| 0 | 77% |
| 1 | 21% |
| 2 | 1.8% |
| 3 | 0.06% |
| 4 | 0.0006% |
| 5 | 0.0000013% |

# 未来又将去向哪里

- 置放/编排策略 – 基础设施感知驱动




**存储配置服务的新 Cell 的置放策略**

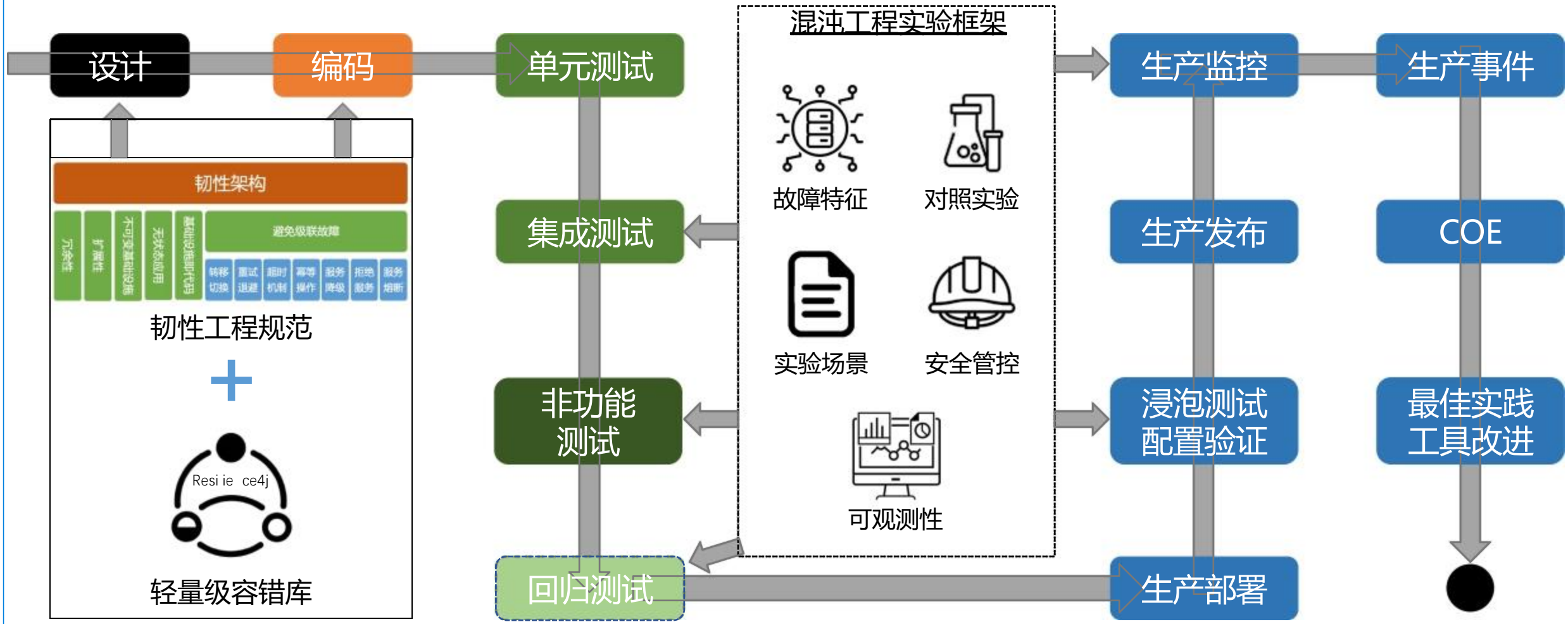
控制平面从数据中心自动发现系统，获得对数据中心电源和网络拓扑的信息，为配置服务的新 Cell 选择置放节点。

*不远不近的置放策略*

1. **不远**：靠近计算实例放置，通过网络和电源拓扑的逻辑距离衡量，以确保因远离计算实例的网络故障不会影响 Cell。
2. **不近**：必须以足够的多样性置放，不能聚集，以确保小规模故障不会导致配置服务 Cell 大面积发生故障。

# 未来又将去向哪里

- 上线交付结合新的混沌工程实践，也是对抗爆炸半径的有力武器



麦思博(msup)有限公司是一家面向技术型企业的培训咨询机构，携手2000余位中外客座导师，服务于技术团队的能力提升、软件工程效能和产品创新迭代，超过3000余家企业续约学习，是科技领域占有率第1的客座导师品牌，msup以整合全球领先经验实践为己任，为中国产业快速发展提供智库。

高可用架构主要关注互联网架构及高可用、可扩展及高性能领域的知识传播。订阅用户覆盖主流互联网及软件领域系统架构技术从业人员。 高可用架构系列社群是一个社区组织，其精神是“分享+交流”，提倡社区的人人参与，同时从社区获得高质量的内容。