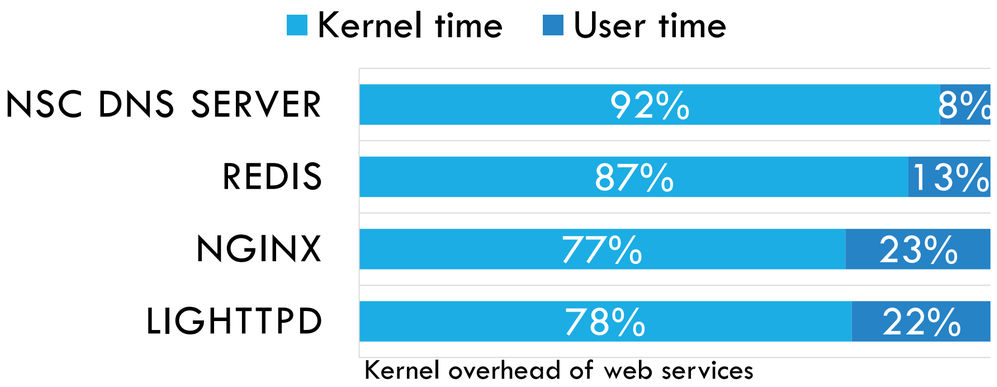
Microsoft Research

IPC-Direct: Fast and Compatible Inter-Process Communication in User Space

SOSP'17 SRC #39

Bojie Li, Tianyi Cui, Zibo Wang, Lintao Zhang

Kernel overhead dominates web service performance

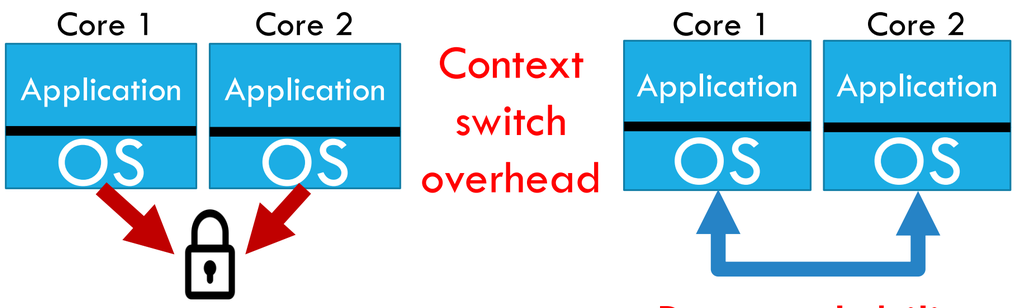


Kernel overhead of web services

Sources of kernel overhead:

- Unnecessary processing in kernel
- Memory copy and context switch
- Distributed coordination under high contention

Bottleneck of kernel-based IPC



Lock contention

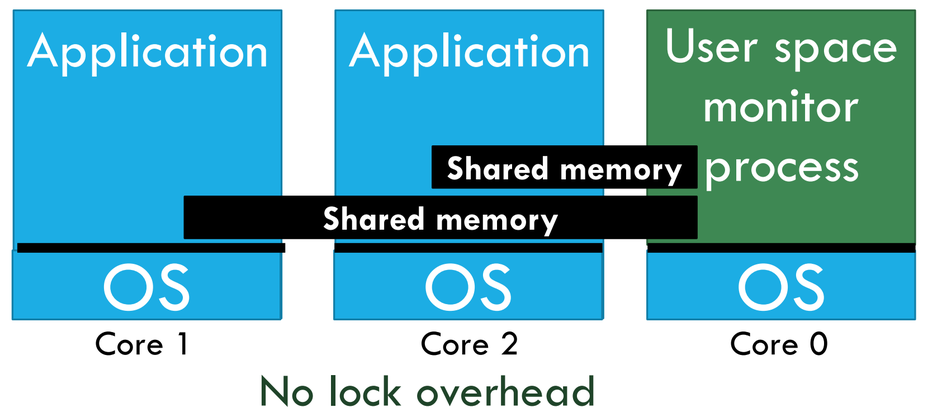
Traditional Inter-process communication with lock

