

## 如何度量高可用架构设计指标

原创

疾风先生

小坤探游架构笔记

2025年06月01日 20:26

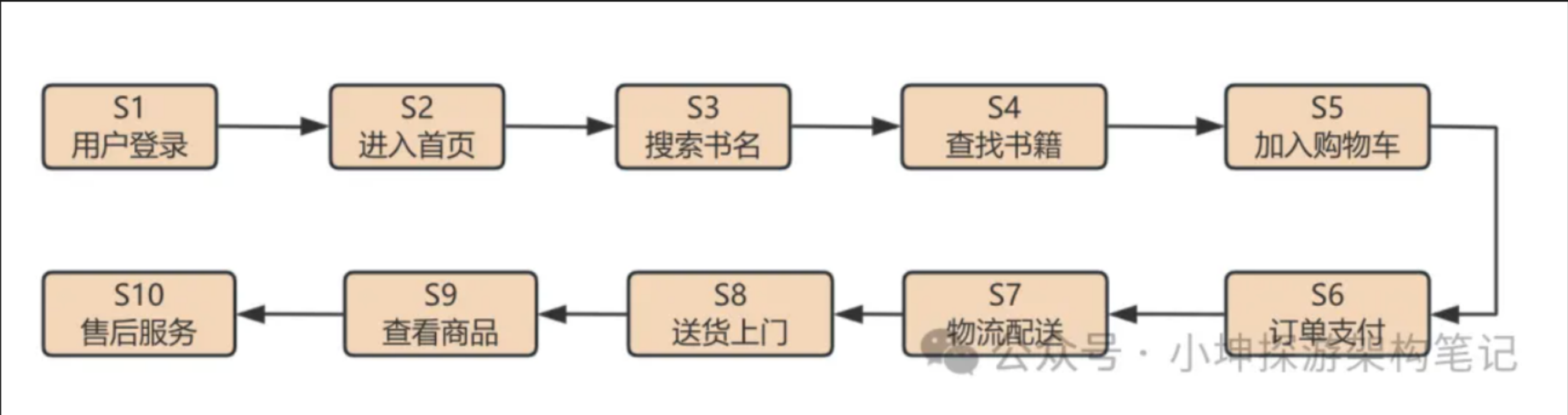
广东

点击上方[小坤探游架构笔记](#)可以订阅哦

在实际工作中, 实现一个高可用架构要比实现一个高性能架构复杂得多,因为存在故障诸多不确定性因素,比如区域性灾难、机房故障、网线电缆被挖断、网络延迟、软硬件层面故障甚至是人为失误等引起的, 这些都是现实中无法避免且存在的现象. 因此当我们设计一个高可用架构系统时,并不仅仅是冗余 + 自动故障转移的技术策略就完事了, 这只是一个开始, 因为我们还需要具备度量高可用系统的指标来持续完善与迭代.

### 什么是 SLA & SLO & SLI

对于SLA定义, 同样我觉得还是先举一个例子说明. 假如我现在要登录一个电商平台购买一本书, 那么这个时候我购买书的流程如下:



上面的流程并不重要, 重要的是我在整个过程中的体验, 即:

- 我登录网站的时候, 平台别给我来一个总是登录失败的提示,即我希望能正常登录系统,即可用性;
- 我进入首页的时候, 平台不要让我看见一个空白的页面或者是要等上好几秒才出来商品列表信息, 即我希望自己在进入首页的时候不会有延迟的, 即加载时间;
- 我进行书名关键词搜索的时候, 不要给我来一些与我搜索无关的商品,即我希望搜索出来的商品是我想要的, 即商品与搜索强相关;
- 当我要将商品加入购物车并进行支付的时候, 千万不要给我一个无库存的提示, 即我希望能看到的商品是有库存且能够进行购买的,即商品有效;
- 当我支付订单之后, 我希望物流是能够在指定的时间内(用户期望的时间)帮我送到, 即规定送货时间.

