**过载保护技术方案**

# 综述

过载是一个比较宽泛的词，可以指电气设备负载过大，也可以指物体承受的作用力过大。对这些超出“负荷”的行为做出的保护，统称过载保护。

软件架构设计中所指的过载保护，是指当系统负载超过一个软件系统的处理能力时，如果不对软件系统进行保护，可能导致软件系统对外呈现的处理能力表现为0，且在负载降低后，不能自动恢复的现象，未实施过载保护的软件系统处理能力示意图如下：

未实施过载保护示软件系统处理能力意图

我们期望通过增加过载保护系统，可以保证软件系统在负载不断增加直至超过其最大负载能力的情况下，软件系统可以保持一个稳定的最大处理能力的最佳状态，不产生“掉底”现象，实施过载保护软件系统处理能力示意图如下：

实施过载保护软件系统处理能力示意图

# 负载监控

## 监控对象

随着行业的发展，越来越多的系统实现了分布式集群，这些集群正在逐渐迁移到IaaS平台之上，形成了如下的部署结构：

现代软件系统的整体部署结构

基于现代软件系统的整体部署结构特点，我们需要对负载进行监控的对象包括了应用、中间件、虚拟机、物理机。

### 应用

应用系统是功能实现的载体，是整个软件系统的核心，也是负载监控的重心。对应用系统的负载监控可以集中在RPS（Requests Per Second）指标的监控上，即监控应用系统的每秒请求数量，对应用的监控要能够估算出应用的最大RPS。

应用最大RPS的最佳衡量方式是在生产环境进行压测，但由于生产环境及其重要，生产环境是否稳定直接决定了是否可以提供良好的用户体验，因此不能过度依赖于生产环境，应用最大RPS的最佳衡量可以采用基准测试和在线压测两种方式进行，通过基准测试提供评估，通过在线压测进行误差调整。

**基准测试**：需采用某型号服务器（与生产环境相同最佳）进行部署、测试，以便获取在不同请求数量下，应用系统的RPS表现。测试完成后，可根据测试结果及服务器配置，对生产环境或其他环境应用系统的RPS进行估算。

**在线压测和监测**：基准测试侧重于估算，估算存在误差，误差过大可能会导致RPS失真，进而影响过载保护系统的正常运行，因此，在基准测试的基础上，需要进行生产环境在线压测和监测，以衡量估算最大RPS的准确度，降低误差，对系统过载的判断提供有力依据。

### 中间件

中间件作为应用系统运行的容器，提供了应用系统运行的环境，其影响应用程序负载的因素较多，主要包括线程数量、连接数大小以及内存大小。

为了防止线程数量无限增长，大多数中间件的实现上都会限制最大线程数量，负载的增加并不会导致线程数量无限增长，而线程数量接近或达到最大值并不一定意味着中间件过载，因此并不适合使用线程数量来作为衡量中间件的负载的参数。

对于高并发web应用系统，系统的并发数量同时受限于可以建立的连接数，Http协议基于TCP实现，如果中间件作为连接的发起方（即作为客户端），受TCP协议的限制，其能够发起的最大连接数理论上不会超过65535个，而实际上中间件作为客户端发起的连接较少，并不会达到其理论限制。如果中间件作为连接的接收方（即服务端），其能够建立的连接数受限于所属操作系统的限制，以Linux为例，其能建立的连接数受限于单一进程能打开的最大文件数（使用命令ulimit –n可以查看），该参数可以调整，调整后的数值大小受限于/proc/sys/fs/file-max，这些参数值调整后可以打开数万的文件，即理论上单节点中间件至少可以支持建立数万的连接，资产管理系统并没有这么高的并发要求，因此不适合使用连接数作为系统衡量负载的参数。

随着中间件的负载增大，其所消耗内存的数量不断上升，当所需内存超过其分配内存后，便会发生内存溢出错误，大量的内存溢出错误的产生，会导致中间件运行变慢、线程假死等一系列性能问题，严重时会导致中间件崩溃，因此，可以采用中间件的内存消耗情况来衡量中间件的负载情况。

