有意思的论文FPGA Catapult（P1）

Original 2018-01-20 Accela Zhao Accela推箱子

论文如下：

[Catapult v1: A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services](https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/Catapult_ISCA_2014.pdf)

[Catapult v2: A Cloud-Scale Acceleration Architecture](ftp://ftp.cs.utexas.edu/pub/dburger/papers/MICRO16.pdf)

计算和存储规模不断上升，CPU核数和主频上升达到瓶颈。FPGA、ASIC、RDMA等原本见于HPC（高性能计算，High Performance Computing）加速，泛称作加速器（Accelerator），其功能、技术演进，成本下降，开始大范围被互联网和数据中心采用。FPGA比CPU价格低，能效（Power-efficiency）高，可定制、专用场景计算力强悍；ASIC性能、能耗全面优于FPGA，但开发困难、掩模（Mask）昂贵，不可重编程；GPU则风起于深度学习对计算力的如饥似渴，浮点运算强，大批量计算，可软件直接开发。FPGA、ASIC、GPU各有特性和应用，本文着重FPGA。

Catapult v1/v2来自微软在Bing搜索引擎和Azure SDN中应用FPGA的研究和实践，架构有数次变迁。Bing的页排序（Page Rank）和搜索对低延迟流处理（用户请求流）的大规模计算和降低成本有天然需求。在Bing产线成功的FPGA又被推广到Azure SDN。SDN（Software Defined Network）对硬件加速（Offloading）也有天然需求；例如40Gb/s网络，即5GB/s / CPU核2.4GHz ~= 2，即如果每个Byte用1 CPU cycle处理，如加解密，都需要专用2个核；对服务器这难以接受。而CPU主频和核数提升有发展瓶颈，且这个性能问题又无法通过Scale out解决；这就有了FPGA等硬件加速的需求。

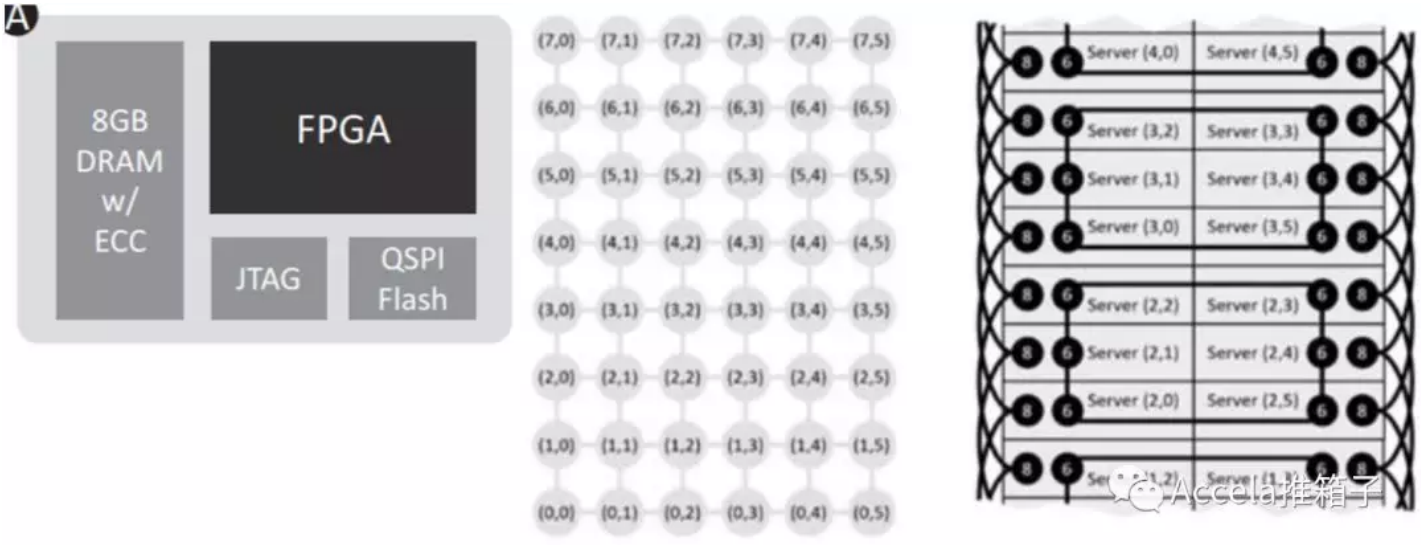
对下面的链接详述了Catapult 项目在微软的发展脉络。

[Programmable chips turning Azure into a supercomputing powerhouse](https://arstechnica.com/information-technology/2016/09/programmable-chips-turning-azure-into-a-supercomputing-powerhouse/)

