The Anatomy of Networking in High-Frequency Trading

PJ Waskiewicz

Oct. 27, 2022

NetDev 0x16, Lisbon, Portugal

Goals

- What is High-Frequency Trading (HFT)?
- “Traditional” networking vs. HFT
- Jitter the Killer
- HFT and HPC
- Where to go from here?

What is HFT?

- Secretive!
- Algorithmic-based trading, typically with very high volumes
- Trading strategies rooted in Quantitative Research
- Strategies executed via software or hardware
- Massive amounts of data to analyze
- Predictable latency is paramount!

What is HFT?

- Exchanges have individual protocols, nuances and quirks
- Exchanges eventually run on standard Ethernet
  - Typically 10GbE
- Quantitative research requires CPU horsepower
- HPC environments / Grid networks are different
- Predictable latency is STILL paramount!

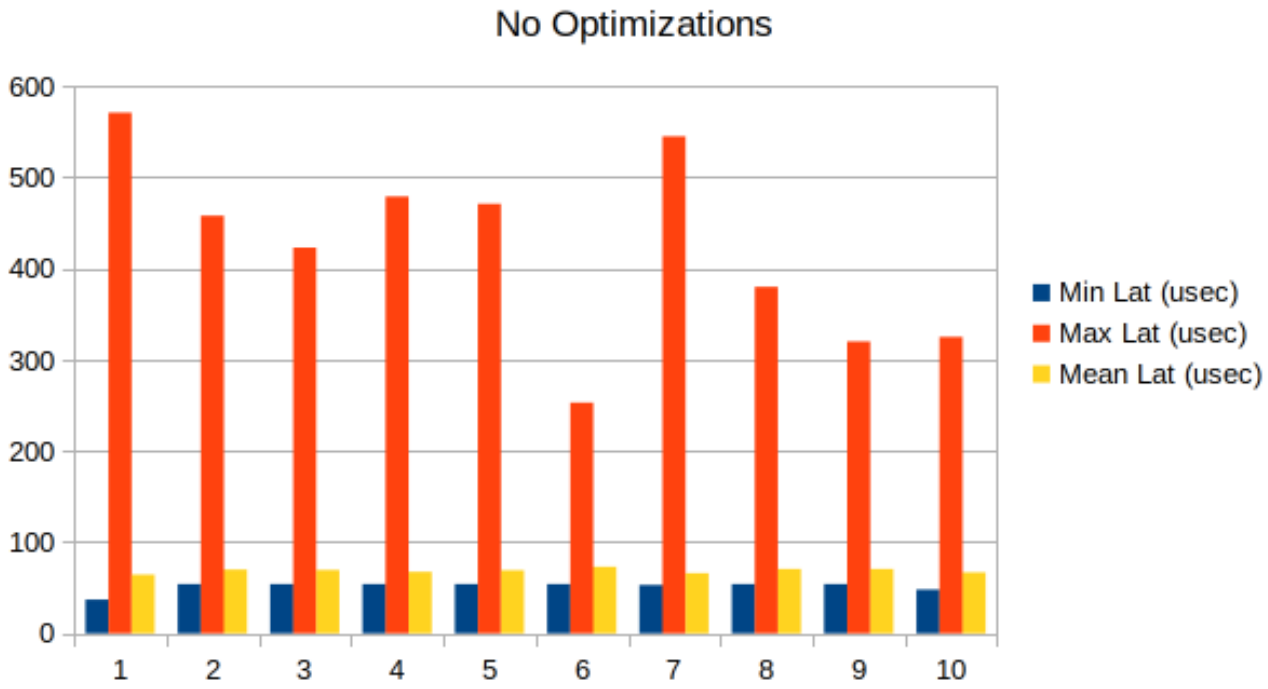
Traditional Networking vs. Latency

- Kernel stack latency performs poorly without help
- Pinning workloads
- NUMA locality
- Interrupt affinity
- CPU isolation
- Benchmarks become very synthetic

