

# 什么是系统可用性？如何提升可用性？

猿java 2024-08-11 542 阅读4分钟 专栏：技术干货

你好，我是猿java。

日常开发中，我们经常听到系统的可用性是几个 9 这样的描述，因此，这篇文章，我们将探讨什么是可用性、如何计算可用性以及提高可用性的一些常用策略。

## 什么是系统可用性？

系统的可用性（Availability）是衡量一个系统在特定时间段内能够正常运行并提供服务的能力。

可用性计算方式：

▼ text 代码解读 复制代码

```
1 Availability = Uptime / (Uptime + Downtime)
```

- Uptime：运行时间，系统正常运行且可访问的时间段。
- Downtime：停机时间，由于故障、维护或其他问题而导致系统不可用的时期。

举个例子，假如一年 365天，停机总时间 2天，那么可用性的计算为：

▼ text 代码解读 复制代码

```
1 Availability = (365-2) / 365
2               = 363 / 365
3               = 0.99452
4 转换成百分比 = 99.452 %
```

## 可用性等级

系统可用性，可用性通常用“9”表示，可用性越高，停机时间就越少。如下图所示：

| Availability %<br>可用性 | Downtime per year<br>全年不可用总时间 | Commonly referred as<br>通常叫法 |
|-----------------------|-------------------------------|------------------------------|
| 99%                   | 3.65 days                     | "Two nines"<br>2个9           |
| 99.9%                 | 8.76 hours                    | "Three nines"<br>3个9         |
| 99.99%                | 52.56 minutes                 | "Four nines"<br>4个9          |
| 99.999%               | 5.26 minutes                  | "Five nines"<br>5个9          |
| 99.9999%              | 31.5 seconds                  | "Six nines"<br>6个9           |

## 如何提升系统可用性？



猿java

@公众号【猿java】

作者榜No.4 优秀作者 人气作者

240

文章

593k

阅读

1.7k

粉丝

关注

私信

目录 收起

故障检测与自动恢复

数据备份与恢复

负载均衡

容错设计

定期维护与更新

使用高可用性云服务

网络优化

总结

学习交流

相关推荐

你如何理解高可用性？

234阅读 · 1点赞

什么是高可用性 （HA）？

110阅读 · 0点赞

什么是CDN？它是如何工作的？

972阅读 · 19点赞

漫话：如何给女朋友解释什么是系统可...

5.3k阅读 · 22点赞

什么是网站监测？网站监测包括哪些方...

139阅读 · 0点赞

精选内容

SpringBoot中使用ResponseBodyEmitt...

码猿技术专栏 · 187阅读 · 2点赞

【DLL】基于Python与C++的FFmpeg与...

小小杨树 · 29阅读 · 1点赞

10 . Docker 中的 DockerFile 解析(各种...

RainbowSea · 56阅读 · 0点赞

9. Docker 当中的复杂安装(MySQL主从...

RainbowSea · 37阅读 · 2点赞

Pandas高级数据处理：数据仪表板制作

Jimaks · 24阅读 · 0点赞

找对属于你的技术圈子

回复「进群」加入官方微信群



Captured by FireShot Pro: 19 2月 2025, 15:59:48

https://getfireshot.com



提升系统可用性的方法包括但不限于以下几种：

### 冗余设计

冗余设计是提升系统可用性常用的方式，比如，分布式部署，异地多活等，冗余设计常见的技术策略主要有以下 3 种：

- **服务器冗余**：部署多个服务器来处理请求，确保如果一个服务器出现故障，其他服务器可以继续提供服务。
- **数据库冗余**：创建一个副本数据库，如果主数据库发生故障，该数据库可以接管。
- **地理冗余**：将资源分布在多个地理位置，以减轻区域故障的影响。

### 故障检测与自动恢复

当检测到故障时，故障切换机制会自动切换到冗余系统。常用的技术策略有：

- **监控系统**：使用监控工具（如Nagios、Zabbix）实时监控系统状态，及时发现问题。
- **自动化恢复**：配置自动化脚本或服务（如AWS Auto Scaling）在检测到故障时自动重启或替换故障组件。

### 数据备份与恢复

在实际开发中，绝大部署业务都是对数据进行处理，因此数据的重要性不言而喻，对于数据可用性常用的技术点有：

- **定期备份**：定期备份重要数据，确保在数据丢失或损坏时能够快速恢复。
- **灾难恢复计划**：制定并测试灾难恢复计划，以确保在重大故障或灾难发生时能够迅速恢复系统运营。

### 负载均衡

负载均衡在多个服务器之间分配传入的网络流量，以确保没有单个服务器成为瓶颈，从而提高性能和可用性。

- **负载均衡器**：使用负载均衡器（如Nginx、HAProxy）将请求分发到多个服务器，避免单个服务器过载。
- **分布式系统**：设计分布式系统架构，将工作负载分布到多个节点。