### 虚拟机

随着虚拟化技术的发展，虚拟机技术得到了充分的应用。虚拟化技术的发展带来了很多的便利，透明性就是其中非常重要的一项。应用程序在虚拟服务器中运行，无法感知亦无需关注所在环境是虚拟服务器还是物理服务器，因此，对于虚拟机的负载监控与物理机无异，包括了对CPU、内存、网络IO。

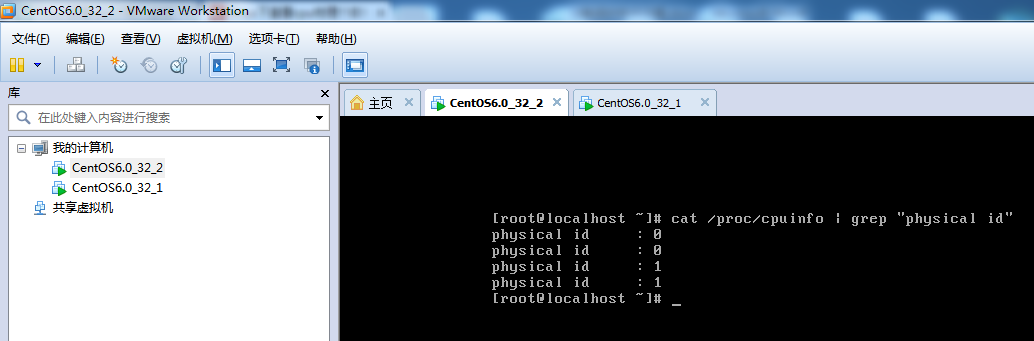
### 物理机

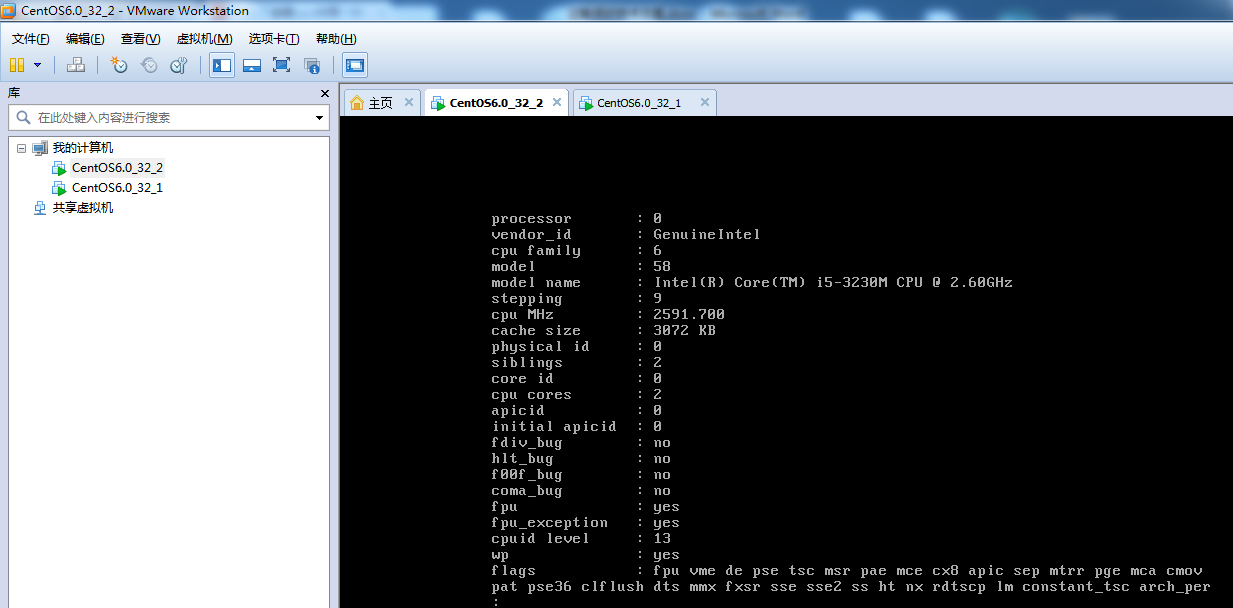
既然有了对虚拟机的监控，是否还需要监控物理机？