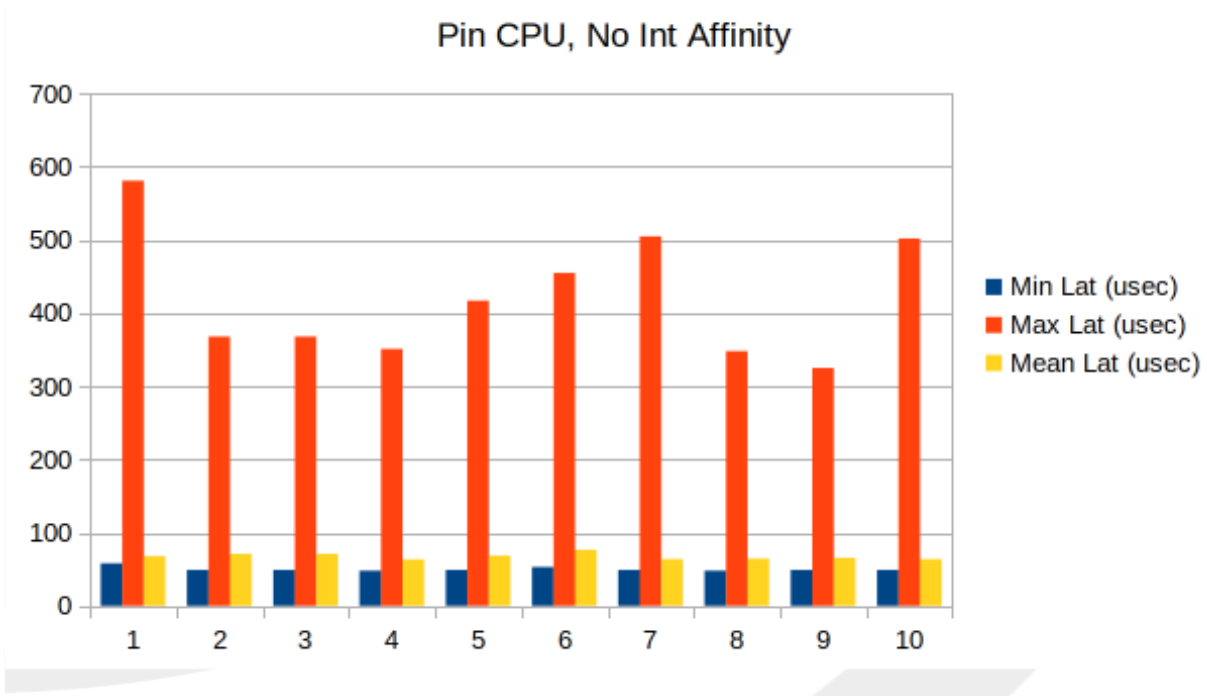
Benchmarks

- Simple as possible setup: netperf with TCP_RR
- Measure ping-pong E2E latency
- Start standard, extend to max custom tuning
- Configuration
  - SUT: Intel® Xeon® ES-2640 (Sandy Bridge), Broadcom bnx2x 10GbE adapter, stock Fedora 36
  - Peer: Intel® Xeon® Platinum 8180 (Skylake), Intel 82599ES 10GbE adapter, Gentoo Linux with 5.15.72 kernel
  - Switch: Mellanox SN2100, Direct Attach Twinax 10GbE passive cables

