基于RapidIO的低功率、低延迟服务器和存储网络

近来，作为云中数据中心采用的传统架顶式系统的替代系统，刀片式及微型服务器和存储系统已经受到了极大关注。典型云基础设施中的服务器和存储系统要求更易于扩展、能高效率地实现虚拟化并需要互连延迟很低的高性能多核处理器，以在确保低延迟且安全的用户体验的同时，提供可靠的低功率系统解决方案。

Mohammad Akhter

刀式架构具备机架内交换板和更少的上行链路电缆，相对于架顶式系统而言，成本更低、可靠性更高。另一方面，相对于刀片式解决方案，微型服务器和存储系统占用空间更少，功耗更低。视处理器内本机 I/O的架构和可用性的不同而不同，微型服务器和存储系统在提供更高的可靠性、可扩展性和安全虚拟化的同时，延迟可以极大地降低。

服务器和存储架构一般采用以太网、PCIe和InfiniBand互连技术。尽管（具备TCP/IP）的以太网主要用于传送网络流量，但是通过采用融合以太网，该技术也可用来传送混合型流量（例如存储、网络和计算流量）。然而，在这类传送混合型流量的系统中，由于采用了计算密集型协议栈，所以延迟和功耗很高。由于采用基于帧的拥塞管理机制，而且以太网不能中断大型帧传送，所以以太网的QoS受到了极大影响。

另一方面，PCIe和InfiniBand协议主要用于计算和存储流量。PCIe互连没有本机消息机制支持，且依赖单根分层架构，因此在可扩展性方面受到限制。而基于InfiniBand协议的服务器和存储系统缺乏本机处理器支持，会遭遇较高的系统成本和端到端延迟的问题。

云应用举例

刀片式及微型服务器架构可使很多应用受益，例如在云中进行的、时间受限的大数据分析，就能受益于这类架构。这类应用需要支持与网络、存储和计算有关的混合流量。为了确保更高的QoS，这类应用还需要在给定时间段内，完成全部大数据集的处理工作并实时传送数据流。人们一般用具备HBase的Apache Hadoop框架来完成大数据分析任务。

Hadoop-HBase框架有3个组件：MapReduce（计算）、HBase（存储）和互连交换矩阵。在这类框架中，通过网络、跨服务器集群并行加载与MapReduce引擎有关的数据并执行相关的数据分析任务，视数据量的不同而不同，服务器集群需要具备可扩展性和高效互连能力。在这类框架中，Map和Reduce功能用来执行计算任务，而HBase则用来处理对网络上大型分布式存储节点的快速随机存取。就大量并行工作的Mapping引擎而言，大的数据集首先被划分成若干较小的数据集。之后，Mapping引擎将数据转换成中间格式（与“密钥”配对），并将数据传送给最靠近这些计算节点的存储系统。

数据加载和变换完成以后，来自不同存储节点的输出数据就被完全打乱，并基于数据“密钥”，通过网络传送给Reducer服务器。主服务器确保基于中间数据的位置向Reducer服务器分配数据。一旦Reducer完成处理任务，就对数据进行整合，以产生最终结果，这些结果通过网络回写到存储服务器内的一个或更多的输出文件中。

在这种框架中，软件基础设施能受益于具备高效率互连的服务器和存储架构，以在给定时间段内完成处理（计算和存储系统存取）任务。例如，MapReduce模式能利用低延迟基础设施，跨大规模可扩展计算和存储节点，并行加载和执行任务。这类框架具备卓越和基于硬件的容错、错误恢复及低延迟同步功能，还可减少与系统管理及可靠性（在MapReduce和HBase中）有关的软件开销。总之，在这类框架中，处理过程对数据中心互连提出了以下要求。