Poor scalability

Multikernel Inter-process communication with message passing

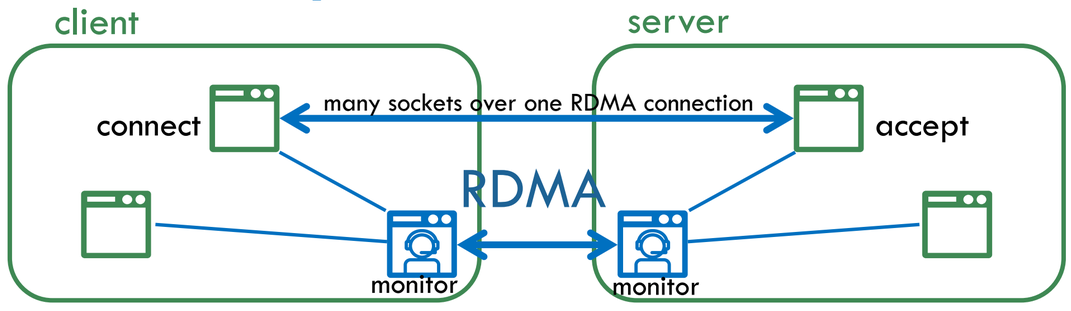
Our design: use dedicated coordination core to process IPC in user space

No context switch, better locality



No lock overhead

Scale to multiple servers with RDMA

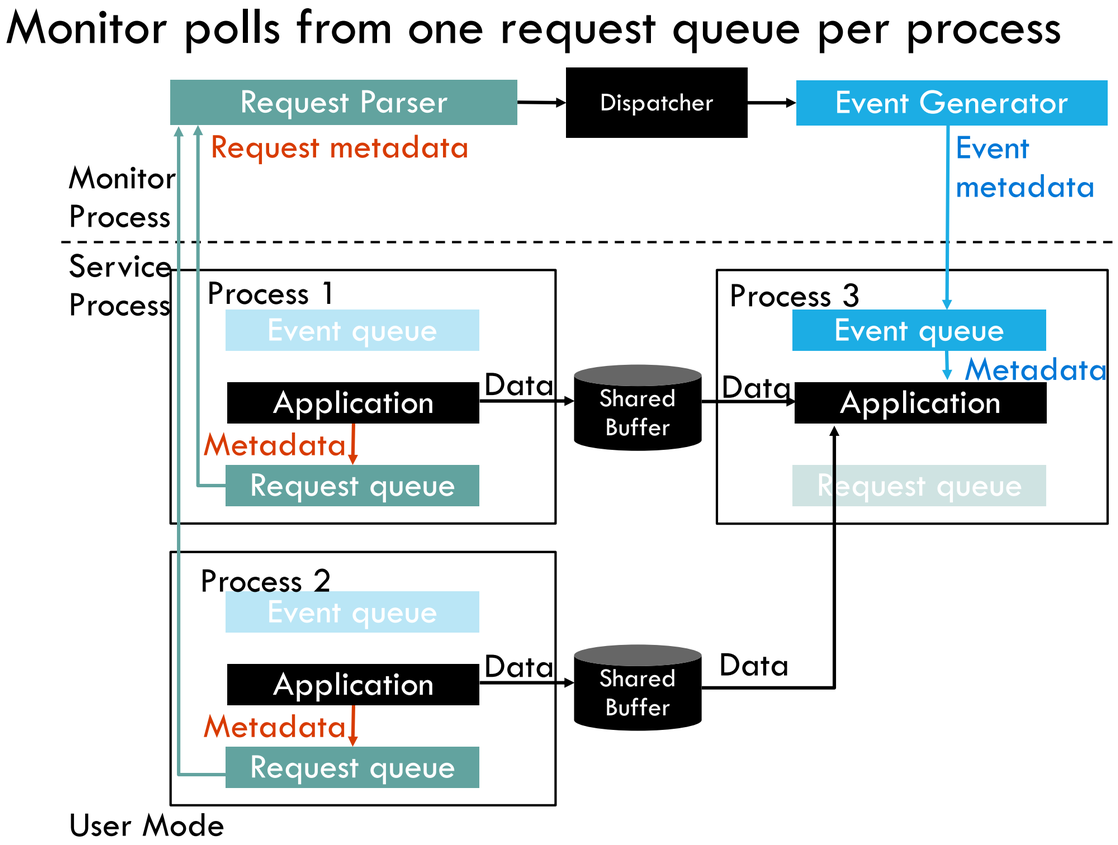


IPC-Direct advantages

- High throughput
- Low latency
- Compatible with POSIX API (using LD_PRELOAD)
- Preserve process isolation
- No new hardware or kernel modification

