04 | 分布式系统（上）：学会用服务等级协议SLA来评估你的系统

2019-04-24 蔡元楠 来自北京

《大规模数据处理实战》

你好，我是蔡元楠。

从今天开始，我们进入专栏的第二模块。通过这一模块的学习，带你一起夯实大规模数据处理的基础。

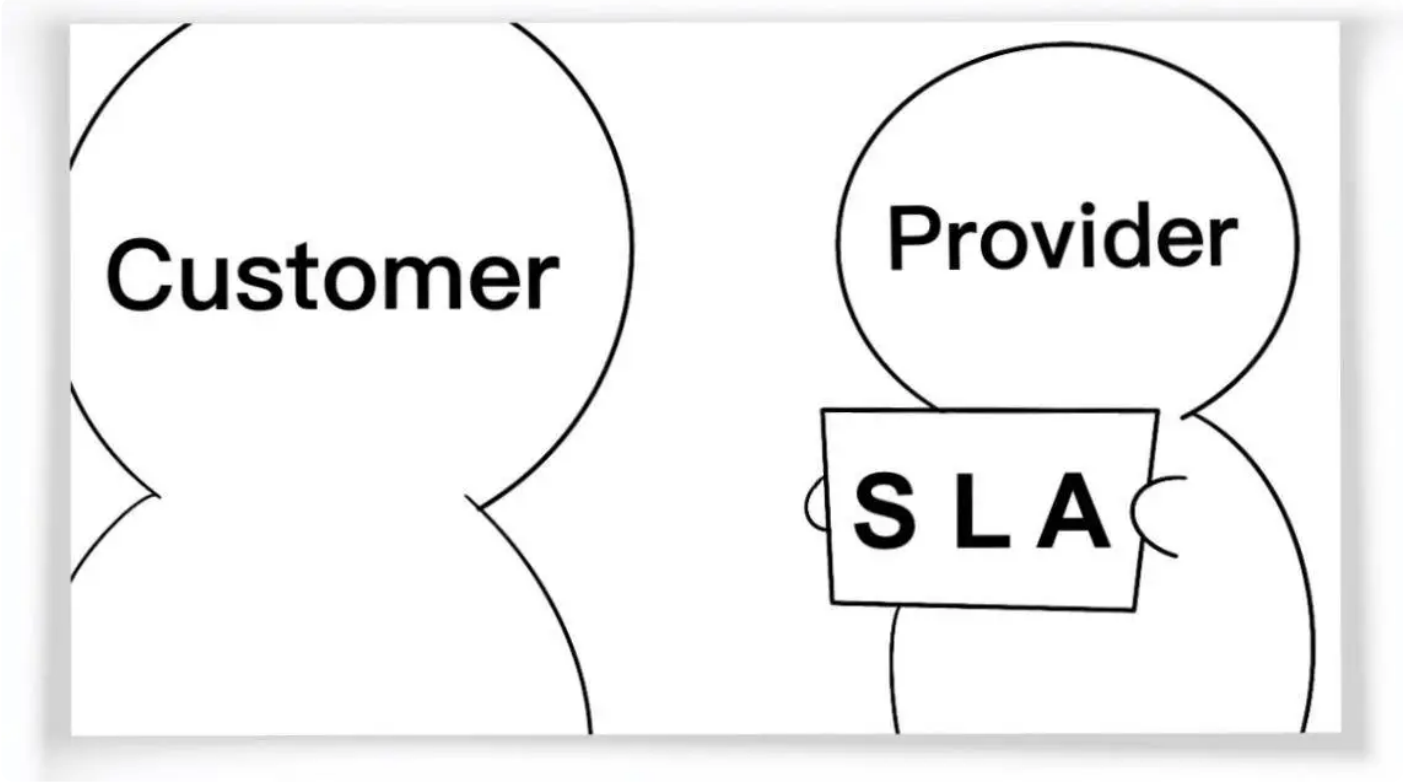
首先，我将结合硅谷顶尖科技公司的**最佳实践** (Best Practice)，和你一起分享在设计分布式系统架构时，我们有可能会碰到哪些雷区？又有哪些必备的基础知识？

