各位评委好，我今天将要讲述的项目的主题是基于可编程交换机的高效键值存储监控系统，接下来是项目介绍。

首先是项目背景。如今，键值存储系统是社交网络等在线服务平台的重要基础，而这些在线服务对键值存储的服务水平目标有着严格的要求，满足其性能的一个重要挑战是热点问题，即一小部分受欢迎的项相较于其他项被服务不成比例地访问。想要解决热点问题，首先要找到热点的位置，那么如何进行热点感知、对整个系统进行监控就显得尤为重要。

其次是问题挑战。想要进行热点监控有两大需求，分别是可扩展性即面对大量请求，能够迅速检测热点；和通用性能够检测不同的工作负载类型和过载位置。目前已有的方案仍存在着缺陷，如主机端遥测方案缺乏对交换机的可见性，而基于通用CPU方案存在着性能瓶颈。并且热点问题越来越严重且昂贵：每100毫秒的加载延迟会导致亚马逊销售下降1％，而Google搜索每增加500毫秒的加载延迟会损失20％的流量。我们发现，交换机具备天然的机架级可见性，所以我们利用交换机在数据平面聚合数据，使得可见性问题得以缓解，同时可编程交换机也优化了系统的性能。同时存在着三大挑战即计算资源有限、通道带宽有限和存储资源有限的问题。我们在本项目中利用sketch数据结构来压缩监控数据、将检测工作负载类型抽象成任意键查询、提出一些硬件原语进行整体性能资源的优化。

然后是整体架构，如图所示，这是SpotMon整个系统运作时的工作流程图。整体由数据平面模块、核心算法模块、控制平面模块和多功能热点查询模块组成，接下来会详细的进行讲解。

首先是数据平面模块，这个模块首先利用可编程交换机实时监控通过网络传输的数据流量。然后使用流量采样技术，对数据流进行抽样和聚合，最后将处理得到的数据传递给核心算法模块。

其次是核心算法模块，这是我们实现快速热点监控的关键。这张图就是核心算法如何处理数据的过程，在其中，我们利用了随机方差最小化理论，实现了多类型负载感知。它主要分为了"heavy part"和"light part"两部分。"heavy part" 主要负责处理复杂、耗时较长的请求，即访问频率较高的项。由于这些请求需要更多的处理时间和计算资源，核心算法通过将它们分配给专门的处理器或核心来确保它们得到适当的处理。这些处理器通常是性能较高的核心，可以处理更大的负载。相反，"light part" 负责处理简单、快速的请求，即访问频率较低的项。这些请求通常只需要少量的处理时间和计算资源，因此可以使用较低性能的核心来处理。通过这个数据结构，可以在保证快速响应的同时，充分利用系统资源。

然后是控制平面模块，控制平面模块通过接收查询的信息，从核心算法模块中聚合到的数据中进行检索，并最终得到结果，返回给管理员进行实时的监控。

最后是多功能热点查询模块，这个模块为管理员提供了多类型负载感知查询的接口。操作员可以根据需求查询不同类型的热点负载信息，从而实现精细的负载监控和管理。在其中实现了一种分级查询算法，即先查询超载位置、然后查询热点项目、最后识别工作负载，如图所示。算法从粗粒度到细粒度逐步进行查询，避免了不必要的查询操作，从而提高了查询的效率和精确度。

然后是具体应用，通过前面的部分，我们已经可以监控整个系统的性能。那么在发现热点后，需要我们迅速的进行热点缓解。我们在这里利用贪心算法，即将利用率低的通道、服务器、进程调整为优先级最高，和清除缓存中的写入密集型项目。通过这些优化手段，可以提高系统的吞吐量和响应时间。

接着是环境搭建，我们整个项目使用了两台服务器和一台Tofino交换机，利用一台服务器作为客服端生成键值查询，并将请求发送给另一台作为键值存储服务器的机器上，利用容器化技术增加节点个数。我们在这里也将整体的网络拓扑图进行了勾勒，本项目使用了一个k=4的fat-tree结构，如图所示。右侧图示为交换机。

接着是实验测试，在本页上的四张图分别展示了SpotMon消耗有限的硬件资源、SpotMon消耗有限的带宽、SpotMon可以处理高请求率以及SpotMon对重要负载有高准确性。这页的四张图展示了通过测试召回率、准确率、F1分数和平均相对误差，得到了SpotMon在不同的内存预算下保持低错误率的结果。这页的四张图也是通过检测同上页的四个数据指标，得到了SpotMon在不同负载偏度下保持低误差的结果。这一页说明了SpotMon通过O1-O4提高了系统吞吐量、SpotMon在不同的写率下保持低错误率和SpotMon具有高度的可扩展性。

最后是研究意义，其学术价值在于：1、提出了快速和全面进行热点缓解的系统。2、推动了可编程交换机和键值存储系统的研究。3、为数据中心网络的设计和优化提供了借鉴。其社会价值在于：1、支持了关键业务服务的性能优化。2.降低了系统管理和调优的复杂性。3、为未来网络优化提供了新思路。本项目总体上实现了对于热点进行缓解的监控系统，相关的专利也处于审核阶段。前端的部分因为时间原因，在另一个视频中进行了详细演示。感谢您的观看。

1. 本项目解决的问题是什么？

在键值存储系统中的热点问题。

1. 可编程交换机为什么可以优化系统？

首先是可以实现软件无法达到的低延迟，同时自身具备着一些计算性能，可以缓解整体的压力。

1. 本项目的创新点是什么？

我们利用了可编程交换机来进行网内的监控。

1. 本项目的应用价值是什么？
2. 对交换机的可见性？

交换机中也会存在过载位置，如各个端口，寄存器热点问题，而主机端则无法具体观测到这些过载位置。

1. 能不能通过主机测量交换机的信息？

无法做到，无法观测到各个端口，寄存器热点问题，而主机端则无法具体观测到这些过载位置。

1. 什么是多类型负载感知，包括哪些类型？

键值对本身为内容负载，键值对的长短和键值对的处理操作：读写。

1. 能检测哪些过载位置？

服务器内的过载位置、交换机中的各种位置。

1. 具体应用的讲述修改