通过上述购书流程, 可以看出用户在整个过程中的期望, 而这些期望都可以用SLA来表述成对应的期望, 即如下:

#### 可用性

用户期望电商平台的登录功能是随时可用的,如果我们将这一点描述为最小可运行的时间百分比, 那么换成SLA进行表述, 那就是电商平台的登录功能至少99.9%是可用.

#### 加载时间

用户期望首页能够快速加载, 其实就是首页的响应RT, 同样地我们换算成SLA进行表述, 那就是首页的T99的RT是200ms以内.

#### 搜索相关性

用户期望通过关键词进行搜索得到自己想要的商品, 那么其实就是商品与关键词的匹配度,如果我们换算成SLA进行表述, 那么为就是搜索关键词与商品的强相关匹配度,比如是至少80%的匹配度.

#### 商品有效

用户期望购买的商品是有货的并能够立即发货, 同样我们换算成一个SLA的方式来进行表述,那么我们可以这样去表述, 至少分类中70%的商品都是有现货的.

#### 规定送货时间

用户期望自己购买的商品能够在指定的时间内送到, 那么我们换成SLA表述, 可以为“我们将在24小时内送达商品”.



这个时候我们再来看SLA可能会更容易理解, SLA即Service Level Agreement, 中文为服务等级协议, 从上述的例子可以看出我们的SLA是一种期望,什么样的期望呢? 即一个提供某种级别可靠性和性能的承诺.如果我们把这样的承诺写入正式合同,这个时候具备法律效力, 它是对外的一种承诺, 这就是SLA.比如云厂商承诺容器可用性达99.9%,如果没有达成将会按照违反合同承诺将会支付一定的赔偿.

那么什么是SLO呢? SLO即Service Level Objective.它是对内的一种承诺, 不具备法律约束, 可以说是我们团队自主制定的高可用目标,我们可以通过改进SLO来确保SLA的达成.关于SLA以及SLO的区别总结如下:

对比维度	SLO	SLA
性质	内部目标, 无法律约束	外部合同, 有法律约束
约束力	未达标无直接惩罚, 需改进	未达标需承担经济/法律责任
公开性	可公开, 也可不公开	必须公开, 避免歧义
灵活性	可调整, 适应业务变化	需法律审查, 调整成本高
示例	99.99%的可用性 (内部目标)	99.9%的可用性 (客户承诺)

谈到SLO我们也会想到另一个词SLI,什么是SLI呢? 用我们编程中的抽象和具体实现来分别描述SLO和SLI是再合适不过,SLO是抽象含义,描述是服务设计目标, 而SLI是具体的量化指标,描述的服务实际检测的具体指标,通过设计多个SLI指标并集来检测我们的SLO是否达成.关于SLO以及SLI的区别总结如下:

对比维度	SLO	SLI
定义	服务的目标值或范围 (如99.99%可用性) <a href="#">1PDF</a> 。	服务的具体量化指标 (如98%可用性、250ms延迟) <a href="#">1PDF</a> 。
作用	指导服务设计和优化, 确保满足客户期望 <a href="#">10</a> 。	监测服务实际表现, 验证SLO是否达成 <a href="#">10</a> 。
法律约束性	通常为内部目标, 无法律约束 <a href="#">4</a> 。	无法律约束, 仅用于内部监控 <a href="#">4</a> 。
灵活性	可根据业务需求调整 <a href="#">10</a> 。	固定指标, 需根据SLO动态调整 <a href="#">10</a> 。
示例	95%的请求延迟≤250ms (SLO) <a href="#">1PDF</a> 。	95%的请求延迟≤250ms (SLI) <a href="#">1PDF</a> 。 <a href="#">公众号 · 小坤探游架构笔记</a>

## 高可用架构的关键指标

我们进行高可用架构目的是以减少故障发生, 缩短故障中断时长, 控制影响范围, 增强用户体验并满足严格的 SLA/SLO要求。这个时候我们就理解了设计高可用其实就是满足用户的期望,那么我们去度量高可用设计的指标呢?