在硅谷一线大厂所维护的系统服务中，我们经常可以看见 SLA 这样的承诺。

例如，在谷歌的云计算服务平台 Google Cloud Platform 中，他们会写着“99.9% Availability”这样的承诺。那什么是“99.9% Availability”呢？

要理解这个承诺是什么意思，首先，你需要了解到底什么是 SLA？

SLA（Service-Level Agreement），也就是**服务等级协议**，指的是系统服务提供者（Provider）对客户（Customer）的一个服务承诺。这是衡量一个大型分布式系统是否“健康”的常见方法。



在开发设计系统服务的时候，无论面对的客户是公司外部的个人、商业用户，还是公司内的不同业务部门，我们都应该对自己所设计的系统服务有一个定义好的 SLA。

因为 SLA 是一种服务承诺，所以指标可以多种多样。根据我的实践经验，给你介绍最常见的四个 SLA 指标，可用性、准确性、系统容量和延迟。

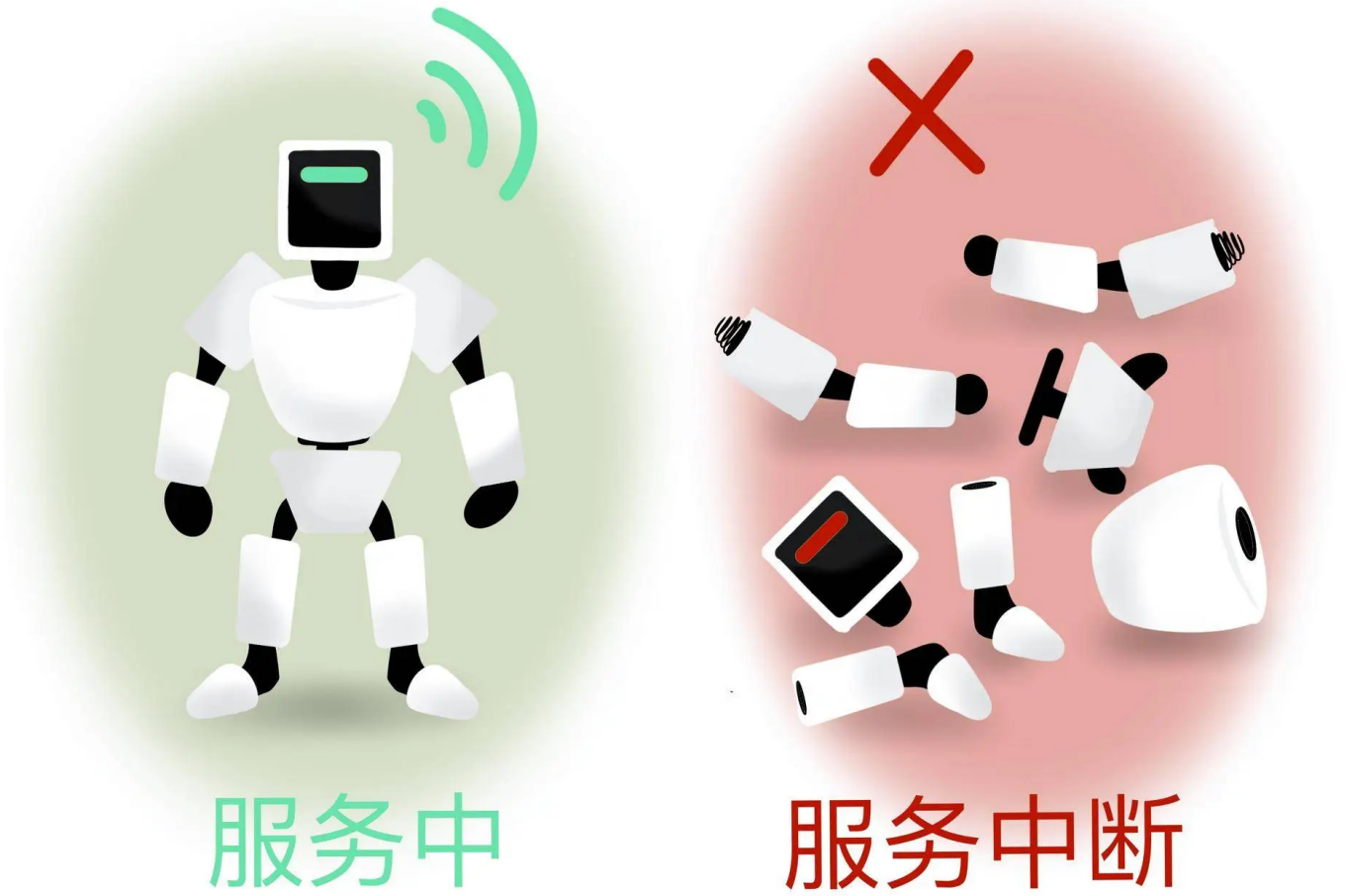
1. 可用性（Availabilty）

可用性指的是系统服务能正常运行所占的时间百分比。

如果我们搭建了一个拥有“100%可用性”的系统服务，那就意味着这个系统在任何时候都能正常运行。是不是很完美？但真要实现这样的目标其实非常困难，并且成本也会很高。

我们知道，即便是大名鼎鼎的亚马逊 AWS 云计算服务这样大型的、对用户来说极为关键的系统，也不能承诺 100%的可用性，它的系统服务从推出到现在，也有过服务中断（Service

Outage）的时候。



对于许多系统而言，四个 9 的可用性（99.99% Availability，或每年约 50 分钟的系统中断时间）即可以被认为是**高可用性**（High availability）。

说到这里，我来为你揭开一开始所提到的“99.9% Availability”的真实含义。

“99.9% Availability”指的是一天当中系统服务将会有大约 86 秒的服务间断期。服务间断也许是因为系统维护，也有可能是因为系统在更新升级系统服务。

86 秒这个数字是怎么算出来的呢?

99.9% 意味着有 0.1% 的可能性系统服务会被中断，而一天中有 24 小时 × 60 分钟 × 60 秒，也就是有 (24 × 60 × 60 × 0.001) = 86.4 秒的可能系统服务被中断了。而上面所说的四

