Database Performance at Scale Masterclass

Database Internals

Pavel Emelyanov & Botond Dénes

SCYLLA

SCYLLA

TURSO

Hardware (and kernel) matters

- CPU
- Memory
- Disk I/O
- Networking

CPU

Share nothing

- Cross-cores locking costs
  - Cache lines
  - CPU ticks
- Amdahl’s law

Processes/Threads → Fibers

- Context switch time matters
- Enforced preemption switch also matters
- Locking synchronization is a must

Quirks

- Execution stages
- Branch (mis)prediction

Memory

Allocation

- Fragmentation
- Slab-like allocation
- Log-structured allocation

Cache control

- Let kernel or application do it?
- Caching can exist at different levels
  - Raw IO blocks
  - Decoded objects

Disk I/O

Types of I/O

- Buffered reads/writes
- Memory mapped IO
- Direct IO
- Asynchronous direct IO
    - IOUring

FS vs Disk

- Filesystem adds a level of manageability
- Extra IO operations
  - Blocks allocation
  - Inodes metadata
  - Journal

How moderns SSDs work

- Random reads – YES
- Random writes – NO
  - Blocks vs Pages
  - Internal parallelism
  - FTL and background operations
- Sustained vs Burst throughput
- Mixed workload handling

Networking

Tribute to ANK

- Linux kernel TCP/IP had been kicking arses since day 0
- Coupled with epoll/AIO
- Can be tuned for both
  - RPC messages ping pong
  - Data streaming
- System calls switches still matters

DPDK

- Removes extra switches
- Works better in poll mode

IRQ binding

- NIC IRQ processing times
- NIC Soft-IRQ processing times
- Binding IRQs (and Soft-IRQs) to specific cores

ScyllaDB Internals

Predictably Low Latencies

CPU

- Thread-per core architecture
- User-space scheduler

Memory

- Row-cache
- Page-cache
- Full control over allocations

Disk I/O

- Direct I/O
- User-space scheduler