### 容错设计

容错设计（Fault Tolerance Design），旨在使系统能够在某些组件发生故障时仍然继续正常运行，它的核心理念是通过冗余和其他技术手段，避免单点故障导致系统整体失效。

以下是容错设计的一些具体方法和技术：

- **无状态服务**：设计无状态服务，使得服务实例可以随时被替换而不影响整体系统。
- **数据复制**：使用数据复制技术（如数据库的主从复制）保证数据的高可用性。

### 定期维护与更新

在现实生活中，不管是人的健康还是机器或者其他的健康，都需要定期维护，对于系统来说也是一样的道理，通过定期的维护和更新，可以及时发现和解决潜在问题，防止系统故障，提升系统的整体可用性。

以下是定期维护与更新的主要策略：

- **补丁管理**：及时应用安全补丁和系统更新，防止已知漏洞被利用。
- **健康检查**：定期进行系统健康检查，发现潜在问题并及时修复。

### 使用高可用性云服务

- **云服务提供商的HA解决方案**：利用云服务提供商提供的高可用性解决方案，如多区域部署、自动故障转移等。

### 网络优化

- **冗余网络连接**：配置冗余的网络连接，避免单点网络故障。
- **优化网络配置**：使用CDN（内容分发网络）加速内容交付，减少网络延迟。

### 总结

可用性是我们在做系统设计时一个重要指标，它确保用户可以可靠且持续地访问服务。因此，我们可以结合真实的业务需求，在上面提供的一些技术策略中灵活选择。

### 学习交流

如果你觉得文章对你有帮助，请帮忙 [点赞](#) + [收藏](#) 。关注 [猿java](#) ，查看更多原创技术文章。



本文收录于以下专栏



技术干货 专栏目录

技术干货，硬核文章

94 订阅 · 107 篇文章

订阅

上一篇 Netty 是如何解决半包和粘包问题？

下一篇 Kafka ACK机制详解！

评论 4



[登录 / 注册](#) 即可发布评论！

最热 | [最新](#)



Coding小菜鸡 Software Developer

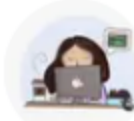
感谢，学习了，受益颇多

6月前 👍 点赞 💬 1 ⋮



猿java [作者](#)：谢谢支持，常来

6月前 👍 点赞 💬 回复 ⋮



女程小v @字节

一更新就推送过来了，6

6月前 👍 点赞 💬 1 ⋮



猿java [作者](#)：常来

6月前 👍 1 💬 回复 ⋮

为你推荐

系统设计，建议做好这 10 个权衡！

猿java | 7月前 | 👁 608 👍 16 💬 3

后端 设计模式 Java

面试官：如何构建一个高可用的系统？

托尼学长 | 1年前 | 👁 4.5k 👍 27 💬 4

后端 面试 架构

无缝切换在线升级的终极探索

蒋老湿 | 4年前 | 👁 3.0k 👍 16 💬 2

架构

让我们聊一聊分布式事务

JAVA大贼船 | 4年前 | 👁 771 👍 5 💬 评论

分布式

架构权衡评估方法（ATAM）：如何评估一个系统的质量

JAVA前线 | 3年前 | 👁 2.5k 👍 9 💬 1

架构 后端

工作十年，在腾讯沉淀的高可用系统架构设计经验

腾讯云开发者 | 1年前 | 👁 3.7k 👍 53 💬 4

架构

如何避免单点故障？

猿java | 4月前 | 👁 482 👍 17 💬 评论

后端 Java 面试

稳定性建设 -高可用系统建设的必备知识-下

梦尘啊 | 10月前 | 👁 527 👍 3 💬 评论

后端 架构

开发者视角：如何保证RabbitMQ的高可用和可靠性

旧梦呀 | 10月前 | 👁 131 👍 2 💬 评论

Java

对高可用系统的一点理解

ksfzhaohui | 3年前 | 👁 576 👍 1 💬 评论

Java 架构

什么是分布式缓存？

猿java | 4月前 | 👁 370 👍 19 💬 评论

后端 架构 面试

如何设计一套分布式任务调度系统？

猿java | 5月前 | 👁 655 👍 20 💬 3

后端 架构 面试

架构师养成记（二）架构设计理论

李少博 | 1年前 | 👁 1.5k 👍 11 💬 评论

后端 架构

Eureka 高可用服务集群搭建

大飞学习笔记 | 1年前 | 👁 1.2k 👍 2 💬 评论

后端 Spring Cl... 架构

探讨打造「高可用架构」秘籍

京东云开发者 | 7月前 | 👁 378 👍 10 💬 1

后端