个 9 的高可用性服务就是承诺可以将一天当中的服务中断时间缩短到只有 (24 × 60 × 60 × 0.0001) = 8.64 秒。

2. 准确性（Accuracy）

准确性指的是我们所设计的系统服务中，是否允许某些数据是不准确的或者是丢失了的。如果允许这样的情况发生，用户可以接受的概率（百分比）是多少？

这该怎么衡量呢？不同的系统平台可能会用不同的指标去定义准确性。很多时候，系统架构会以**错误率**（Error Rate）来定义这一项 SLA。

怎么计算错误率呢？可以用导致系统产生内部错误（Internal Error）的有效请求数，除以这期间的有效请求总数。

$$\text{错误率} = \frac{\text{导致系统产生内部错误的有效请求数}}{\text{期间的有效请求总数}}$$

例如，我们在一分钟内发送 100 个有效请求到系统中，其中有 5 个请求导致系统返回内部错误，那我们可以说这一分钟系统的错误率是 5 / 100 = 5%。

下面，我想带你看看硅谷一线公司所搭建的架构平台的准确性 SLA。

Google Cloud Platform 的 SLA 中，有着这样的准确性定义：每个月系统的错误率超过 5% 的时间要少于 0.1%，以每分钟为单位来计算。

而亚马逊 AWS 云计算平台有着稍微不一样的准确性定义：以每 5 分钟为单位，错误率不会超过 0.1%。

你看，我们可以用错误率来定义准确性，但具体该如何评估系统的准确性呢？一般来说，我们可以采用**性能测试**（Performance Test）或者是**查看系统日志**（Log）两种方法来评估。

具体的做法我会在后面展开讲解，今天你先理解这项指标就可以了。

3. 系统容量（Capacity）

在数据处理中，系统容量通常指的是**系统能够支持的预期负载量是多少**，一般会以每秒的请求数为单位来表示。

我们常常可以看见，某个系统的架构可以处理的 QPS （Queries Per Second）是多少又或者 RPS（Requests Per Second）是多少。这里的 QPS 或者是 RPS 就是指系统每秒可以响应多少请求数。

我们来看看之前 Twitter 发布的一项数据，Twitter 系统可以响应 30 万的 QPS 来读取 Twitter Timelines。这里 Twitter 系统给出的就是他们对于系统容量（Capacity）的 SLA。

