Password Cracking HPC

Jeremi M Gosney
Founder & CEO, Stricture Consulting Group

Passwords^12 Security Conference
December 3, 2012

The Problem

Password crackers need more power!

- Yes, really.

A GPU is great!

More GPUs are better.

How many GPUs are enough?

Potential Solutions

Build up

- $$
- Motherboard, BIOS, Driver limitations

Build out

- Distributing load can be tricky

Enter Virtual OpenCL (VCL)

Makes remote GPUs appear as if they were local

Implements entire OpenCL 1.1 Standard

Created by Amnon Barak and Amnon Shiloh, Hebrew University

Distributed by MOSIX (www.mosix.org)

VCL – Pros

Free (gratis)

Supports any and all OpenCL devices

Works with any unmodified* OpenCL app

Eliminates the complexity of distributing load

Makes clustering ridiculously easy

VCL – Cons

Closed source

Closed license

64-bit Linux only

- (Not sure this is really a con)

Need highspeed LAN – No Internet clustering

How VCL Works

Provides a pool of shared devices

Completely transparent to app & user

Two-Layer model

- Broker node
  - Executes kernels on compute nodes.
- Compute nodes
  - Compute.

The Broker Node

This is the front-end

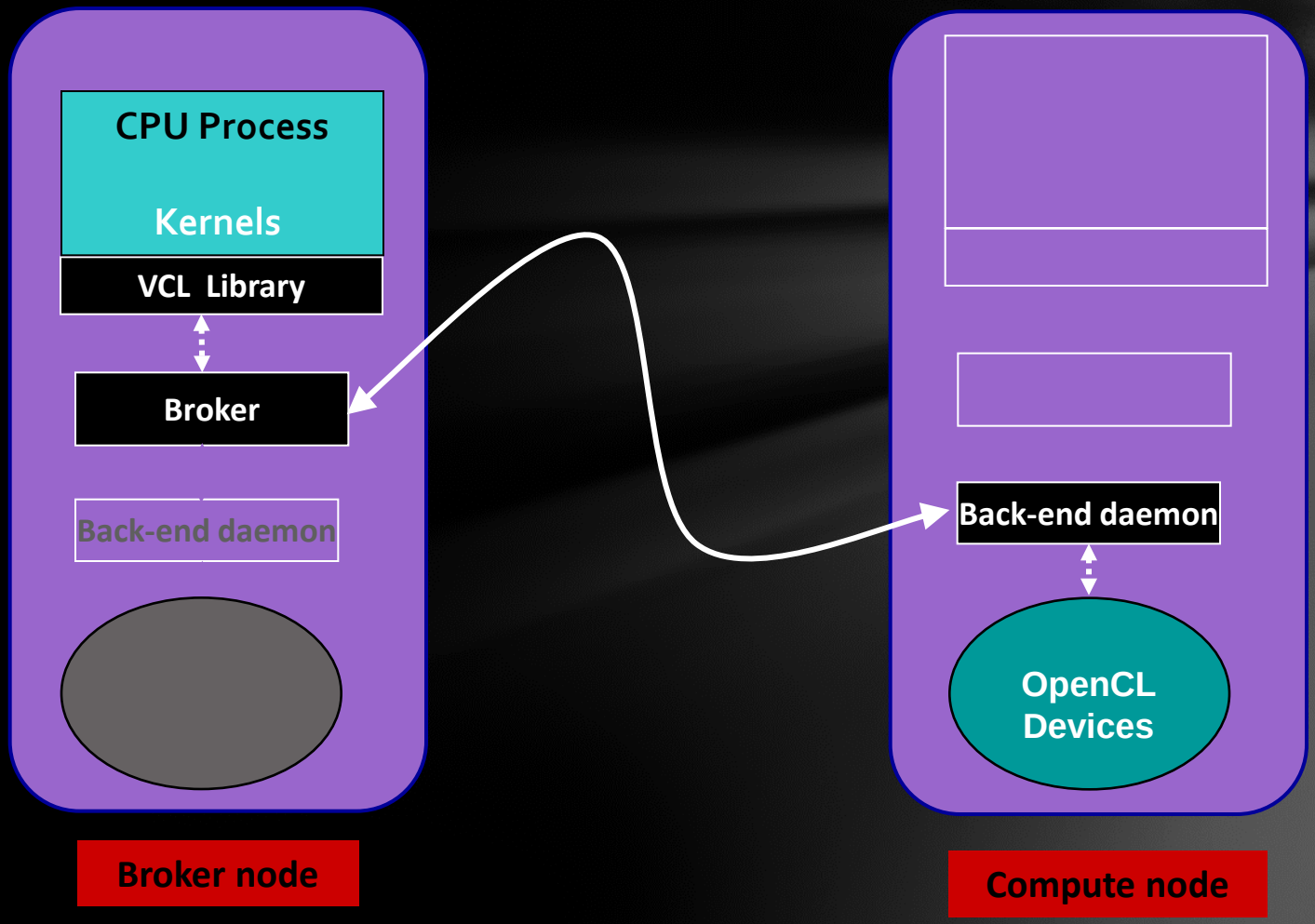
- Has your software stack installed
- Only needs the VCL library and broker daemon
- Does not need any OpenCL devices
- Does not need any drivers

