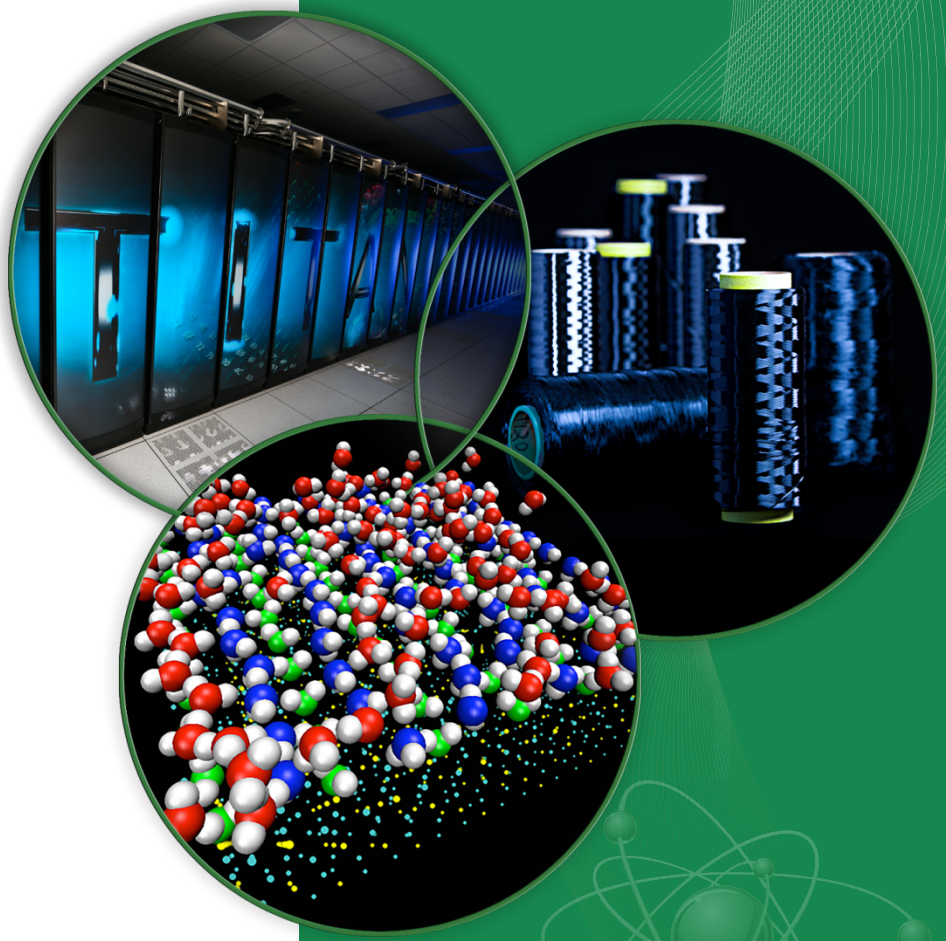
Scripts for InfiniBand Monitoring



Blake Caldwell

April 30, 2015

ORNL is managed by UT-Battelle
for the US Department of Energy

OAK RIDGE National Laboratory | OAK RIDGE LEADERSHIP COMPUTING FACILITY

Monitoring tools

- IB Health Check
- D3.js visualizations
- Perfquery balancing

Available on github:

https://github.com/bacaldwell/

IB Health Check

- Host-based shell script
- Designed to be run from snmpd or from CLI
- Returns Nagios status code
  - 0: OK
  - 1: WARNING
  - 2: CRITICAL
  - 3: UNKNOWN

IB Health Check

- HCA and local link health
  - Local errors check (HCA port)
  - Remote errors check (switch port)
- PCI width/speed of each HCA
  - Identify failed hardware or firmware issues
  - Appropriate slot placement
- Port in up/active state
- Link speed/width matches capability
- SM lid is set

IB Health Check Output

- CLI output

  Two interfaces, no errors:

```
mlx4_1-ib0:OK ;; mlx4_1-ib1:OK ;;
```

  Two interfaces, LinkDownedCounter above threshold triggers warning:

```
mlx4_1-ib0:WARNING - Direct-attached ddn-d-1 HCA-1 port 2: [LinkDownedCounter == 48] ;; \
mlx4_1-ib1:WARNING - Direct-attached ddn-d-0 HCA-1 port 2: [LinkDownedCounter == 66] ;;
```