你可能会问，我要怎么给自己设计的系统架构定义出准确的 QPS 呢？以我的经验看，可以有下面这几种方式。

第一种，是使用**限流**（Throttling）的方式。

如果你是使用 Java 语言进行编程的，就可以使用 Google Guava 库中的 RateLimiter 类来定义每秒最多发送多少请求到后台处理。

假设我们在每台服务器都定义了一个每秒最多处理 1000 个请求的 RateLimiter，而我们有 N 台服务器，在最理想的情况下，我们的 QPS 可以达到 1000 * N。

这里要注意的雷区是，这个请求数并不是设置得越多越好。因为每台服务器的内存有限，过多的请求堆积在服务器中有可能会导致**内存溢出**（Out-Of-Memory）的异常发生，也就是所有请求所需要占用的内存超过了服务器能提供的内存，从而让整个服务器崩溃。

第二种，是在系统交付前进行**性能测试**（Performance Test）。

我们可以使用像 Apache JMeter 又或是 LoadRunner 这类型的工具对系统进行性能测试。这类工具可以测试出系统在峰值状态下可以应对的 QPS 是多少。

当然了，这里也是有雷区的。

有的开发者可能使用同一类型的请求参数，导致后台服务器在多数情况下**命中缓存**（Cache Hit）。这个时候得到的 QPS 可能并不是真实的 QPS。

打个比方，服务器处理请求的正常流程需要查询后台数据库，得到数据库结果后再返回给用户，这个过程平均需要 1 秒。在第一次拿到数据库结果后，这个数据就会被保存在缓存中，而如果后续的请求都使用同一类型的参数，导致结果不需要从数据库得到，而是直接从缓存中得到，这个过程我们假设只需要 0.1 秒。那这样，我们所计算出来的 QPS 就会比正常的高出 10 倍。所以在生成请求的时候，要格外注意这一点。

第三种，是分析系统在实际使用时产生的**日志**（Log）。

系统上线使用后，我们可以得到日志文件。一般的日志文件会记录每个时刻产生的请求。我们可以通过系统每天在最繁忙时刻所接收到的请求数，来计算出系统可以承载的 QPS。

不过，这种方法不一定可以得到系统可以承载的最大 QPS。

在这里打个比喻，一家可以容纳上百桌客人的餐馆刚开业，因为客流量还比较小，在每天最繁忙的时候只接待了 10 桌客人。那我们可以说这家餐馆最多只能接待 10 桌客人吗？不可以。

同样的，以分析系统日志的方法计算出来的 QPS 并不一定是服务器能够承载的最大 QPS。想要得到系统能承受的最大 QPS，更多的是性能测试和日志分析相结合的手段。

4. 延迟（Latency）

延迟指的是**系统在收到用户的请求到响应这个请求之间的时间间隔**。

在定义延迟的 SLA 时，我们常常看到系统的 SLA 会有 p95 或者是 p99 这样的延迟声明。这里的 p 指的是 percentile，也就是百分位的意思。如果说一个系统的 p95 延迟是 1 秒的话，

那就表示在 100 个请求里面有 95 个请求的响应时间会少于 1 秒，而剩下的 5 个请求响应时间会大于 1 秒。

下面我们用一个具体的例子来说明延迟这项指标在 SLA 中的重要性。

假设，我们已经设计好了一个社交软件的系统架构。这个社交软件在接收到用户的请求之后，需要读取数据库中的内容返回给用户。

为了降低系统的延迟，我们会将数据库中内容放进缓存（Cache）中，以此来减少数据库的读取时间。在系统运行了一段时间后，我们得到了一些缓存命中率（Cache Hit Ratio）的信息。有 90% 的请求命中了缓存，而剩下的 10% 的请求则需要重新从数据库中读取内容。

这时服务器所给我们的 p95 或者 p99 延迟恰恰就衡量了系统的最长时间，也就是从数据库中读取内容的时间。作为一个优秀架构师，你可以通过改进缓存策略从而提高缓存命中率，也可以通过优化数据库的 Schema 或者索引（Index）来降低 p95 或 p99 延迟。