本文以下章节探讨的RapidIO协议的各种功能支持上述以及很多其他与云中服务器及存储应用有关的要求。

处理步骤	对互连技术的要求
任务划分、分配以及数据跟踪和监视	可靠、可扩展的低延迟互连，以通过恰当的流量控制，控制、跟踪和管理任务及数据。
数据加载	在存储系统和处理节点之间提供可预测和可靠的低延迟数据传输。
数据复制	具备多点传播功能的可靠互连，以支持初步和定期的数据复制。
Mapper 输出处理	基于“密钥”、以低延迟向 Reducer 服务器重新分配数据。
Reducer 输出处理	低延迟数据传送，以产生最终结果，并回写存储系统。
容错和可靠性	容错网络，以减少与可靠性有关的软件开销。
融合流量	为存储、计算和网络流量保持卓越的 QoS。

■表1：处理过程对数据中心互连提出的要求

■表2：适用于刀片式和微型服务器的关键RapidIO协议功能。

属性	RapidIO 协议功能
延迟	➔ 凭借确定性的数据包交付，实现有保证的最低应用至应用延迟（交换机延迟约为 100ns）。 ➔ 控制符号提供延迟最低的控制和同步。
成本和功率	➔ 成本最低的交换机和 NIC（约 2W/NIC、10W/交换机）。
可扩展性	➔ 很容易扩展至数千个节点和数百万路数据流。
QoS	➔ 通过高效率报头、数据包格式、流量控制和拥塞管理，实现卓越的 QoS；延迟最低的短控制符号、VOQ-BP 和数据流传送流量管理。
容错/可靠性	➔ 快速检测和隔离故障情况。 ➔ 在硬件中有序、可靠地交付——更低的软件开销。
吞吐量/拓扑	利用卓越的交换矩阵和高速协议，实现 I/O 带宽与处理器性能的匹配。 ➔ x1、x2、x4、x8、x16 通道。 ➔ 高达 100GBaud/端口（第二代）。 ➔ 超过 160GBaud/端口（第三代 10 x N)和 400GBaud/端口（25 x N 路线图）。 ➔ 任何拓扑——圆环形、网状、树状、超立方体形，等等。
混合流量	以卓越的 QoS、通过高效率数据包格式和基于短控制符号的流量控制，支持计算、网络和存储流量。支持 VC。
数据复制	➔ 通过 RapidIO 交换矩阵、利用 RapidIO 多点传播功能，支持单点传播、多点传播和广播。
安全性	➔ 安全的 Queue-Pairs、通过加密实现的端口级安全性以及通过滤波实现的协议级安全性。
虚拟化	➔ 基于硬件的轻量型协议栈允许直接共享来自应用的 I/O 资源，简化了 I/O 虚拟化。低延迟、高吞吐量网络允许更容易地进行 VM 迁移。

RAPIDIO概述

2002年推出的高性能RapidIO协议是一种基于数据包的开放数据通信标准。自推出以来，全世界大量设备制造商和芯片供应商交付了数百万套基于RapidIO的设备，以满足3G/4G无线基站、视频服务器、军事通信、嵌入式处理和高性能计算的互连需求。在这些应用中，有些受益于延迟最低的RapidIO的可扩展交换矩阵架构，而另一些则利用该协议确定有保证的数据交付、容错和可靠性特点，连接了大量多核CPU。