- Nagios view

Host	Service	Status	Last Check	Duration	Attempt	Status Information
ninja17	IB_HEALTH	CRITICAL	08:55:47	3d 10h 12m 35s	10/10 #2	mlx4_0-ib0:CRITICAL - Link rate: 40Gbps is inappropriate for FDR IB ::

select all (hosts) - unselect all - all problems - all with downtime

Host	Service	Status	Last Check	Duration	Attempt	Status Information
atlas-spare03	IB_HEALTH	CRITICAL	08:58:21	4d 21h 36m 7s	10/10 #20	mlx4_1-ib0:CRITICAL - Link rate: 20Gbps is inappropriate for FDR IB - IB HCA errors for "atlas-spare03 mlx4_1" GUID 0x2c90300185511 port 1: [SymbolErrorCounter == 5583] ::

select all (hosts) - unselect all - all problems - all with downtime

Configuration

- /usr/local/etc/monitor_ib_health.conf

```
oss:mlx4_0:1:56
oss:mlx4_0:2:56
oss:mlx4_1:1:56
mds:mlx4_0:1:56
```

- /usr/local/etc/ib_node_name_map.conf

```
# Compute nodes
0x0002c903001c5e31      “oss1”
0x0002c90300dde441      “oss2”
# Core switch
0x0002c903008726a0      “ibsw-core1 Spine 1”
0x0002c903008726b0      “ibsw-core1 Spine 2”
0x0002c903008726c0      “ibsw-core1 Spine 3”
0x0002c903008476d0      “ibsw-core1 Line 1”
0x0002c90300847790      “ibsw-core1 Line 2”
0x0002c903008477f0      “ibsw-core1 Line 3”
```

D3.js visualizations

- Explored pydot for visualizations IBUG 2013
  - Too much information, wanted ability to collapse by switch (or chassis)
- Discovered of D3.js from Florent's IBUG 2014 presentation
  - JavaScript library written by Mike Bostock of New York Times
- Demo
  - https://github.com/bacaldwell/ib_d3viz

ib_d3viz Workflow

Create ibnetdiscover output file:

```
ibnetdiscover -p --node-name-map > fabric_ibnetdisc.out
```

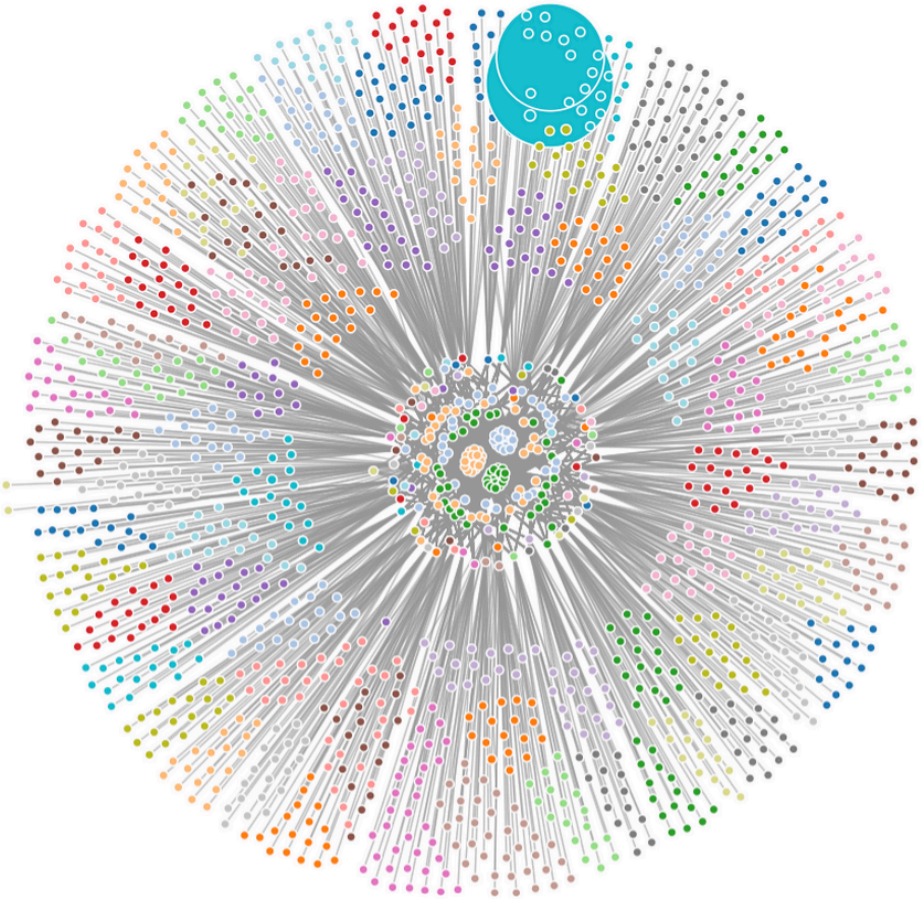
Create JSON file describing a graph:

```
python ib_topology_graph.py -g fabric_graph.json fabric_ibnetdisc.out
```