Compute Nodes

This is the back-end

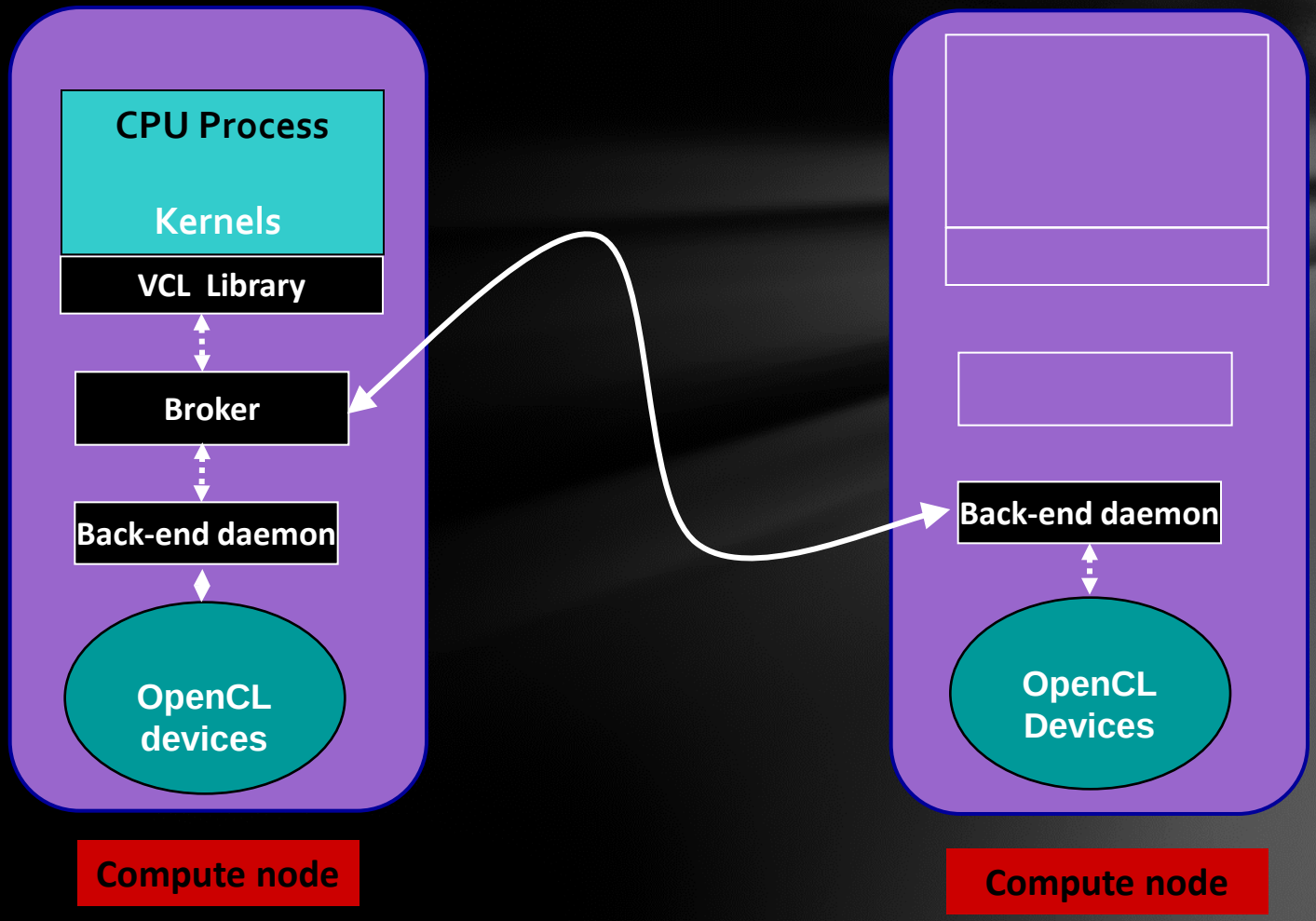
- No software to install outside of VCL back-end daemon
- Needs OpenCL devices
- Needs proprietary drivers & OpenCL runtime