Benchmarks

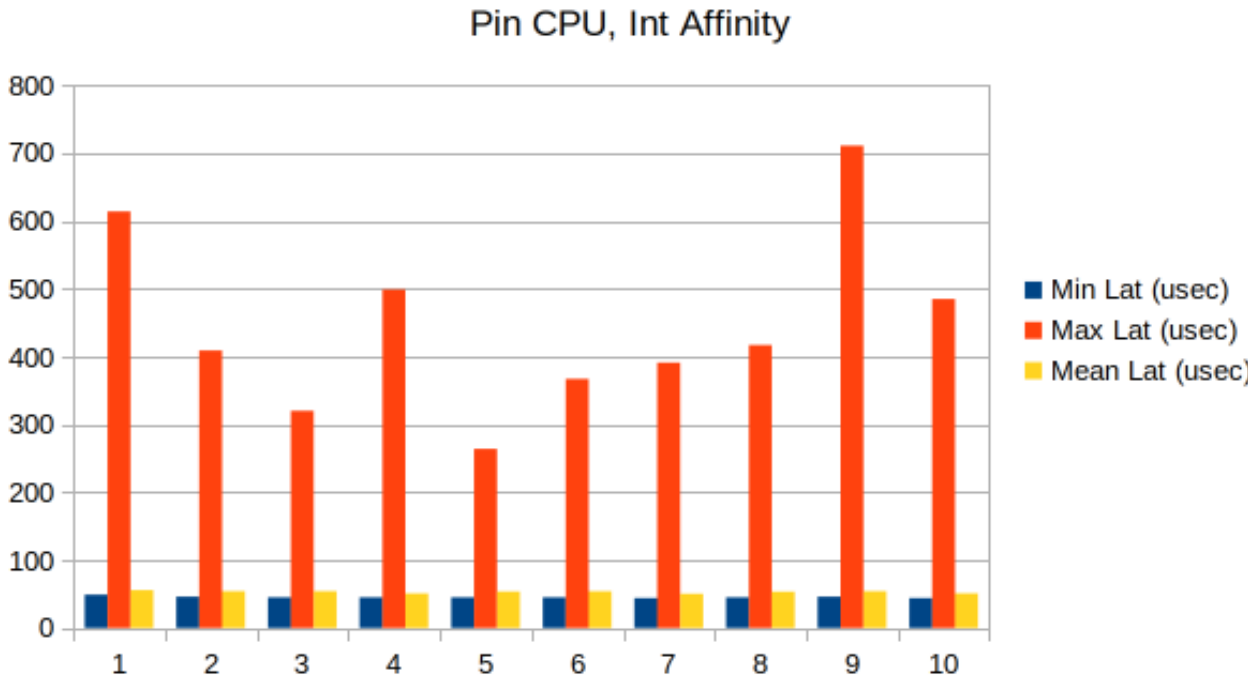


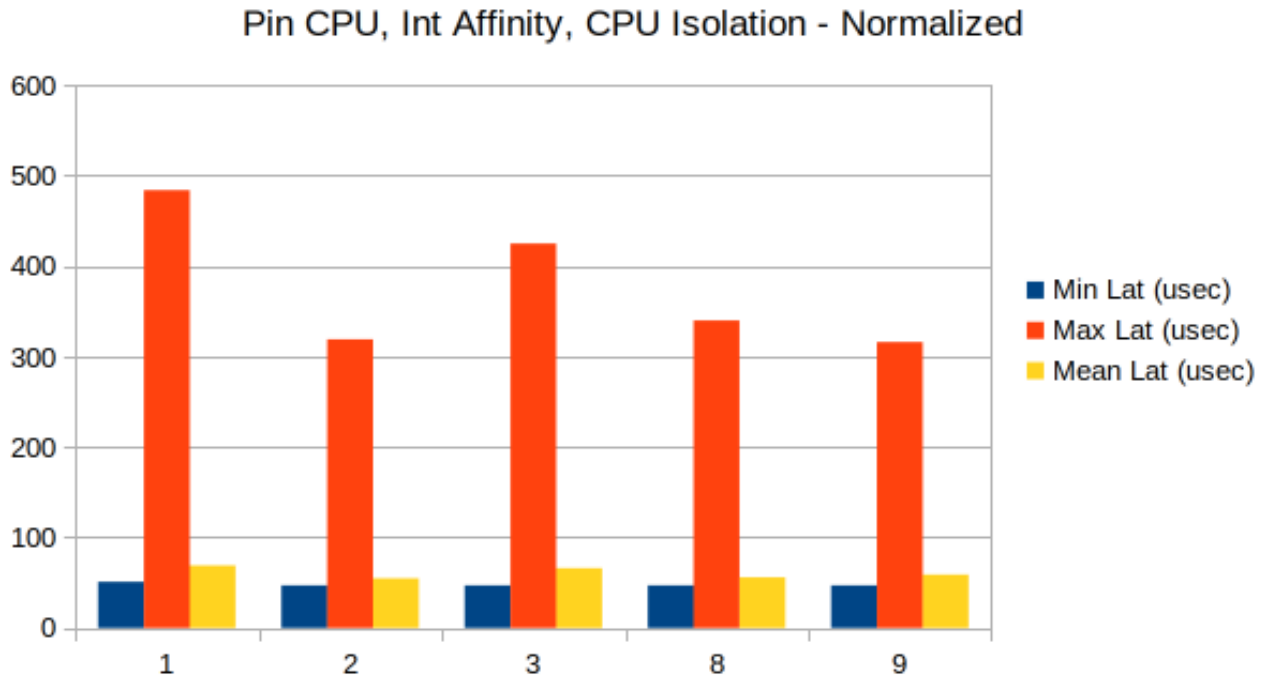
Avg Min Lat: 51.6 usec
Avg Mean Lat: 68.7 usec



