应用保护

系统压力太大，负载过高。

数据库慢查询，

核心思想：优先保证核心业务，优先保证大部分用户。

降级

论坛：

思想：丢车保帅。保证核心业务可用。

1。系统提供后门接口。

2。独立降级系统。对1封装。redis。

if(ifxxList){

getXXList();

}else{

getXXlis备用（ ）；

}

前提：代码提前规划好。

自动开关降级

提起写好降级逻辑。

触发降级的条件：

1。超时。阈值。

2。异常。 阈值。 1001 运行时异常。

3。限流。排队页面，卖光了。

降级手段：

减少不必要的操作，保留核心业务功能。

停止读数据库，准确结果转为近似结果。静态结果（猜你喜欢），同步转异步（写多读少）。功能裁剪（推荐干掉）。禁止写（高峰期减少不必要的写）。分用户降级。工作量证明 POW（验证码，数学题，拼图题等，滑块）。

熔断

A服务：X功能（熔断 ）--B T。

阈值怎么定？5次/1s--3次，先预估，上线观察，最后调优。

保险丝。

熔断策略：

根据请求的失败率：熔断开关 打开。过一个时间窗口（10s），熔断开关关闭。开关再打开。--半开。Fail Fast。

hystrix：yml。每20个请求，当失败率50%，打开开关，等过5s，开关关闭。

根据响应时间：sentinel。时间阈值，超过 熔断开关打开。

限流

外部（请求太多了）、内部（资源）。

基于请求的限流

请求的总量。

直播间超过100人，出去。

限制时间量。

一个时间窗口内：接100个请求。

上线观察、调优。

1s 1万人。结果 5k人就扛不住了。

提前压测。

排队：缓冲。MQ，长连接给用户响应。

基于资源的限流

连接数，线程数，请求队列。cpu参数。

难点：确定关键资源，阈值。

池化技术：连接数，线程池。

队列大小：请求队列。10个。

cpu参数：perl。

tomcat:

<connector>

acceptCount: 如果tomca线程池满了，

maxConnections：最大链接数。

maxThread：

</connector>

固定时间窗口

滑动时间窗口

突刺。

漏桶算法：

漏桶的队列长度。

漏桶里有请求等着处理，但是我的出口，确实处理不了。

一个是浪费。

O(1)。

令牌桶算法

漏桶和令牌桶

漏桶：限制平滑的流入：

令牌桶：可以容忍部分突刺。

节流

去抖。 减少网络的请求呢？

定义一个时间窗口。1s 开源。

前置：a记录的转发，cdn。缓存扛不住） 限流。

请求来-中间环节--处理完。

（应用保护）

序列化（字节流），对象。

隔离

数据隔离：数据重要性排序。分库。

机器隔离：给重要的用户单独配置服务器。用户的标识去路由。

线程池隔离：线程池分配。hystrix。

信号量隔离：计数器。hystrix。

集群隔离：服务分组（注册中心），秒杀。（单独分出一组服务给 核心业务）

机房隔离：3个服务。局域网ip、路由。

读写隔离：主从。create update delete \ select (数据延迟，前面的课里讲过)

动静隔离：识别动静态数据，进行隔离。nginx，apache。

爬虫隔离：openrestry user-agent。 对ip的访问频率。

冷热隔离：秒杀，抢购。读：缓存。写：缓存+队列。

恢复

撤出限流，消除降级，关闭熔断。

半开关。

让吞吐量爬升。缓存预热。

1。类实例创建，反射调用创建实例。

2。缓存数据预热。

灰度发布。限流逐步提升：阈值。用户打标签（18以下用户10个，18以上 1万人。）

异地多活

异地

多活（不是备份）

1。请求某一个节点，都能正常响应。

2。某些系统故障，用户访问其他系统也能正常。

分类：

同城异区，跨城异地，跨国异地（隔离）

跨城异地：质变：RTT。Round Trip Time 50ms。京广。 京沪：30ms。 **数据不一致。**

跨国异地：延迟，已经无法让系统提供正常服务。

负载均衡。---dns cdn。分流。

数据不一致问题：

1。保证核心业务多活。

用户系统：注册，登录，修改用户信息。 广州 北京。

注册：

登录：

修改用户信息：根据时间合并。分布式ID。

1000万用户。每天注册 几万个。修改用户信息的呢 几千个。 1000w登录。

基于已有的数据。上天总是公平的。

2。保证核心数据最终一致性。

自建网络。国字头。

session数据。token。（量大，同步不值得。）

同步手段：

中间件的主从复制。mysql ，redis。

消息队列：订阅发布。

二次读取：

回源读取：

重新生成：

保证大部分用户：小部分用户忍。

异地多活设计步骤：

1。业务分级。

微信聊天和朋友圈。

广告收入。新闻网站。

公司、用户影响。

2。数据分类。

唯一性：手机号。额外存储。crud。

实时性：

可丢失性：session

可恢复性：session可恢复的？

3。 数据同步。

4。异常处理。

多通道同步：mysql主从复制+消息队列。两种方式走不同的网络。

同步和访问结合：

广州，北京。

A用户登录广州，生成 session，sessionID（带有广州气息的sessionID xxxxx1）。

session不同步给北京。

A访问北京的服务器：参数 sessionID（带有广州气息的sessionID xxxxx1）。

if(endWith(1)){

String session = http.调用（拿对应用户的session）；

sesson.set(K,V);

}

javaer。

j2cache，nginx+lua

OVER!!!

方案性东西。找我讨论。

从0到1 的代码课，商城。

重试，持久化，重发。

迭代。

代码量不够：

system.out.println("hello world")。