Forwarding Table Scalability For Cluster Switches

Dong Zhou, Bin Fan, Hyeontaek Lim, David G. Andersen, Michael Kaminsky*, Michael Mitzenmacher†
(CMU, *Intel, †Harvard)

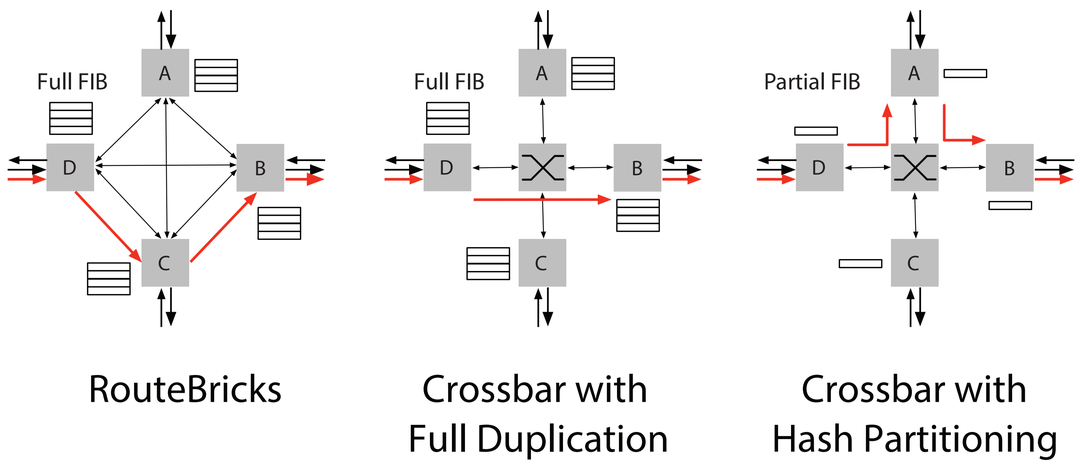
Problem

- **FIB capacity does not scale out with the number of servers (line cards)**
- **Goal: achieve FIB scalability without increasing the amount of internal traffic**

Potential Applications

- **Huge flat-addressed networks**
- **Hardware-based switches?**
- **Flow-oriented applications (SDN? NAT?)**
- **We are looking for more! Ideas?**

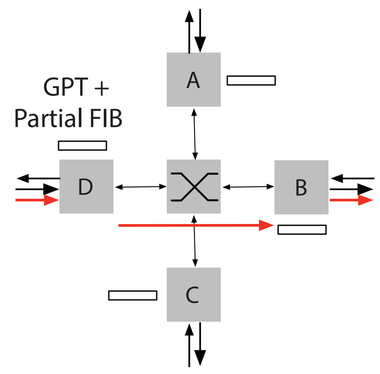
Existing Architectures



RouteBricks | Crossbar with Full Duplication | Crossbar with Hash Partitioning

Solution: XBricks

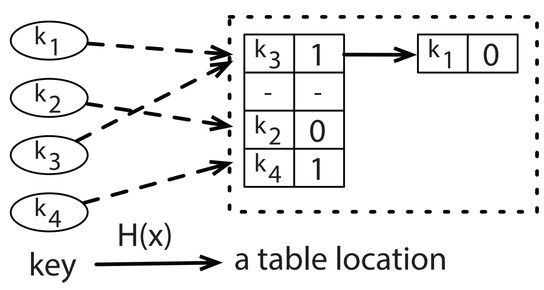
- **Each node is responsible only for FIB entries that have the node itself as egress node — partitioned FIB**
- **Each node uses a global partition table to map all the known addresses to egress nodes — one hop latency**



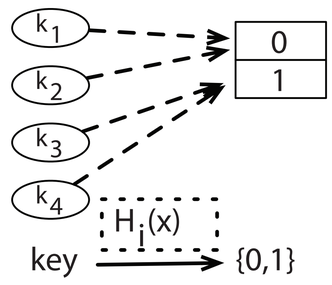
XBricks

Global Partition Table: XSep

- **Two observations**
- **The range of possible values is *very small***
- **Unknown keys can be mapped to *incorrect values* instead of "not found"**
- **Set separation instead of general key-value mapping**
- **2-4 bits per entry for a small number of servers**

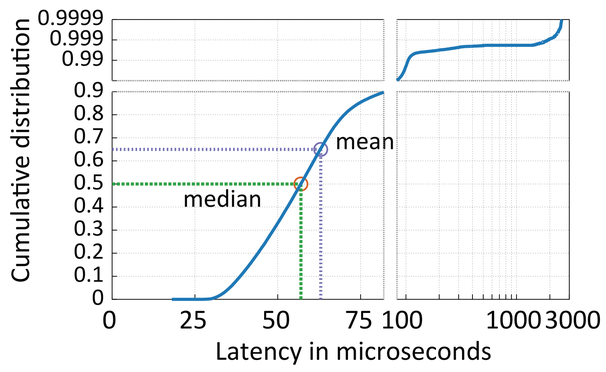


Hash Table

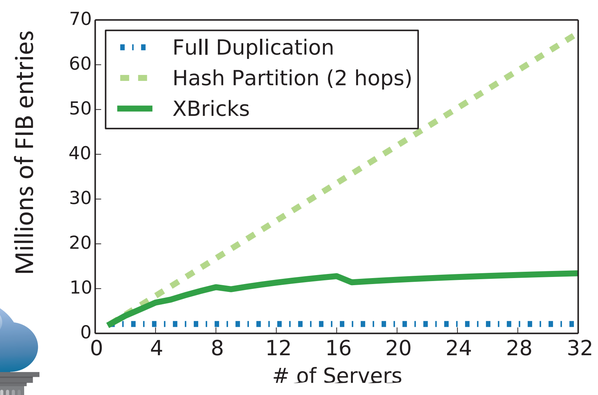


XSep

Evaluation

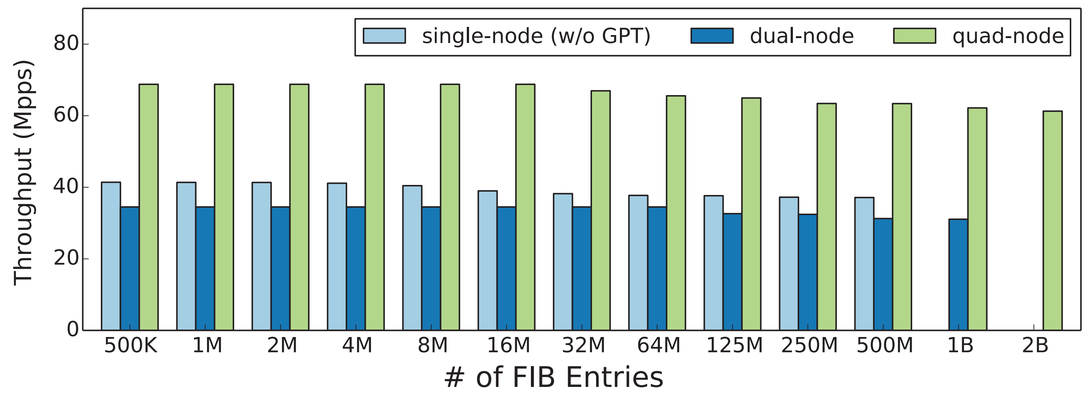


End-to-end latency



Scalability

Throughput of XBricks vs. FIB size



Intel Science & Technology Center for Cloud Computing

Carnegie Mellon University | Georgia Tech | intel | Princeton University | UC Berkeley | University of Washington