总而言之，当 p95 或者 p99 过高时，总会有 5% 或者 1% 的用户抱怨产品的用户体验太差，这都是我们要通过优化系统来避免的。

小结

通过今天的内容，你可以发现，定义好一个系统架构的 SLA 对于一个优秀的架构师来说是必不可少的一项技能，也是一种基本素养。

特别是当系统架构在不停迭代的时候，有了一个明确的 SLA，我们可以知道下一代系统架构的改进目标以及优化好的系统架构是否比上一代的系统 SLA 更加优秀。

我们通常会使用可用性、准确性、系统容量、延迟这四个指标来定义系统架构的 SLA。

思考题

你可以思考一下，在自己所在的开发项目中，系统的 SLA 是什么呢？又有什么方面可以优化的呢？

欢迎你把答案写在留言区，与我和其他同学一起讨论。如果你觉得有所收获，也欢迎把文章分享给你的朋友。

精选留言 (32)

Joseph
2019-04-24

在实际应用SLA的时候，有两点不解：
1. 在设计系统之初，大家拍脑袋来定义SLA。这个时期，SLA对应需要付出的成本还不明确，这样大家都会趋于订出很高标准。这种情况有好的解决办法吗？
2. 虽然定义了SLA，但在架构设计的时候，如何评估架构是否能满足SLA呢？等到软件实现了再来测试，似乎有点太晚了……请教老师一般是如何处理这两个问题的呢？

共 1 条评论 >　　49

Dany
2019-04-27

我觉得这一节超赞。基础概念很重要，很重要，很重要。
实战这种事情，有的是时间去practice，而SLA这几个关键概念，会成为很多人理解的迷雾。
当我发现我的lead，拉着一大堆自己也还没理解清楚SLA指标去拉KPI，扛大旗的时候，我才进一步，深刻体会到这一节内容的重要性^ ^

25

时间是最真的答案
2019-04-24

作为一个学习大数据的新手，希望作者能用几篇文章讲解大数据处理中使用的技术如何搭建，运行，优化的，以及各个技术如何结合使用，这样新手也能玩起来

> 作者回复: 谢谢你的建议！我在第10和第11讲里有根据所讲的基础知识展开介绍两种实战中的经典架构，也配合了一些硅谷这边的应用实例作讲解，希望能对你有所帮助。

20

sxpujs

2019-04-24

这几节干货有点少啊，也缺少一些实战和实例。

作者回复: 这都是后面马上要用到的知识。不可能纯狠操猛干而没有知识支撑的。

20

mgxian
2019-04-24

这个SLA和一般服务监控指标 RED 原则有点像
R rate 请求速率 qps
E errors 错误数错误率
D duration 延迟
再加一个 服务可用性指标等级 就是今天讲的服务等级了

13

孙稚昊
2019-04-25

我们公司为了高并发QPS, 以前的python server 全部换成 Golang 了，Golang 做高并发是真的有优势

10

明翼
2019-04-24

我们系统一般可用性，系统容量有做定义也多在招投标的文件中，至于准确性和延迟更没有严格测试，准确性这个我觉得不好测试吧，如果知道出错了，干嘛不修改那，老师业界硅谷大厂如何测试准确性那?

7

zhihai.tu
2019-04-24

在银行做大数据平台的研发工作，也从可用性、准确性、系统容量、延迟四个指标来谈谈SLA，理解的不是很深，如有错误和不妥，请老师指导和更正：
1、可用性：不管是hadoop还是mppdb，数据库本身提供了本地高可用，另外，采用了双园区主备设计，提供了园区自动切换服务，保证了园区之间的高可用。
2、准确性：流数据处理平台，存在数据丢失的可能性。具体百分比应该是小于5%的。

3、系统容量：采用限流的方式，通过参数设置，从而控制最大的并发数量。
4、延迟：hadoop平台由于延迟较高，设计了异步处理请求及多线程技术，提高用户体验。

5

何妨
2019-12-17

记笔记:
定义SLA的四个维度
可用性:4个9，1日8秒
准确性:容错标准
系统容量:QPS,RPS
延迟:p95,p99

4

Tomcat
2019-04-24

SLA，即服务等级协议，规定了我们的工程的质量和目标，这使得我们的工作具有可衡量的尺度。
以前我在中国移动做专线提供服务的时候，对这个颇为敏感，移动的专线产品，确实有许多不足之处，但是这让我构建了服务质量可以使用具体技术指标度量的理念。
对于现在我正在做的产品，同样也有一些苛刻的要求，所以通过本文，我构建了服务质量度量体系～

作者回复: 谢谢你的经验之谈!