Catapult v1

Catapult v1论文采用的是机架级（Rack-scale）FPGA架构，每个服务器配备一个高端FPGA卡。与之相对的没有被采用的方案是由专用机架集中提供FPGA卡；这个方案引入异构服务器而不便管理，FPGA专用机架易成为单点故障，网络易出现TCP in-cast问题（多对一数据传输，见DCTCP论文）。

FPGA卡上配有8GB内存，使用ECC保护；一方面是为了提高容错，从而不必在FPGA中设计复杂的带重传的网络协议（重传由应用控制），占用卡上面积。服务器使用SAS连接FPGA，一对四组成网络；FPGA中实现专门的网络协议，SAS连接可达到亚微秒级延迟，单向传输带宽达到10Gb/s。因为使用SAS连接，可接入结点数和连接距离受限，于是成了机架级（Rack-scale）；另一方面，SAS连接也是超级计算机（Supercomputer）的做法。



配备FPGA卡后，服务器TCO增加小于30%，耗电增加小于10%；Bing支持同样吞吐量和延迟只需一半服务器，可以说成本优势显著。但这里可以看到FPGA与GPU的差异：采用FPGA需要专门设计板卡（通常需2年研发才能上线）；虽然FPGA可编程，但多是用Verilog等硬件语言开发，且复用组件（FPGA厂商常常提供IP（Intellectual Property））、开发生态等偏弱；而GPU可立即安装在服务器上，可由软件语言开发，接入的软件生态有大量复用组件和开发人员。

另一特点是Shell Architecture。FPGA卡与周边硬件、网络、PCIe、内存DMA的交互，都需要编写程序支持（类似驱动）；应用程序可由Partial-reconfiguration编写在另一处。这就区分出了通用程序和应用程序，通用程序被称为Shell，可复用。而应用程序被称作Role，例如Bing搜索所需的不同流水线（Pipeline）组件，都可利用FPGA编程能力快速迭代；这是ASIC无法做到的。