Two implementations of request queues

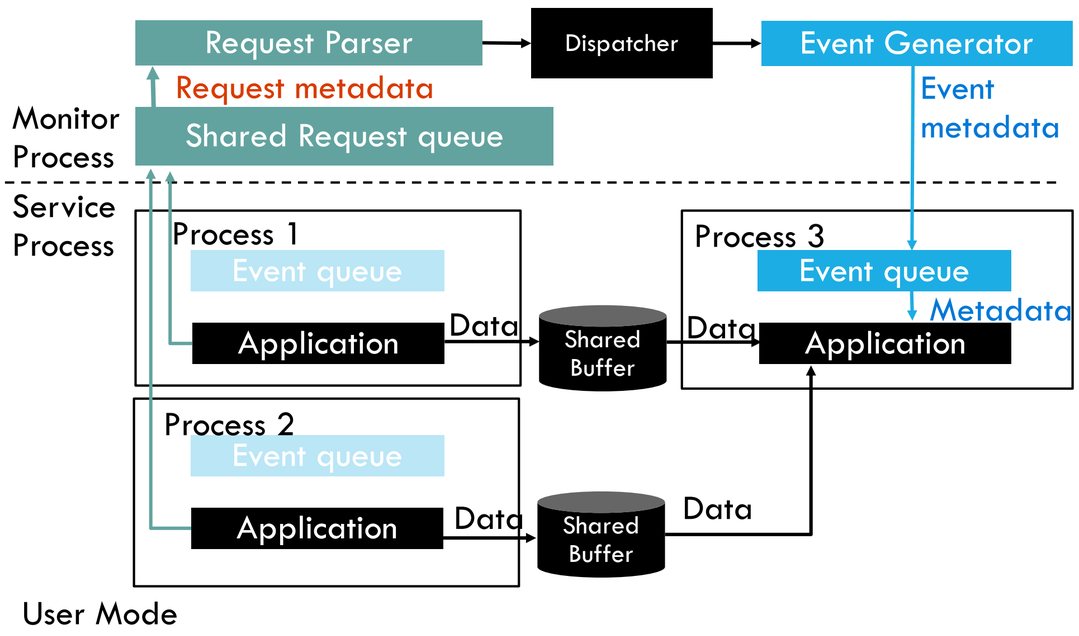
Monitor polls from one request queue per process



Pros: No client contention

Cons: Waste polling when most processes are idle

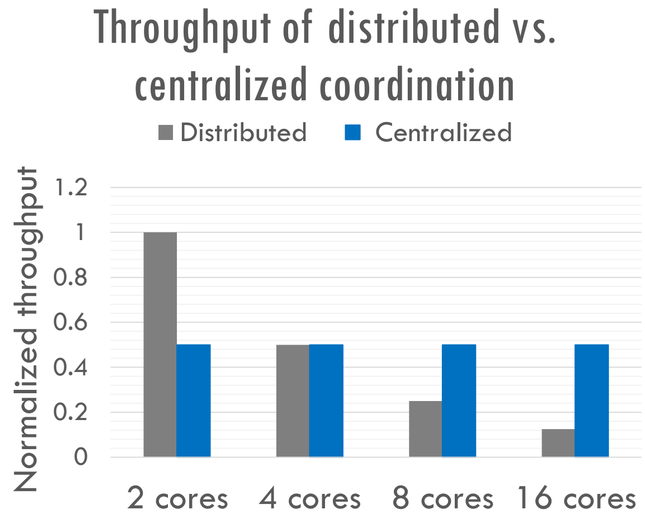
Applications send to a shared request queue