VCL Architecture



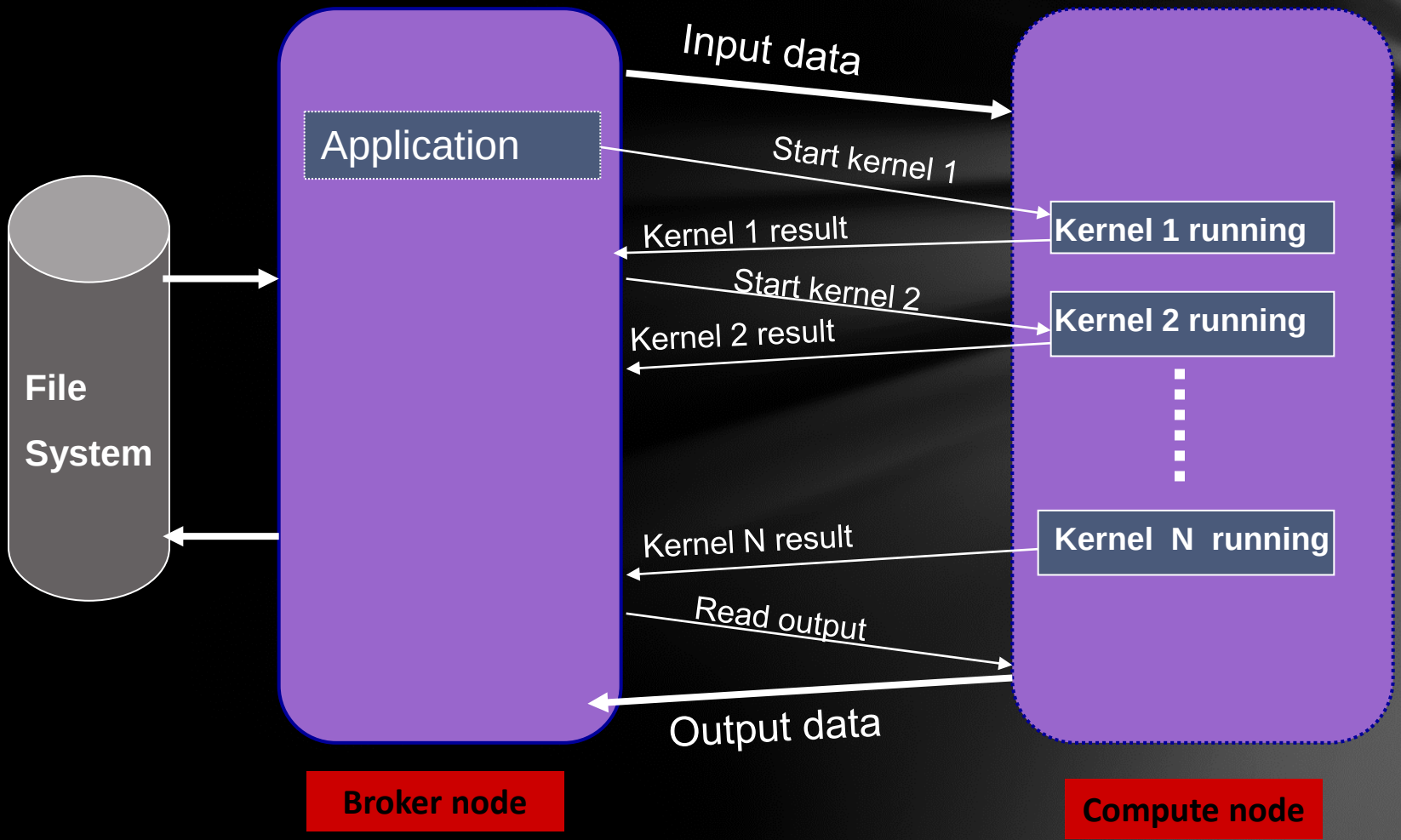
Diagrams Courtesy of Amnon Barak

Alternate Architecture



Diagrams Courtesy of Amnon Barak

VCL Workflow



Diagrams Courtesy of Amnon Barak

SuperCL

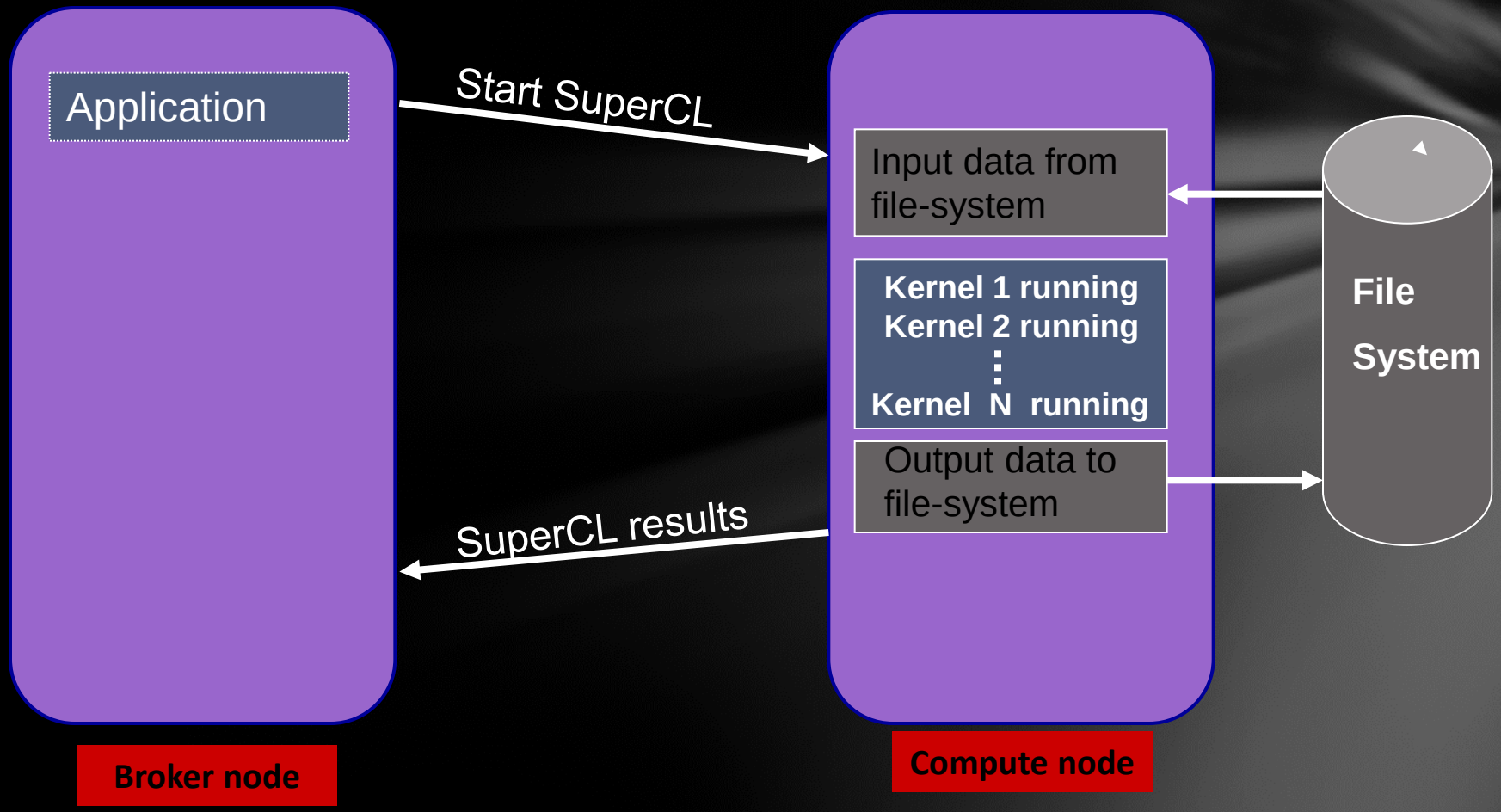
Network optimization library for VCL

Single call for multiple executions

Direct file I/O to/from OpenCL memory object

Async data transfer with broker

SuperCL Workflow



Diagrams Courtesy of Amnon Barak

VCL + Hashcat

We can use VCL for password cracking!

- You knew this part was coming.