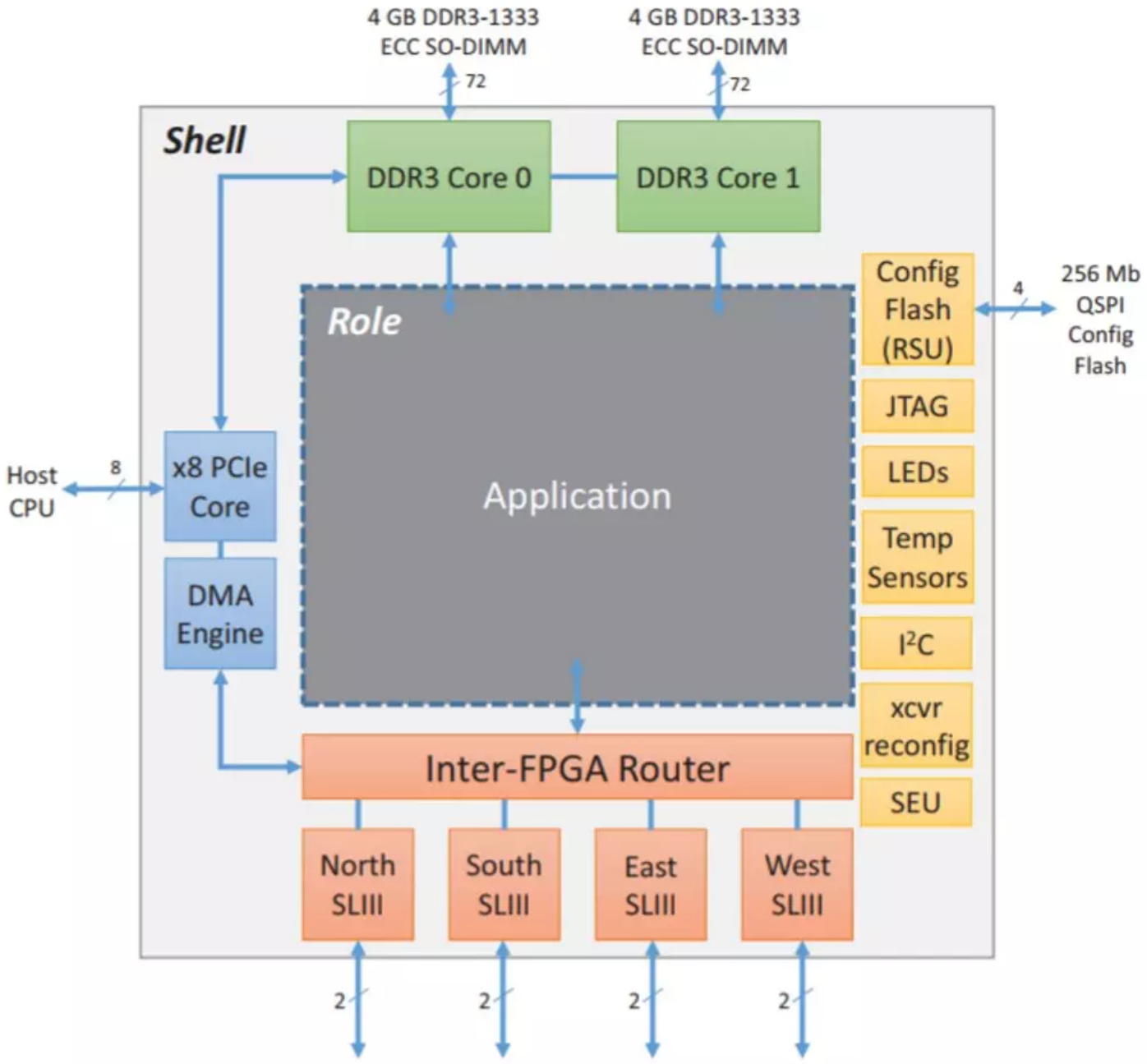


Figure 3: Components of the Shell Architecture.

另一方面，与GPU不同，FPGA中没有核（core、CPU核）的概念（用MemBlaze等自己编个核除外），计算流水线可被转化为硬件逻辑直接铺写在FPGA上；如果有剩余面积（即剩余的逻辑单元），还可以编写更多的流水线，一起异步工作。而GPU仍是基于核的，拥有成千上万的核，同步迭代（Wrap）地批量执行指令，以实现高吞吐量。对FPGA，复杂逻辑可被平铺为硬件流水线，可能在一个时钟周期得出结果，延迟极低；而对GPU，复杂逻辑对应许多指令，需要多个时钟周期执行，延迟难以降低。可以想见，FPGA适用于低延迟的流处理（< 10 us），而GPU则可用于批量计算（> 1 ms）。下面李博杰的知乎回答有更多深刻见解。

[FPGA - 李博杰在知乎的回答]
(https://www.zhihu.com/question/24174597/answer/138717507)

Catapult v2

论文名改成了“Cloud-scale”，或者叫“Datacenter-scale”，这就是Catapult v2与v1的最大区别。在Catapult v1中使用SAS线缆互连FPGA，v2指出这限制了互连的规模（Scale），且在SAS一端的FPGA崩溃时，可能卡死或崩溃另一端结点；可以看到v1为其实现了特殊的软件隔离机制。Catapult v2使用以太网（Ethernet）互连FPGA，取代了SAS；连接规模从v1的6×8结点直连网络（但部署了1.6K+服务器），发展到支持25K服务器互连（需要多层交换机）。以太网延迟比SAS略高，在10us左右，而SAS可达到亚微秒级；但FPGA终于实现了数据中心级互连，即Datacenter-scale”、“Cloud-scale”（下一个大概可以叫“Web-scale”什么的）。