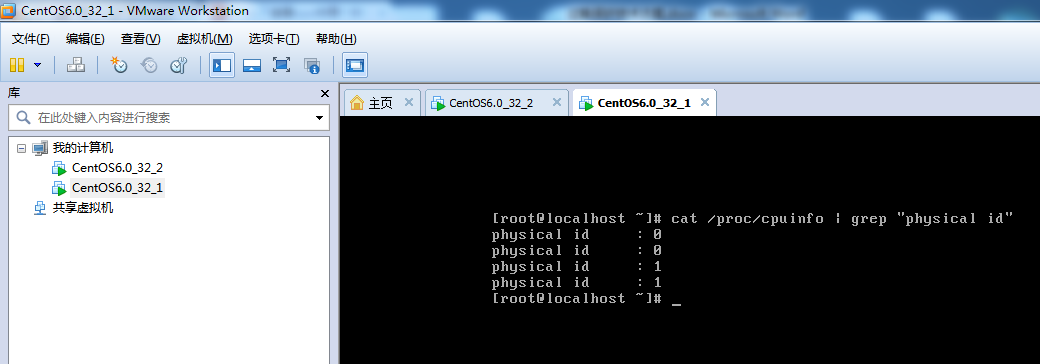
虚拟化的技术实现上并没有限制虚拟设备的性能要严格受限于物理设备的性能，以DELL E5430为例，其物理设备性能如下表：

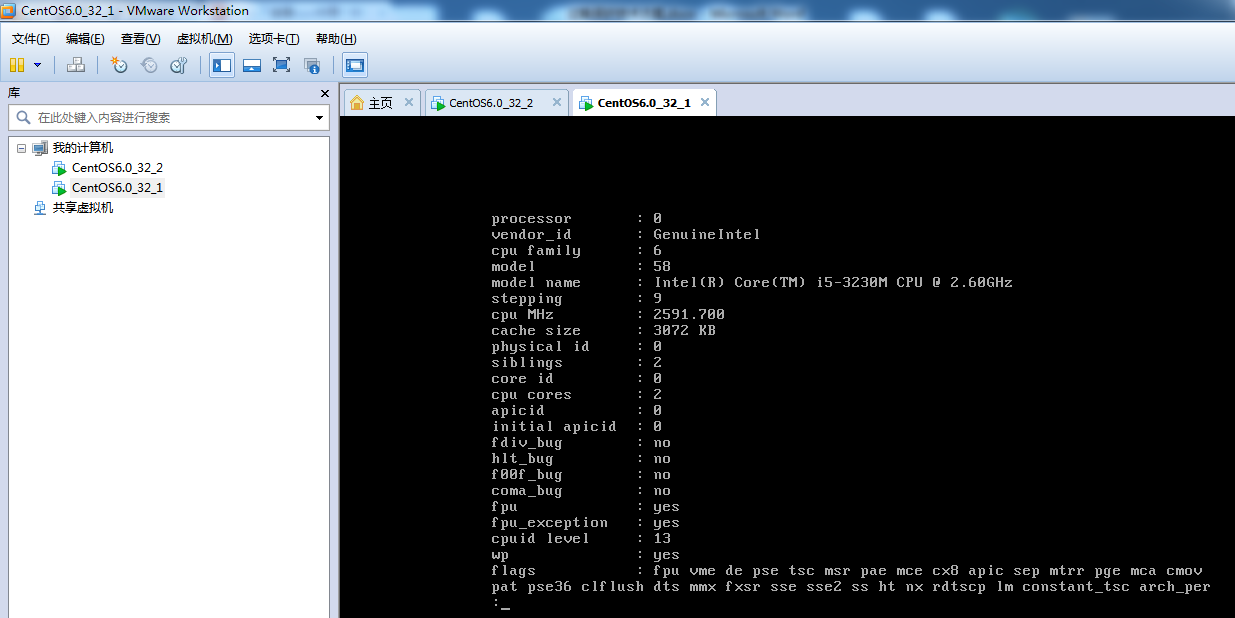
|  |  |
| --- | --- |
| 设备 | 型号 |
| CPU | Intel® Core™ i5-3230M 2.60GHz 双核 |
| 内存 | 4G+4G |
| 磁盘 | 512G |