可用性



别着急, 既然我们谈高可用, 相信我们都会联想到业界常说的N个9可用性指标,其具体含义这里不再阐述,直接摘取网上的图示表示如下:

系统可用性	年故障时间	日故障时间
90% (一个九)	36.5天	2.4小时
99% (两个九)	3.65天	14.4分
99.9% (三个九)	8小时	1.44分
99.99% (四个九)	52分钟	8.6秒
99.999% (五个九)	5分钟	0.86秒
99.9999% (六个九)	32秒	86毫秒

系统采用哪种可用性等级取决于我们的SLA/SLO期望值, 然而对于可用性的测量方法我们可以采用以下的方式来计算:

可用性百分比 = (该期间的总秒数 - 系统宕机的秒数) / 该期间的总秒数

公众号 · 小坤探游架构笔记

比如像一些医疗网站, 比如每周六晚上定期关闭2小时进行内部程序维护,那么这个时候我们的网站可用性为98.8%,即其计算过程如下:

每周的秒数 = 7 day \* 24h = 168h \* 3600s

每周不可用秒数 = 2h \* 3600s

网站可用性 = (168 - 2) \* 3600s / (168h \* 3600s) = 98.8%

因此网站可用性(没有故障) = 98.8%

公众号 · 小坤探游架构笔记

而我们所说的SLI指标可以是上述测量出来的可用性指标,即可以描述为医疗系统网站的SLI的可用性98.8%.

对于可用性,我想在业界还有另一种表示,那就是采用MTTR(Mean Time To Repair)以及MTBF(Mean Time Between Failures)来进行表示,即

Availability = MTBF / (MTBF + MTTR)

其中MTBF是平均故障间隔时间,也就是我们系统正常运行的平均时间, 这个时间越长表示系统越稳定.其计算方式为:

MTBF = 总运行时间 / 故障次数

而MTTR则是平均修复时间,即包含我们的检测、定位、修复以及验证的时间,其一般公式为:

MTTR = 总修复时间 / 故障次数

然而在业界中, 对于MTTR我们一般会通过监控系统指标的告警时间进行采集,即SLI告警时间,那么我们的MTTR也将由我们监控系统告警能力决定,即:

MTTR 平均恢复时长 = 

监控告警能力直接决定 MTTR

平均发现时长 + 平均定界时长

 + 平均处置时长

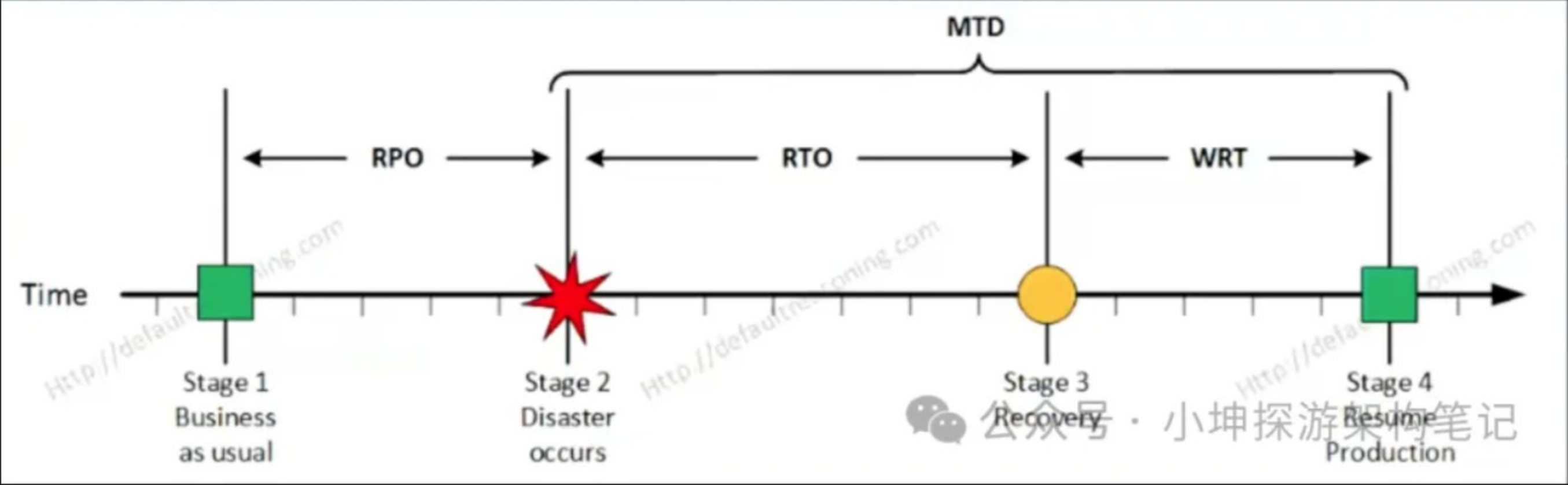
MTTR 平均恢复时长公式

公众号 · 小坤探游架构笔记

通过上述测量可用性的公式,我们可以知道可用性测量是通过非故障时间以及故障时间之间的占比得到的,也就是说如果我们要提升可用性,那么我们就需要针对发生故障的时间做出对应的努力.



于是我们又可以得到以下衡量可用性的关键指标:



RPO – Reovery Point Objective

恢复点目标, 指“最大可接受的数据损失”, 因为数据备份和数据复制都存在时间延迟,存在时间限制,不可能做到绝对实时.

RTO – Recovery Time Objective

恢复时间目标, 指“最大可接受系统的系统恢复所需的时间”, 因为定位、处理、恢复都需要时间.

WRT – Work Recovery Time

工作恢复时间, 指“系统恢复正常之后,恢复业务所需的时间”, 因为要进行各种业务检查、校验、修复.

MTD – Maximun Tolerance DownTime

最大可容忍宕机时间, 即RTO + WRT时间之和.

也就是说如果我们要在提升系统的可用性,那么我们就需要在上述的指标做出努力,对于计算高可用,我们要从RTO以及WRT去做努力; 而对于存储高可用, 那么我们还要从RPO层面去做优化与改进.