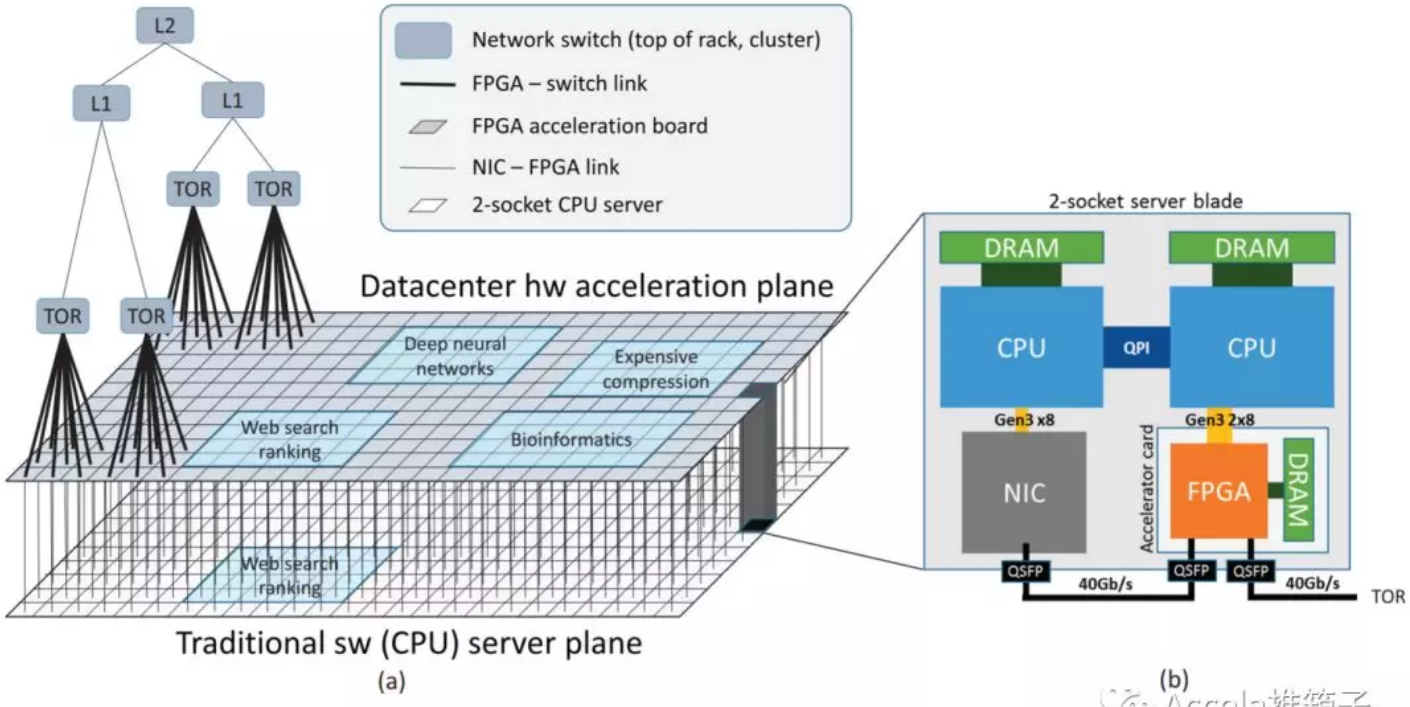


Fig. 1. (a) Decoupled Programmable Hardware Plane, (b) Server + FPGA schematic.

另一大特色是，网卡（NIC）先与FPGA连接，然后FPGA连入以太网；换句话说，网卡不直接连入网络，而是让FPGA来“bump-in-the-wire”地插入到网卡与外部网络之间。这种架构可以说非常创新。因为通常的设计中，即使FPGA与网卡并列，也是由网卡接入网络，FPGA在旁辅助，如KV-Direct论文；它们通常被归为Smart NIC。而Catapult v2的设计中，可以看到其对网络加解密、SDN等的看重；大量功能可以低延迟地完成，不需网卡转接、不走PCIe增加延迟，不需CPU参与。

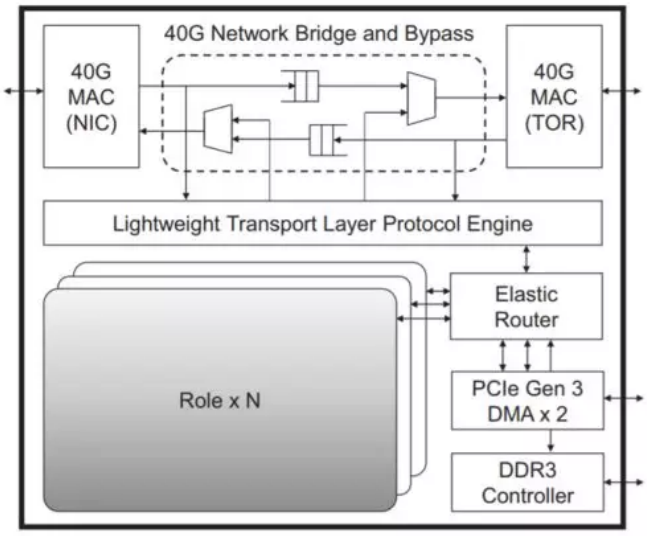


Fig. 4. The Shell Architecture in a Single FPGA.