通过VMWare虚拟化技术，在一台DELL E5430计算机中，构建了2台虚拟计算机，其单台虚拟机CPU为2颗双核2.60GHz CPU，两台虚拟机共计4颗双核2.60GHz CPU，如下图：









两台虚拟计算机可以同时启动，其虚拟设备性能已经超过了物理设备性能，因此，两台虚拟机设备上的CPU，实际上并不可能实现全部满载，即当负载不断增加的同时，CPU负载永远无法同时达到100%。

因此，仅仅通过监控虚拟设备的负载情况，并不能真是反应出系统整体的实际负载情况，监控虚拟机的同时，需要同时监控物理机。

对于物理机监控，主要包含了CPU、内存、网络IO三项指标。

## 过载评估

对于过载的评估，可以采用阀值模式，其阀值可设计为2级阀值，第一级阀值为预警阀值，第二级阀值为过载阀值，通过两级阀值，再加上过载临界点，将系统的处理能力划分为四个区域，分别为：安全区、警戒区、保护区以及过载区。

安全区：被保护系统健康状态良好，系统在处理能力方面仍有较大剩余空间，在此区域的系统，过载保护系统不采取任何动作，不干扰被保护系统的运行。

警戒区：被保护系统健康状态良好，但负载较高，在处理能力方面剩余空间较小，在此区域的系统，过载保护系统不干扰系统的运行，但会向运维人员发送警告信息。

保护区：被保护系统健康状态较好，但负载已接近满载，处理能力方面几乎没有过多剩余空间，在此区域的系统，过载保护系统将启动保护机制保证被保护系统不产生过载，同时，定时向运维人员发送告警信息。

过载区：系统产生过载，产生过载的可能原因包括：（1）过载阀值设置过高，未出发过载保护；（2）已经出发过载保护，但保护机制未完成启动；（3）由于流量控制等原因，导致过载保护机制失效；在此区域，过载保护系统不发送邮件，需要人工介入，重新评估过载保护系统的设置。

## 过载预测

系统产生过载，可能会发生“掉底”现象，过载保护的核心目标即为保证系统不发生过载，因此需要在系统过载前启动、并完成过载保护机制加载。

在系统并未启动过载保护机制的情况下，系统负载及处理能力运行曲线如下：

从上图可以看出，系统会在时间点T2发生过载，根据过载阀值的设计，过载保护系统需要在时间点T1启动过载保护机制，并在时间点T2之前完成过载保护机制的启动，即过载保护系统必须在T2-T1的时间内完成保护机制的启动。

假设过载保护机制启动时间为T，那么显而易见，系统需保证T2-T1>T，由于保护机制启动的时间T是固定的，而T2受限于RPS的增长速度和应用最大RPS，故为了保证过载保护机制的完成，通常需要尽可能的调小T1，即尽可能降低过载阀值。

过载阀值设置过低，可能会引起过载保护机制过早启动，使得保护区增大，在负载机制完成后，系统一直处于较低负载，但系统负载仍高于过载阀值，使得过载保护长时间不能恢复，进而造成一定程度的资源浪费，如下图所示：

过载保护应在不降低过载阀值的基础上尽可能早的启动，因此，需要采用过载预测的方式，提前预测出系统在未来一段时间内的负载情况，以便提早决策是否需要启动过载保护，为过载保护机制留有更多的启动时间。

过载预测可以采用负载历史数据分析和用户行为分析两种方式。

**负载历史数据分析**：系统运行的负载情况往往带有很强的规律性，每年超过80%的时间都在规律性地运行，例如，通常企业应用系统的负载情况，与企业的作息时间存在着密切的联系，如下图，负载高峰产生在早上和下午上班后的一段时间：

常见的做法是衡量负载数据的正态分布，值得注意的是，虽然系统运行的负载情况具有规律性，但这种规律中包含着异常，这类异常具有偶然性，属于小概率事件，可能由于某些业务事件（例如电商系统临时组织的秒杀活动）产生，也有可能来自于硬件的可靠性（例如网络风暴等）或其他不确定因素，而这类异常信息并不能作为过载判断的依据，因此，需要排除此类异常信息。