4

hufox
2019-04-24

今天学到了什么SLA，请问老师，大数据平台中缓存的设计重要吗？一般如何设计？希望老师后面能讲讲新手如何搭建一个大数据平台，把整个流程运行起来，帮助更好的理解大数据处理流程!

3

wmg
2019-04-24

老师我的理解SLA更适用于衡量oltp系统，和大数据处理系统有哪些联系呢？我的理解可能有

误，老师指教

2

leesper

2019-07-17

“当 p95 或者 p99 过高时，总会有 5% 或者 1%的用户抱怨产品的用户体验太差”，这个不可小视，因为很可能这1%或者5%用户就是很资深的用户，比如他/她在这个平台上买过很多东西所以响应慢，这个一定要做优化

1

滨 风暴

2019-05-22

我的理解是为了提高SLA，系统就要达到一定的冗余度，对于大数据来说存储和计算使用的资源就更多，所以定义SLA的时候，是不是还是要考虑一下成本，或者有没有提供高SLA的轻量化系统架构?

作者回复: 的确如此

1

Blakemmmm

2019-05-16

请问老师可用性的数据一般是如何测出或算出的呢？内部测试时不可能测试那么长时间，而短时间的测试又无法反应随着运行时间增长导致的系统更容易出问题的概率。

作者回复: 谢谢你的提问！可用性数据一般都需要系统运行一段时间的，无论是内测也好，还是其他方式也好。如果用户量没有这么大的话，可能需要自己写Prober去模拟一些常用操作去做测试。

1

vic5210jp

2019-04-24

有4个问题不太明白，希望可以交流一下。
1.系统容量和延迟可以理解为吞吐量和响应速度么?
2.不同的业务访问的数据量不同，因而延迟也有所不同，用p95或者是p99这样描述整个系统的延迟是否不太准确。
3.除了介绍的SLA服务等级协议，系统的扩展性和复杂度等这些是否也应该被纳入一个系统的

评价标准中。
4.在高可用中，99.99%这种在系统上线前是如何测试得出的？一般我们是根据运行一段时间的情况来预估的，其实并不准确。

1

再见理想
2022-05-25

服务等级协议（SLA）指标：
可用性 高可用系统需要达到3个9 甚至4个9的服务可用性
准确性 数据一致性
系统容量 系统能承受的最大负载 可以用限流、性能优化等方式提升
延迟 可以使用缓存技术缓解

William Ning
2022-04-04

老师同学好，关于文中的可用性，这个是怎么统计出来的？

有个问题，比如，一个很简单的系统网站，单体架构，在对外提供访问的一年里都很稳定，说可用性100%，不合适吧，可用性如何衡量呢？

高景洋
2021-09-18

对我们的系统而言，我觉得 1、覆盖率、2、更新频率 尤为重要。

1、业务方侧数据量1100W+
2、我们会有同步程序，将业务方侧的新数据及状态变化的数据，同步到我们的全量库（hbase）中。因为只有业务侧的数据，在我们的全量库中，数据才会参与其他信息的补充。补充信息后的数据才会参与下游的数据流转。
3、因为有脏数据的问题，入库前会做过滤。当前千万级的数据每天会有3000-4000条，因为脏数据问题覆盖不到，未覆盖率 0.03% - 0.04%
4、这个未覆盖指标 对我们尤为重要，假如这个指标超过0.1%。 就意味着，脏数据比例变高，有环节出现问题，需要让业务侧排查处理

5、业务侧的数据进到我们的全量库后，我们会对数据的其他信息做补充。

6、例：千万级的数据，我们需要半小时内补充完数据，进入下游的数据流转。
7、这个半小时，也是我们的一个指标。 类似课程里说的 第四点：延时。只不过课程里说的是，单个请求的延时，而我们这个指标是 数据批量处理的频率时间


Phantom01
2020-12-10

是只有服务才有sla吗？软件会有sla吗？ 比如 hadoop，我们可以有吞吐量(iops)，单机容量，资源利用率。但是像可用性和错误率就不好定义。