Jens Steube added VCL support for up to 128 AMD GPUs in oclHashcat-plus v0.09

MOSIX were more than happy to help debug and resolve issues

VCL + Hashcat – Considerations

Bandwidth

- Brute force / mask attacks need very little bandwidth
- Wordlist attacks need a lot

Latency

- Slow hashes can tolerate some latency
- Fast hashes cannot
- No SuperCL support in Hashcat

Memory

- Broker node needs a lot of it
- Higher the –n value, the more you need

Our Cluster

Five 4U servers

25 AMD Radeon GPUs

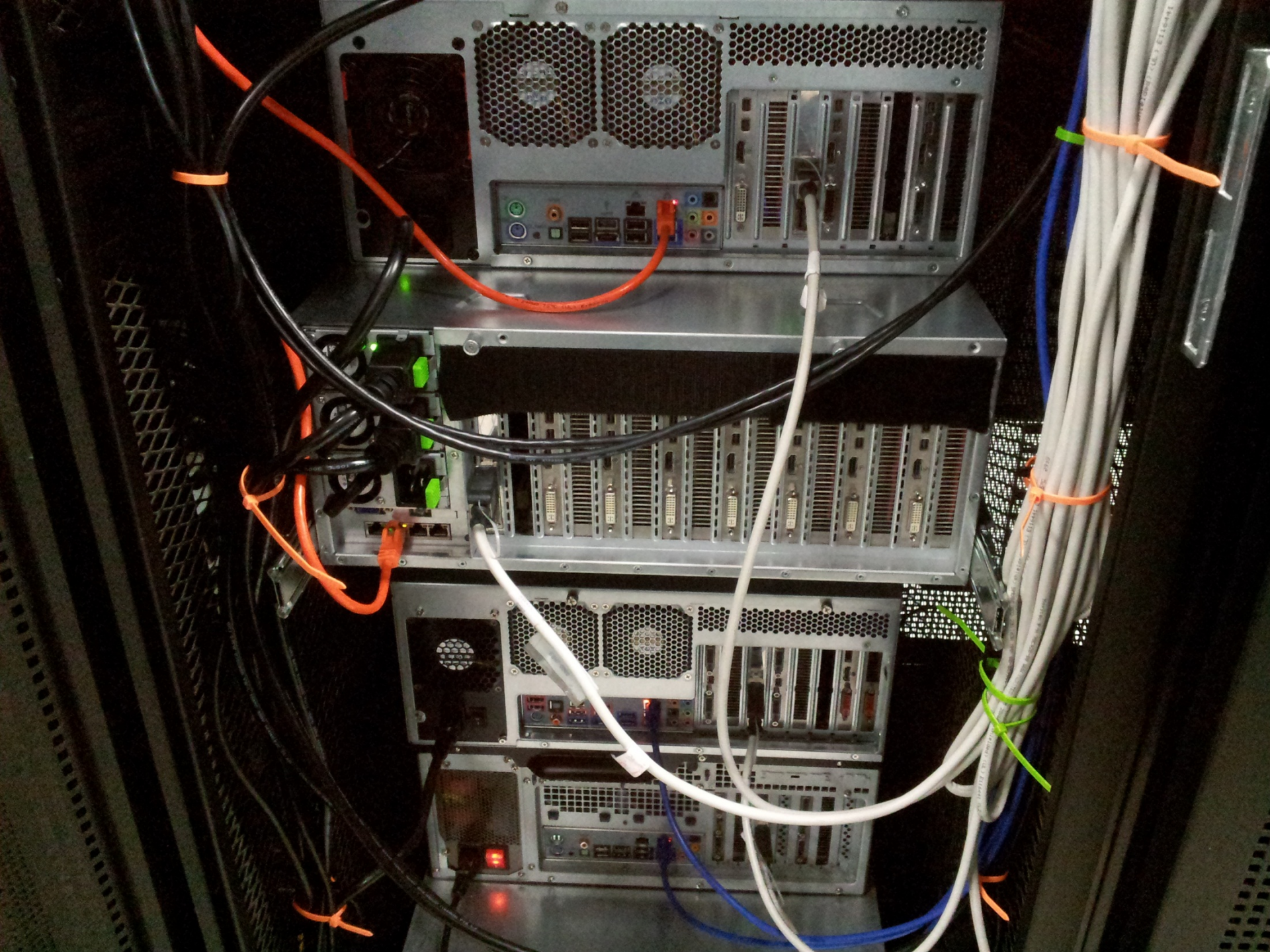
- 10x HD 7970
- 4x HD 5970 (dual GPU)
- 3x HD 6990 (dual GPU)
- 1x HD 5870