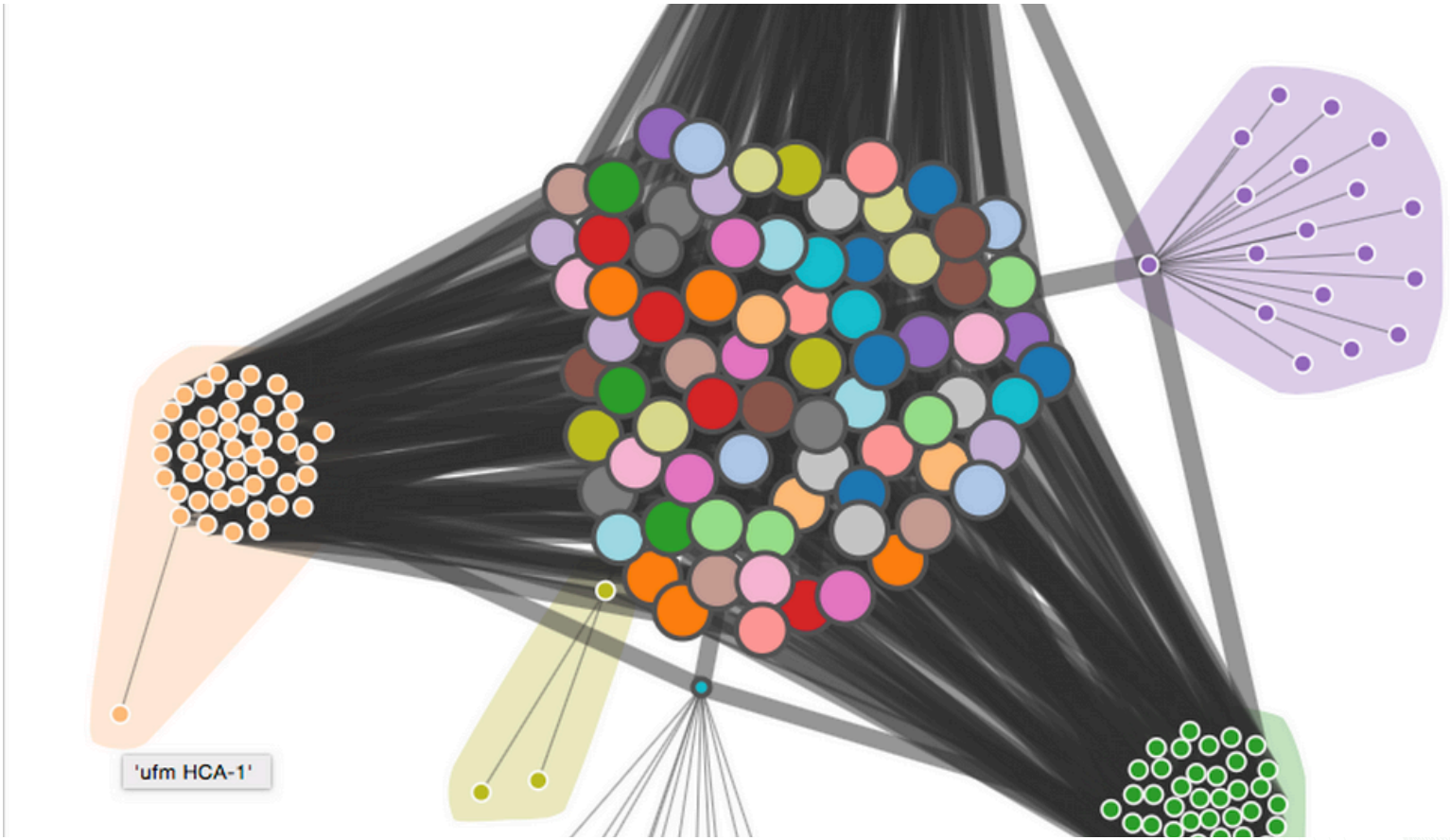
Creates JSON file describing a graph:

```
…
  “links”: [
    { “source”: 0,
      “target”: 1320,
…
   ],
  “nodes”: [
    { “group”: 0,
      “name”: “’node1601 HCA-1’”,
…
```