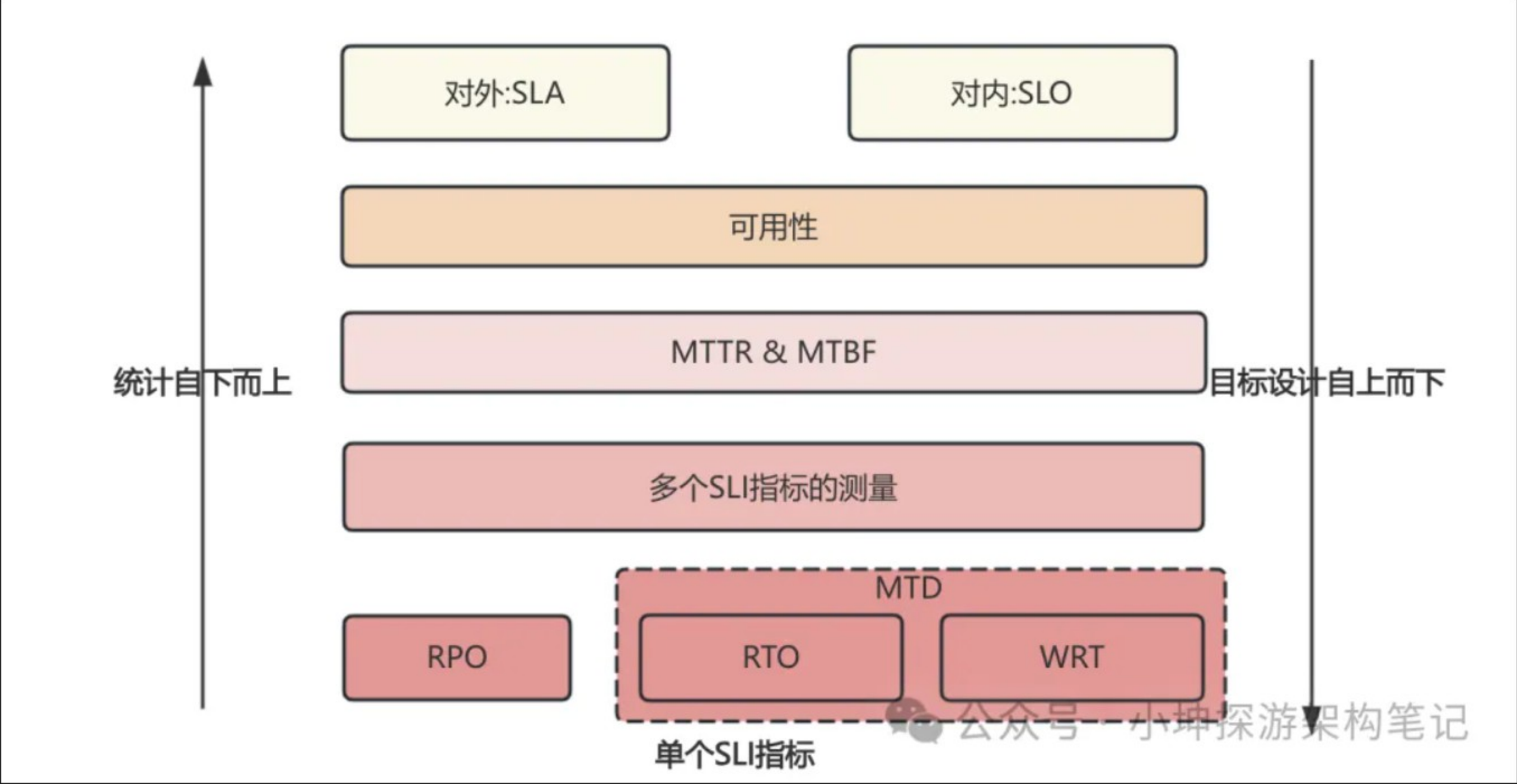
总结

至此我们了解了度量高可用的常用指标,对于SLA以及SLO,我们可以通过以下方式进行区分,其中SLA适用于与客户签订的合同,比如云服务或者是SaaS平台,而SLO更多是互联网公司内部的制定可用性目标.

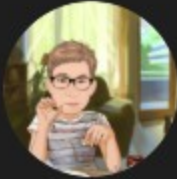
场景	SLO	SLA	法律/约束性
云服务提供商	数据库服务99.99%的可用性	数据库服务99.9%的可用性	有法律约束（未达标需赔偿） 10
电商平台	API响应时间≤100ms	API响应时间≤500ms	无法律约束（但需满足SLA） 6
内部团队	服务器故障恢复时间≤5分钟	服务器故障恢复时间≤30分钟	无法律约束（内部目标） 14
客户与供应商	服务响应时间≤500ms	服务响应时间≤300ms	有法律约束（未达标需赔偿） 4TPDF



最后我基于上述的指标进行总结如下, 我们设计目标是从上而下逐步细化到对于单个SLI,而通过SLI的每个指标自下而上进行统计来验证我们的目标的达成.



你好,我是疾风先生, 主要从事互联网搜广推行业, 技术栈为java/go/python, 记录并分享个人对技术的理解与思考, 欢迎关注我的公众号, 致力于做一个有深度,有广度,有故事的工程师,欢迎成长的路上有你陪伴,关注后回复greek可添加私人微信,欢迎技术互动和交流,谢谢!



小坤探游架构笔记

10年后端技术架构设计 | AI工程化基础建设 & 存储架构设计 & 性能优化 | 前网易、斗鱼、i...  
52篇原创内容

公众号



疾风先生

喜欢作者

架构笔记 · 目录

< 上一篇  
架构建模如何实践

下一篇 >  
什么是架构，如何培养自己的架构思维

个人观点，仅供参考