Avg Min Lat: 50.1 usec
Avg Mean Lat: 67.6 usec

Benchmarks



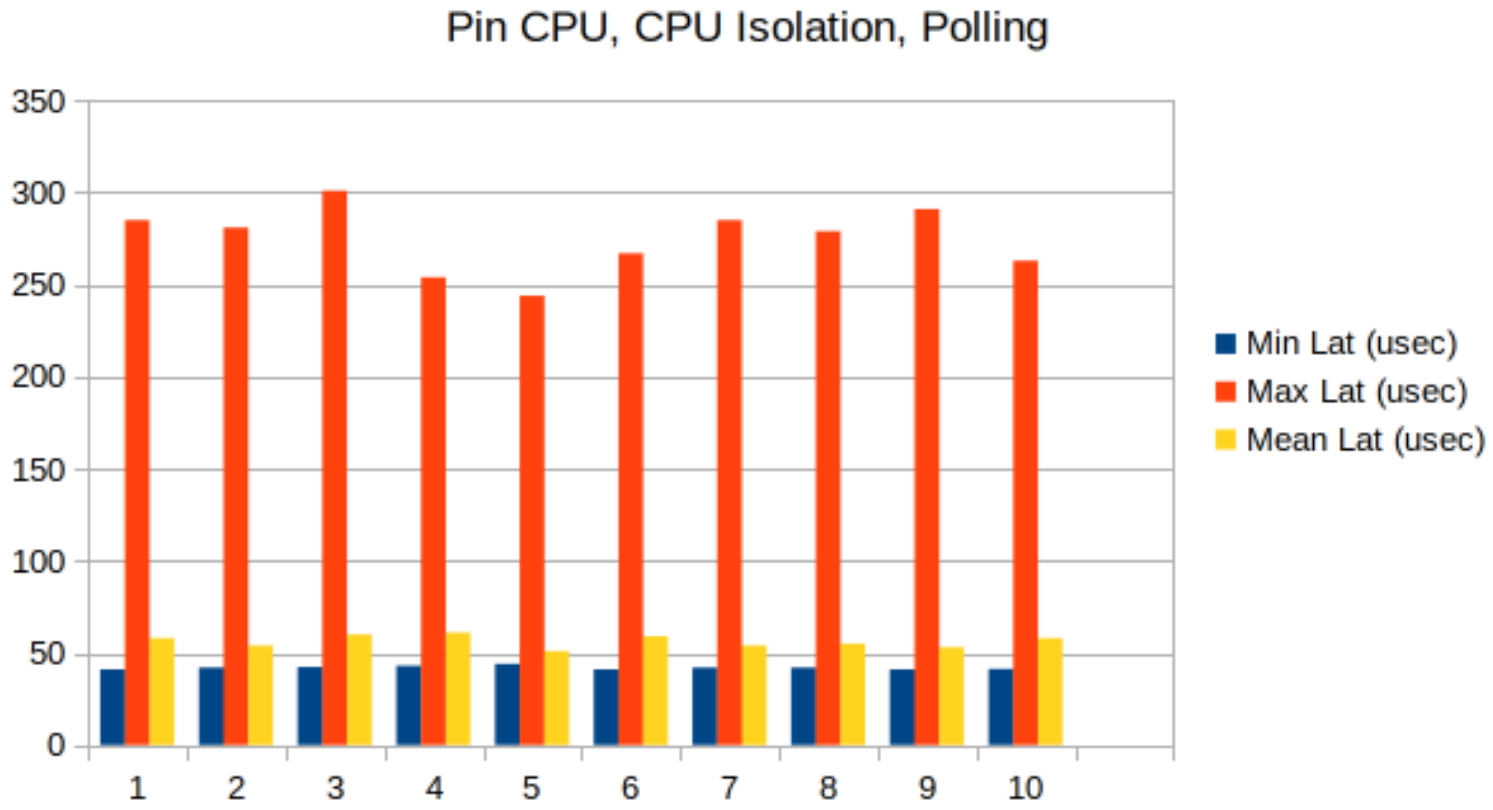


Avg Min Lat: 45.4 usec
Avg Mean Lat: 53.1 usec

Uh-oh...

Avg Min Lat: 47.8 usec
Avg Mean Lat: 61.1 usec

Benchmarks



Avg Min Lat: 41.9 usec
Avg Mean Lat: 56.3 usec

Benchmarks vs. HFT

- Synthetic benchmarks don’t match real-world
- HFT environment allows synthetic benchmark become reality
  - CPU isolation
  - CPU affinity (pin to specific cores)
  - Polling (no interrupts)
  - CPU power states disabled
  - CPU mitigations disabled

AF_XDP to the rescue?

- Throw eBPF at the problem!

AF_XDP to the rescue?

- Kernel bypass without the bypass…
- Hotpath passthru, control traffic to the kernel

- Rx cannot be in `NAPI` / `SOFTIRQ` mode
  - Must poll on Rx, ideally from userspace context
- Tx still requires `sendmsg()` to initiate transmit
  - System call context switches introduce jitter

CPU Isolation Challenges

- `isolcpus` boot parameters work great…
- ...until they don’t
- Sources of “random” IPI’s still exist
- LPC 2022 CPU isolation MC
- System-wide TLB shootdown patches in progress

HPC in HFT

- Exchange interaction requires standard Ethernet at edge
- HPC / Grid environments more purpose-built
- RDMA a much more popular candidate for Grid networks
- Quantitative research on massive datasets
  - Massively-parallel CPU’s can’t wait on one another...
- Predictable latency is still paramount!