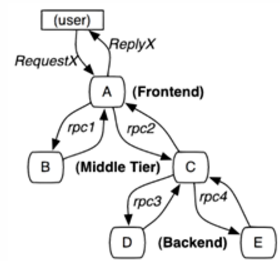
对负载历史数据的分析，并排除运行期负载异常信息后，可以得到相对准确的负载正态分布图，以负载正态分布为基础，对现有数据进行分析，预测出未来一段时间可能产生的负载情况。

由上图可知，系统负载在时间点t1发生偏离，偏离逐渐增大，对近期的负载数据进行采样分析，若负载增速并未放缓或进入负增长状态，可以预测系统负载在未来一段时间内即将发生过载，因此在时间点t2即可启动过载保护机制，相对于没有过载预测的功能，可在不降低过载阀值的情况下，增加t3-t2的时间，来启动过载保护。

**用户行为分析**：通常，业务具有较强的关联性，以电子商务购物为例，通常会经历三个过程：浏览选购——加入购物车——付款，各业务环节间存在着一定的转化率，例如，一个商品在被100人次浏览后，会有10次加入购物车，即商品浏览到加入购物车的转化率为10%，而加入购物车的10件商品，会有2件商品会进行付款，即加入购物车到付款的转化率为20%，进而可以看出，浏览选购到付款的转化率10%×20%=2%，因此，可以通过不同业务的负载监控，来推断出其后续关联业务的负载变化，例如：当商品浏览的请求增加了100%时，可以近似推断，付款服务未来一段时间内，可能会增加2%的请求。

由于实际的业务场景比较复杂，并不像上例中如此简单，服务中存在着各种判断条件和分支逻辑，因此各服务之间的关联度分析，如果采用人工分析方式，复杂度高、工作量大、准确性低。

因此，可以在应用系统中建立全链路应用监控系统，通过日志分析，分析出调用链路，进而完成各服务间关联度分析，同时，还可以通过全链路监控日志，统计分析出各关联服务间的调用比例，为过载预测提供有效的数据支撑。



# 保护机制

常见的过载保护机制是动态扩容，动态扩容可以使得系统处理能力随着系统负载的增加而增加，很大程度保证了系统的可用性和可靠性，但扩容受限于现有技术限制和系统的运行环境，并不能保证随着系统负载的增加而无限制扩容，因此，在动态扩容的基础上，需要增加服务降级机制，保障服务在过载情况下能够运行良好。

## 服务降级

降级服务的划分策略可以包含按照业务功能划分以及按照用户划分两种方式。

**按照业务功能划分**：识别出核心业务功能（服务），优先保障核心功能（服务）的可用性，对于可用性要求并不高的业务功能或服务请求，以错误页面、错误或者Mock数据的方式返回。在过载保护机制中，应保证请求不到达被保护对象，以减少此类已经降级的服务请求给被保护对象带来的冲击。

按照业务功能划分有效地识别了核心业务（服务），从业务的角度出发保证了核心业务（服务）的高可用性，但对于用户范围并没有有效的控制，即过载保护机制的启动会影响到全部用户，这也带来了按照业务功能划分的优缺点，其优点是能够保证所有用户对于核心业务（服务）的使用，其缺点是被降级的业务（服务）对于所有用户都不可用。

**按照用户划分**：识别出系统的关键用户，优先保障核心用户对系统的使用，提供友好的等待提示界面展现给暂时无法使用系统的非核心用户。

按照用户划分的策略，其优点是能够保证核心用户的使用，其缺点是会导致普通用户完全无法使用系统，这其中也包括系统中的核心业务。

无论是按照业务功能划分，还是按照用户划分，都不是万能钥匙，很多情况下我们既需要保障核心用户的使用，也要保证关键业务的运行，因此往往需要在服务降级的设计上组合两种降级策略，做适度的取舍，以便能够兼顾两种策略的优点。

## 动态扩容

随着虚拟化技术的发展，动态扩容变得更加成熟稳定，目前动态扩容技术广泛包含在云平台中。动态扩容可以在很大程度上保证系统在高负载情况下的可用性，同时使得系统具有更高的资源利用率。

## 流量控制

# 技术实现

## 整体架构

## 负载监控

## 保护机制

### 服务降级

### 动态扩容

## 过载告警