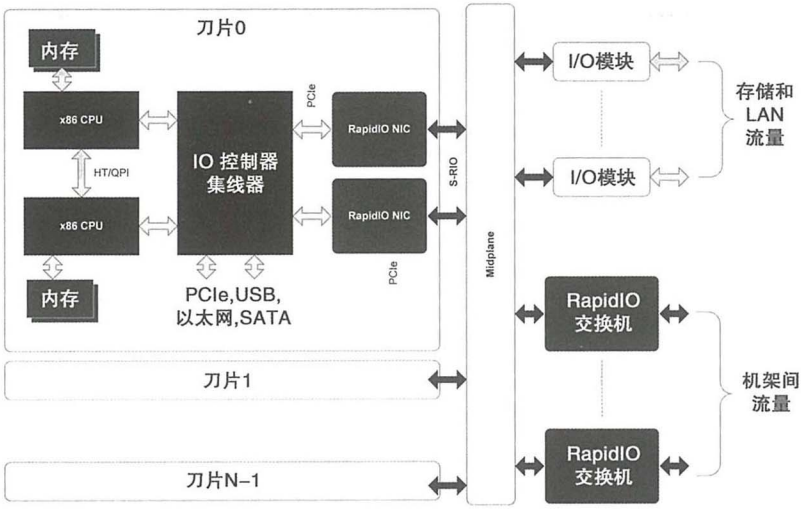
RapidIO协议和数据包格式是在3层分层架构中规定的。该协议支持电路板内或跨电路板的短、中及长距离链路。该标准还支持光纤和电缆链路。

就管理和控制而言，RapidIO协议支持全面的系统bring-up、互操作性、多点传播、错误管理和卓越的故障恢复机制。由于简化的分层架构，所以有可能在硬件中实现协议栈，同时保持总体软件开销、系统成本和功耗很低。表2总结了适用于刀片式和微型服务器架构的关键RapidIO功能。

RAPIDIO用于刀片式及微型服务器

利用卓越的RapidIO功能，有可能设计出使用RapidIO兼容NIC及交换机的x86刀片式或微型服务器（图1）。还有可能设计出使用基于RapidIO的ARM处理器或其他低功率处理器的微型服务器（图2）。凭借集成的RapidIO，有可能在极大型服务器和存储集群中，在应用之间提供最低的系统延迟。凭借低延迟、高吞吐量互连，可以跨计算和存储节点执行并划分应用，就如同应用处于同一位置一样。此外，凭借可靠的容错能力，有可能改善系统可用性，并减少与系统内同步、控制和监视有关的CPU开销。

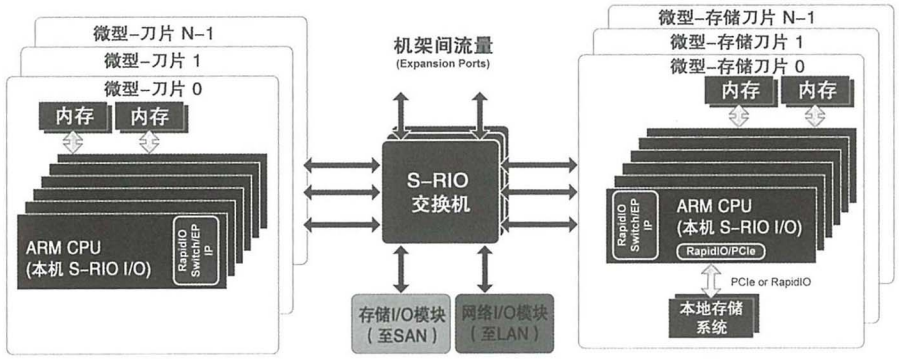
■图1：相对于架顶式系统，基于x86刀片和RapidIO协议的服务器和存储架构具备更低的成本和更高的可靠性。



持卓越的QoS、容错性能和可靠性的同时，可扩展至大量节点和队列。基于RapidIO的架构通过卓越的流量控制和数据包格式，使处理器性能与I/O相匹配，从而能确定性地、准时交付数据，并能高效率运用交换矩阵。总之，RapidIO具有出色的功能，为新一代低延迟且占用空间小的服务器和存储网络提供了卓越的可扩展解决方案。

总结

基于RapidIO的设备可用来运行数据中心中各种不同的应用，例如对延迟敏感的大数据分析、网络搜索、数据分析、多方在线游戏、分布式高速缓存等应用。基于延迟最低的RapidIO协议的系统有一种固有优点，即在保



■图2：与刀片式和架顶式系统相比，基于ARM和其他低功率处理器、具备本机RapidIO器件的微型服务器和存储架构提供最低的系统延迟。