io_uring for HPC?

- `io_uring` showing great potential
  - Originally meant to replace `libaio`
- Recent advances showcasing how versatile it is:
  - https://kernel-recipes.org/en/2022/whats-new-with-io_uring/, Jens Axboe
  - https://lpc.events/event/16/contributions/1213/, Josh Triplett
- Submitting data to hardware-mapped rings to io_uring structs

RoCE for HPC?

- RDMA is still king in HPC networks
- Infiniband is niche
  - Expensive
  - Management / administration skillsets differ from Ethernet
- RoCE or iWARP have own challenges
  - Converged networks still have jitter
  - Often needs other tech like DCB

Homa for HPC?

- New RPC-based approach similar to RDMA Verbs
- Having both kernel and userspace support interesting
- Could allow standardization of network switch hardware
- Could make application development easier

Are there others?

- Possible RDMA replacements for HFT?
- Must eliminate jitter
- Must be lowest-latency possible
- Predictable latency!!

Recap

- HFT has two distinct problem spaces:
  - Exchange / trading environments
  - HPC environments
- **Predictable latency is paramount**
  - Trading strategies assume comm latency to be predictable
  - Kernel CPU isolation still needs work
- RDMA may be challenged by emerging technologies
  - AF_XDP
  - `io_uring` showing great promise
  - Homa and other emerging protocols attacking latency

Questions?