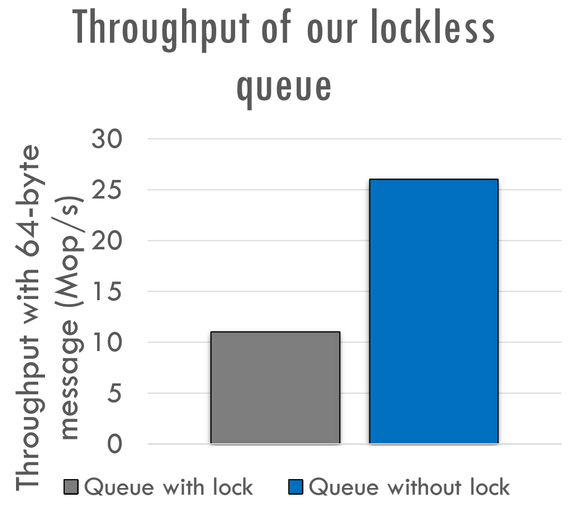


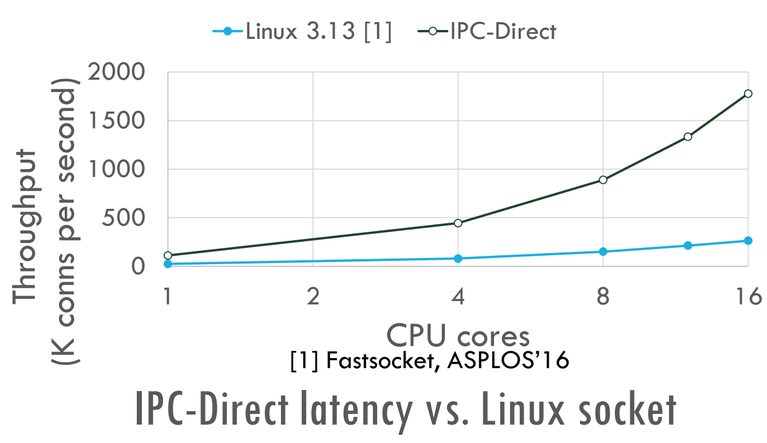
Pros: Monitor receives requests more efficiently

Cons: Client contention in atomic operation

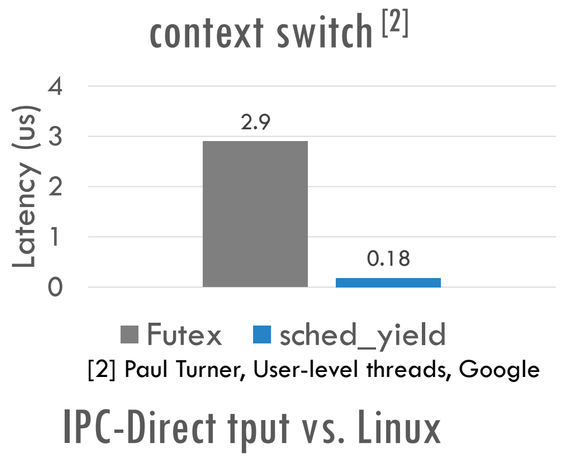
Microbenchmarks



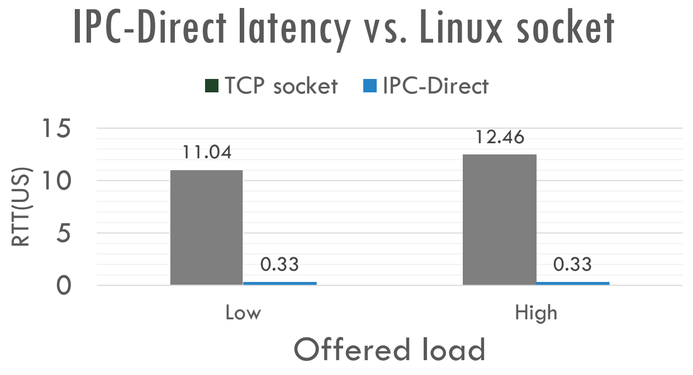


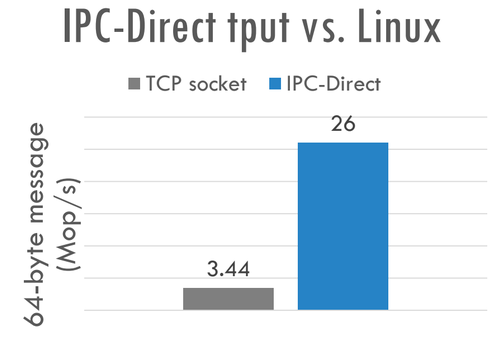


[1] Fastsocket, ASPLOS'16



[2] Paul Turner, User-level threads, Google





Discussion: Use SmartNIC as coordinator

Better throughput, Worse latency