	ALMs	MHz
Role	55340 (32%)	175
40G MAC/PHY (TOR)	9785 (6%)	313
40G MAC/PHY (NIC)	13122 (8%)	313
Network Bridge / Bypass	4685 (3%)	313
DDR3 Memory Controller	13225 (8%)	200
Elastic Router	3449 (2%)	156
LTL Protocol Engine	11839 (7%)	156
LTL Packet Switch	4815 (3%)	-
PCIe DMA Engine	6817 (4%)	250
Other	8273 (5%)	-
Total Area Used	**131350 (76%)**	-
Total Area Available	172600	-

Fig. 5. Area and frequency breakdown of [illegible] with remote acceleration support.

FPGA的卡上程序同样沿袭了Shell Architecture。可以看到通用程序如网络协议、内存驱动、PCIe DMA驱动等卡上通用程序。网络协议实现了传输层，LTL - Lightweight Transport Layer，能够进行流控制、重传、包排序等；而Catapult v1的网络协议则本着尽量简单以降低延迟，没有重传，不保存源数据包，数据纠错通过连接层ECC、CRC、以及应用重传来做。应用被称作Role，可以有多个Role共享一块FPGA。ER – Elastic Router的功能像虚拟交换机，通过端口（Port）管理同FPGA或不同FPGA的Role之间、或与主机之间的通信；数据传输可以是PCIe DMA、卡上 DRAM、LTL等。

说起卡上DRAM，ClickNP论文中提出使用管道（Channel）的方式，让FPGA与主机（Host）通过PCIe DMA直接通信更加高效，而不需要数据到卡上DRAM绕一圈。在李博杰（ClickNP论文第一作者）的知乎回答中也有详述。

在Catapult v2中，应用不仅可以使用本地的FPGA，还可以使用远端的FPGA。FPGA上的Role不仅可以与同FPGA的Role通信，还可以与远端的FPGA通信。由以太网互连，FPGA互相通信构成资源池，一个应用可以利用全数据中心资源为其加速。另一方面，FPGA连接在网卡之外，FPGA对FPGA的通信，只需经过以太网而不需走PCIe；可以说FPGA离FPGA比里同主机CPU更近。相比快速发展的网络带宽而言，PCIe的带宽偏低；大量使用的PCIe v2约有5GB/s ~ 8GB/s带宽（见Wikipedia：PCI Express）。

PCI Express link performance[29][32]

PCI Express version	Introduced	Line code	Transfer rate[i]	Throughput[i]				
				×1	×2	×4	×8	×16
1.0	2003	8b/10b	2.5 GT/s	250 MB/s	0.50 GB/s	1.0 GB/s	2.0 GB/s	4.0 GB/s
2.0	2007	8b/10b	5.0 GT/s	500 MB/s	1.0 GB/s	2.0 GB/s	4.0 GB/s	8.0 GB/s
3.0	2010	128b/130b	8.0 GT/s	984.6 MB/s	1.97 GB/s	3.94 GB/s	7.88 GB/s	15.8 GB/s
4.0	2017	128b/130b	16.0 GT/s	1969 MB/s	3.94 GB/s	7.88 GB/s	15.75 GB/s	31.5 GB/s
5.0[30][31]	*expected in Q2 2019*[33]	128b/130b	32.0 GT/s[ii]	3938 MB/s	7.88 GB/s	15.75 GB/s	31.51 GB/s	63.0 GB/s

FPGA如何与主机相连，插在主机的何处，有许多不同方案；论文中有详细对比，包括CPU/内存整合，加速器（Accelerator）互连规模，加速器类型（GPU、ASIC、FPGA）几个方面。大体而言，FPGA可以加入CPU中，通过QPI总线与CPU核高速连接；可以与内存总线相连；可以放在IO设备附近，通过DMA与内存交换数据；也可以放在网卡附近，进行网络加速，如Catapult v1/v2。不同的方案亲和不同密集型的负载，而数据是否需要频繁通过PCIe总线也会影响延迟。

总之，Catapult v1/v2可算作巨大的技术原创突破，FPGA在Bing搜索和Azure SDN上的大规模产线应用很成功；相关媒体报道丰富。FPGA卡和配套服务器的研制通常需要约2年的时间，加上软硬件程序开发的时间，也给对手追赶方案造成时间差。FPGA加速的服务器设计和数据中心设计可逐渐被用于公司内外的更多服务和产品。回想谷歌TPU在云上开放，定制硬件虽然研发成本高昂，但公有云可成为其廉价复用的方式。

FPGA可编程原理

进一步深入论文的设计，需要理解FPGA的硬件构成，尤其是可编程的原理。理解介质的独特，往往是开发出高效分布式软件的基础。

（未完待续……后面将讲解FPGA可编程的原理，云虚拟化方案，多种应用场景等。注：本文为个人观点总结，作者工作于微软）