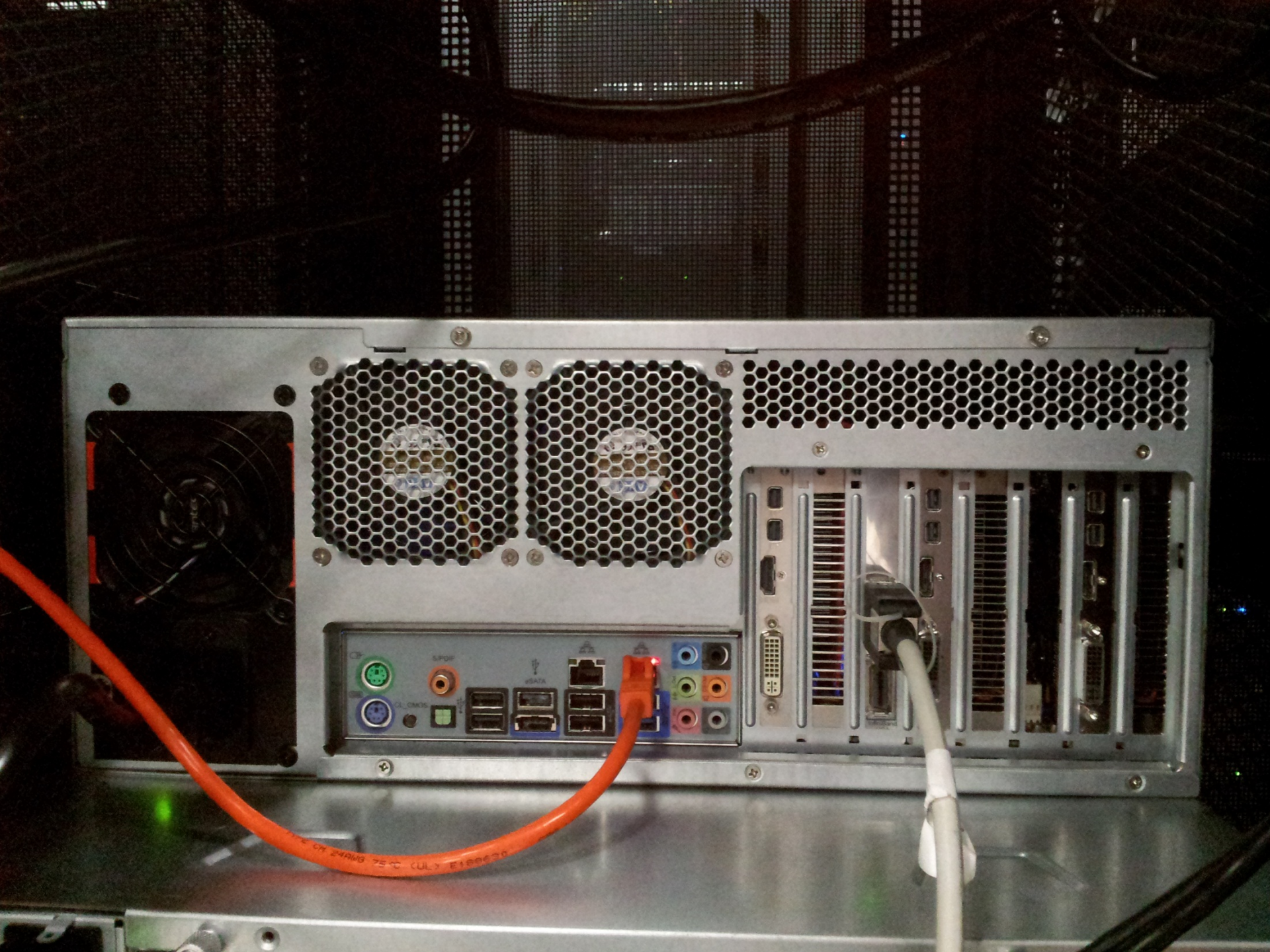
4x SDR Infiniband interconnect

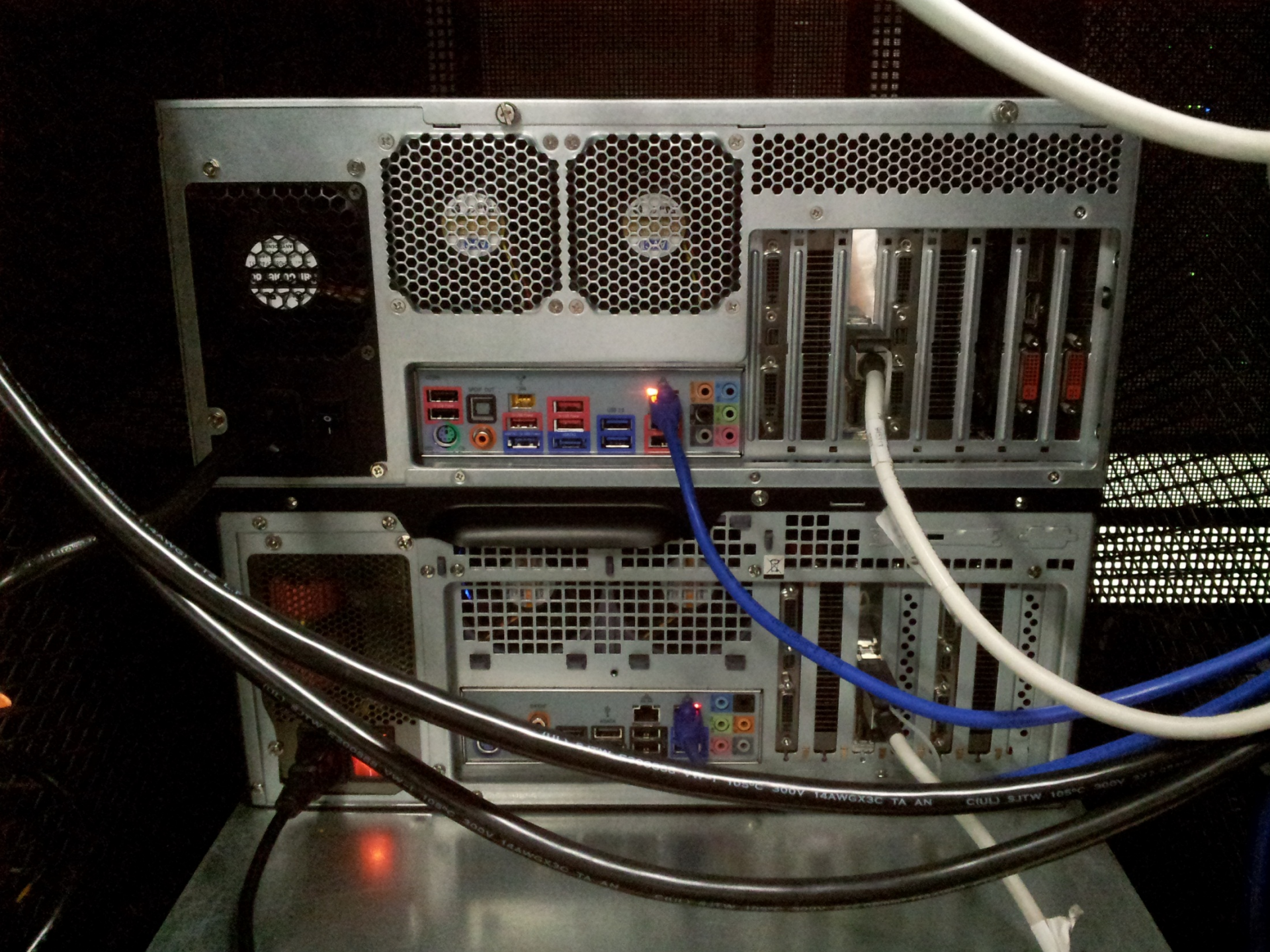
7kW of electricity

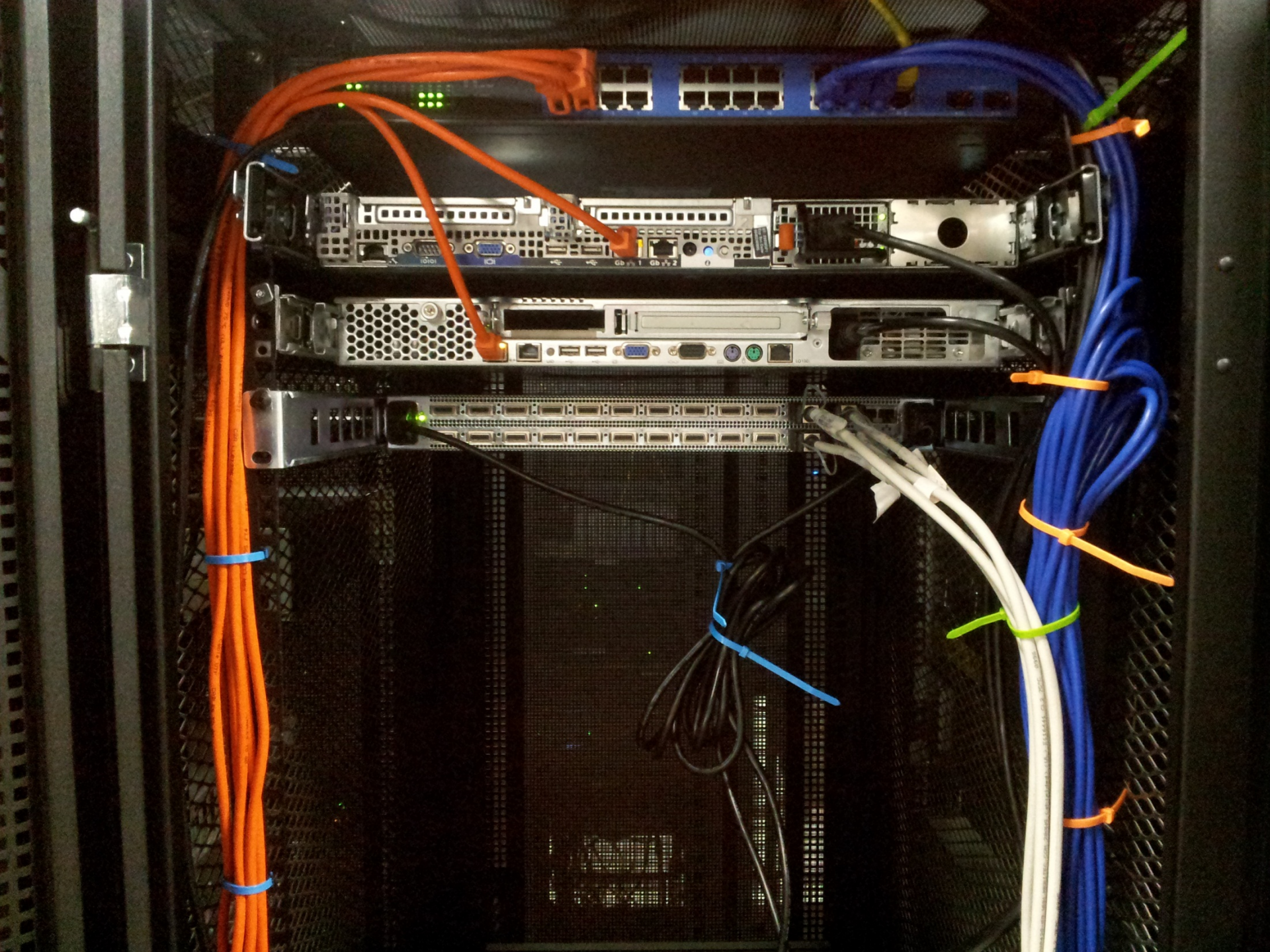
Broker daemon runs on a cluster node

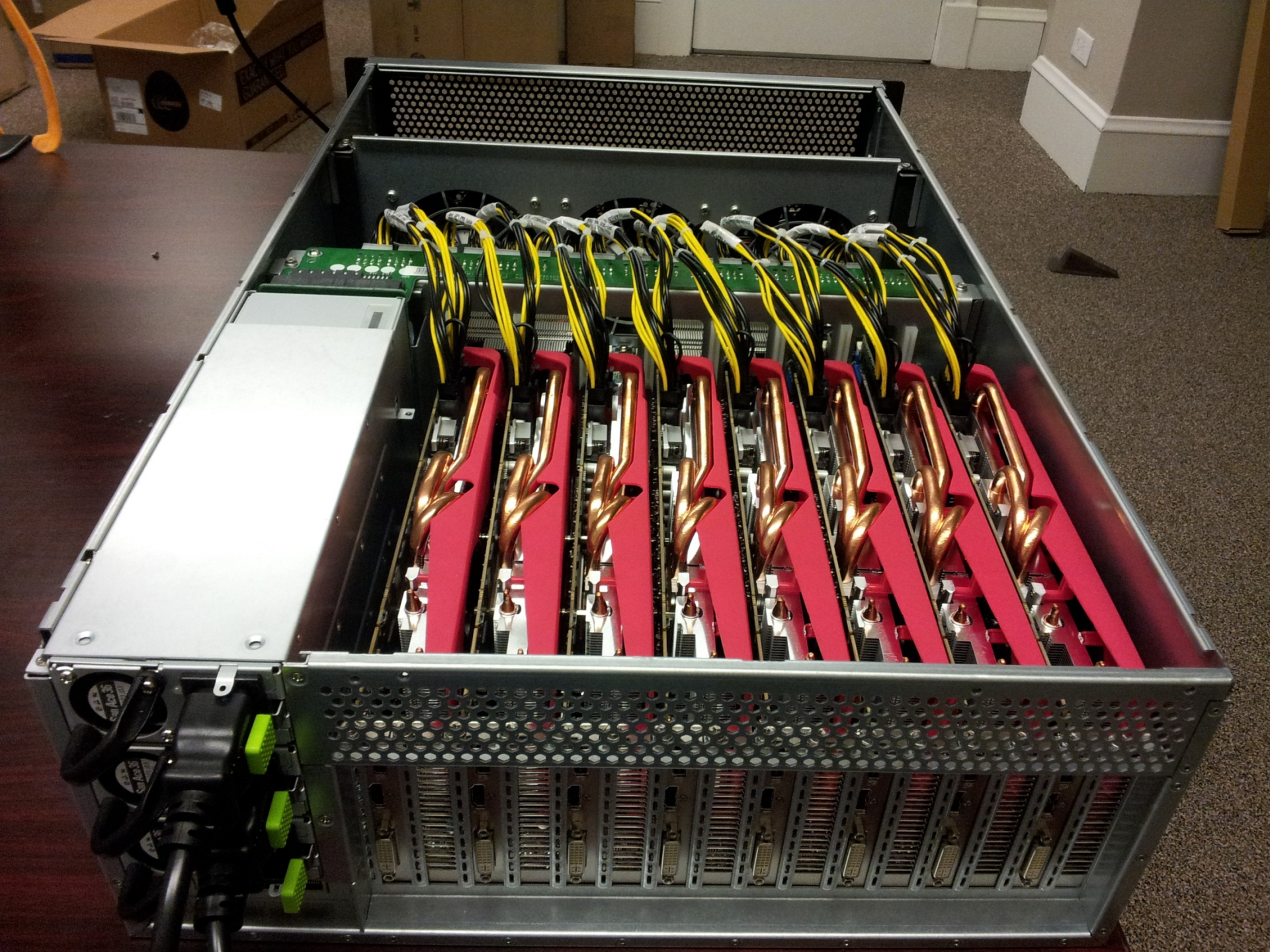
```
epixoip@token:~/oclHashcat-lite-0.11$ LD_LIBRARY_PATH=/usr/lib/vcl vclrun \
./vclHashcat-lite64.bin -b --benchmark-mode 1 –force

oclHashcat-lite v0.11 by atom starting...

Password lengths: 1 - 54
Watchdog: Temperature abort trigger disabled
Watchdog: Temperature retain trigger disabled
Device #1: Tahiti, 2048MB, 1100Mhz, 32MCU
Device #2: Tahiti, 2048MB, 1100Mhz, 32MCU
Device #3: Cypress, 512MB, 830Mhz, 20MCU
Device #4: Tahiti, 2048MB, 1100Mhz, 32MCU
Device #5: Cypress, 512MB, 830Mhz, 20MCU
Device #6: Tahiti, 2048MB, 1100Mhz, 32MCU
Device #7: Cypress, 512MB, 830Mhz, 20MCU
Device #8: Tahiti, 2048MB, 1100Mhz, 32MCU
Device #9: Cypress, 512MB, 830Mhz, 20MCU
Device #10: Cayman, 1024MB, 880Mhz, 24MCU
Device #11: Tahiti, 2048MB, 1100Mhz, 32MCU
Device #12: Cypress, 512MB, 830Mhz, 20MCU
Device #13: Cayman, 1024MB, 880Mhz, 24MCU
Device #14: Tahiti, 2048MB, 1100Mhz, 32MCU
Device #15: Cypress, 512MB, 830Mhz, 20MCU
Device #16: Cayman, 1024MB, 880Mhz, 24MCU
Device #17: Tahiti, 2048MB, 1100Mhz, 32MCU
Device #18: Cypress, 512MB, 830Mhz, 20MCU
Device #19: Cayman, 1024MB, 880Mhz, 24MCU
Device #20: Tahiti, 2048MB, 1100Mhz, 32MCU
Device #21: Cypress, 512MB, 830Mhz, 20MCU
Device #22: Cayman, 1024MB, 880Mhz, 24MCU
Device #23: Tahiti, 2048MB, 1100Mhz, 32MCU
Device #24: Cypress, 512MB, 830Mhz, 20MCU
Device #25: Cayman, 1024MB, 880Mhz, 24MCU

[s]tatus [p]ause [r]esume [q]uit =>
```











Bandwidth Measurements

Brute force consistently uses < 8 Mbps

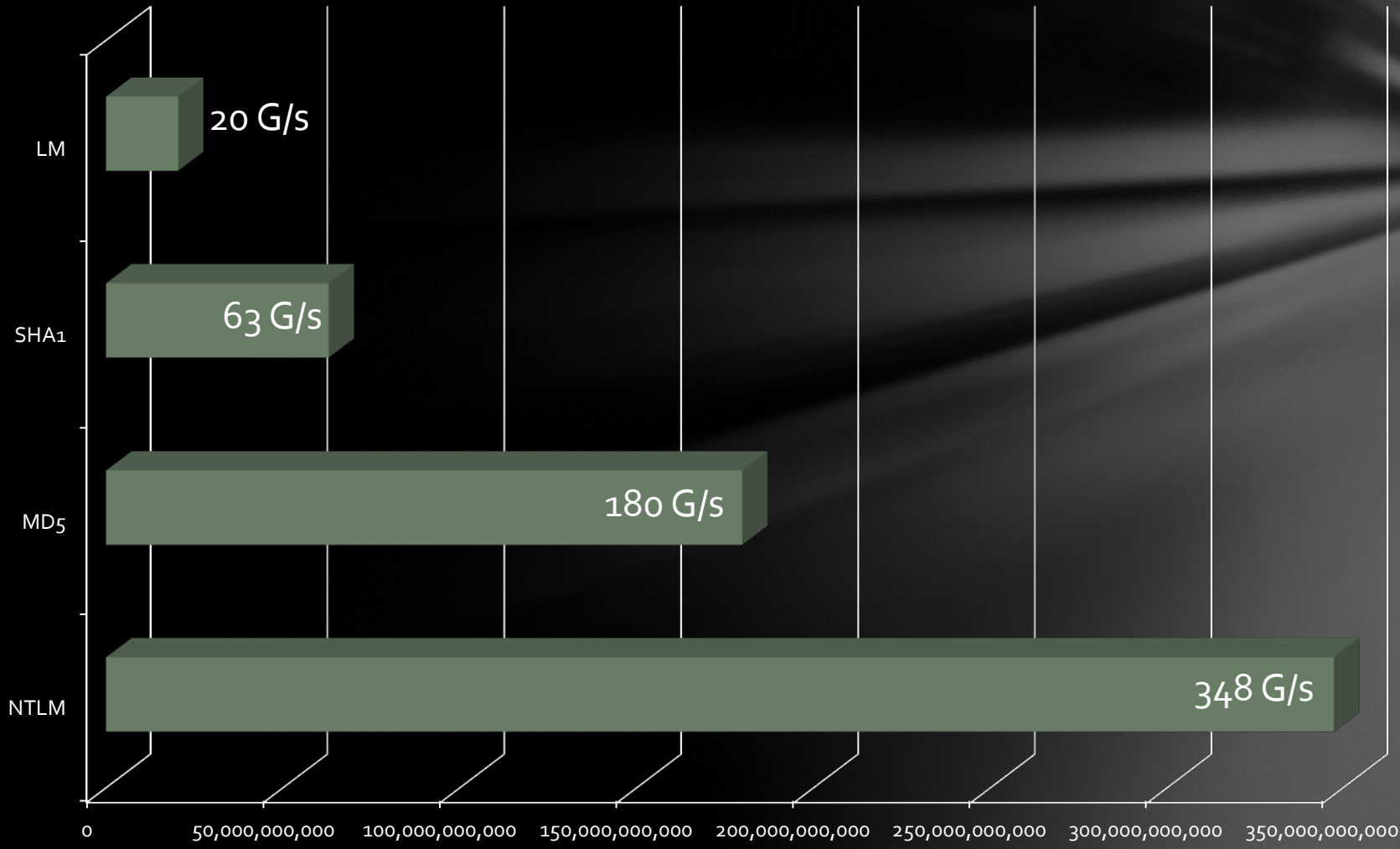
Wordlist attacks on fast hashes use no more than 8oo Mbps

Average peak of 88 Mbit per physical card

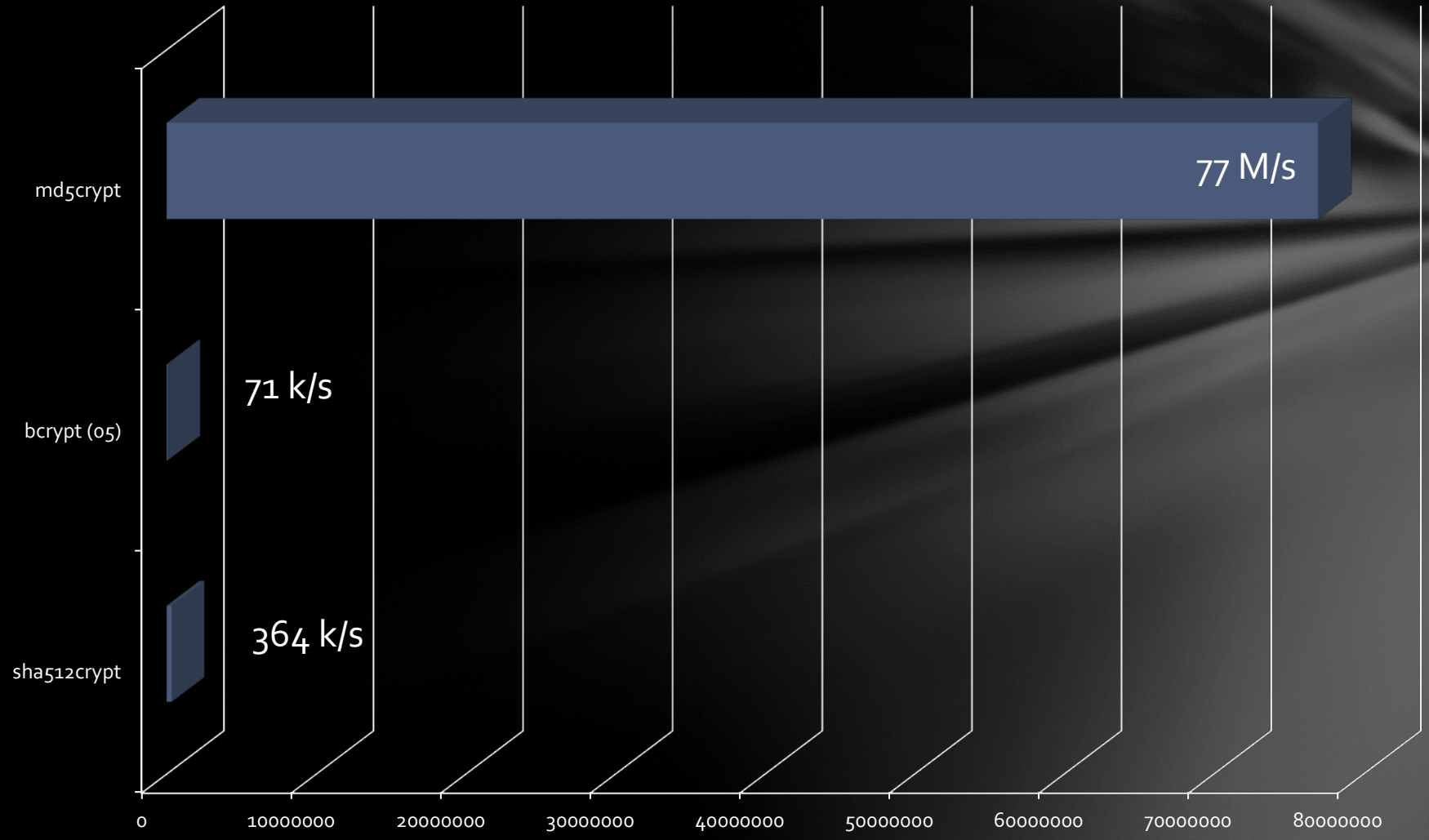
Ethernet latencies are still an issue

- Infiniband helps tremendously

Benchmarks – Fast Hashes



Benchmarks – Slow Hashes



Keeping in touch

IRC: epixoip on EFnet, Freenode

Twitter: @jmgosney