Full topology with data overlay



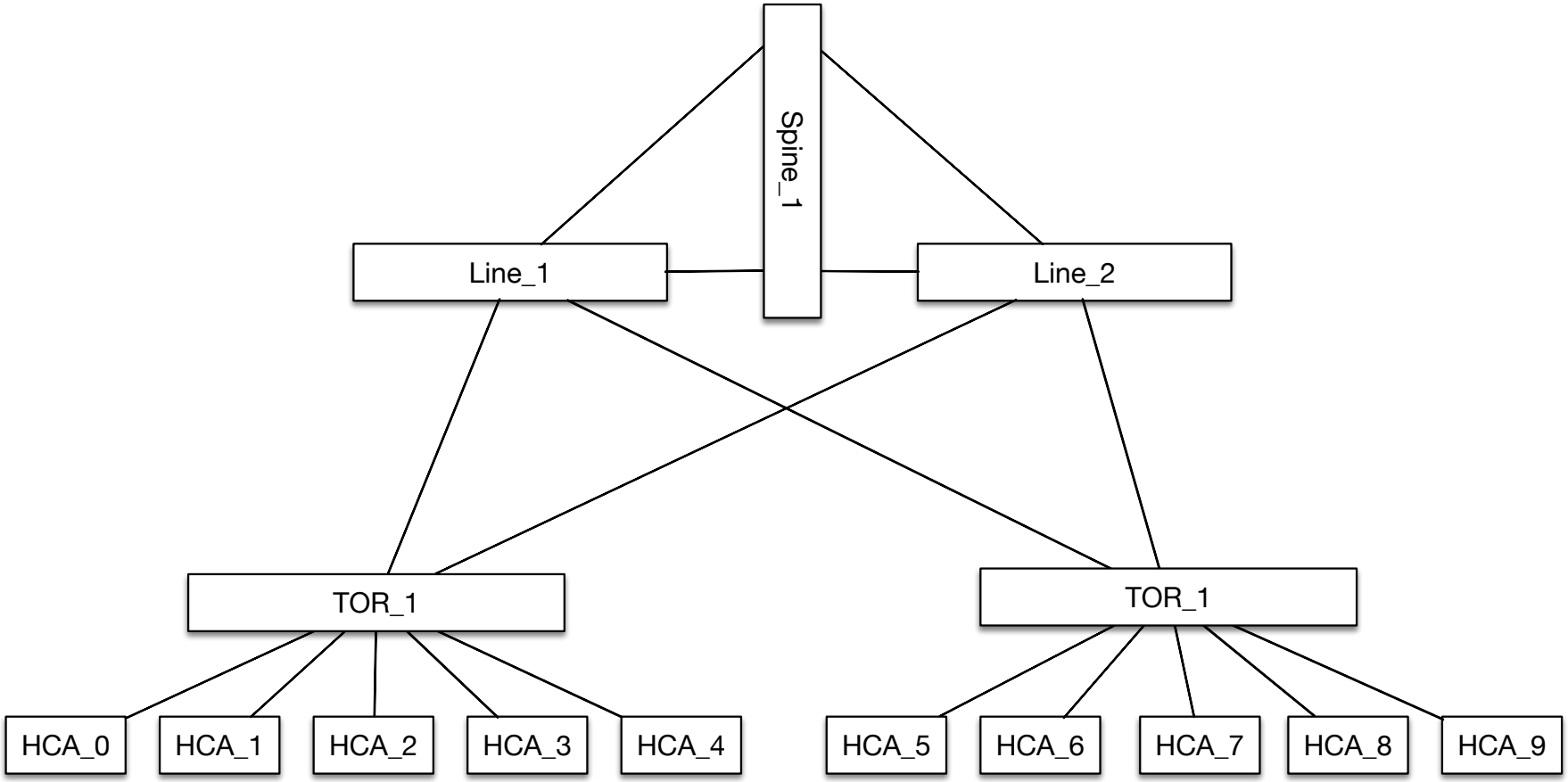
Clustered graph with collapsible nodes



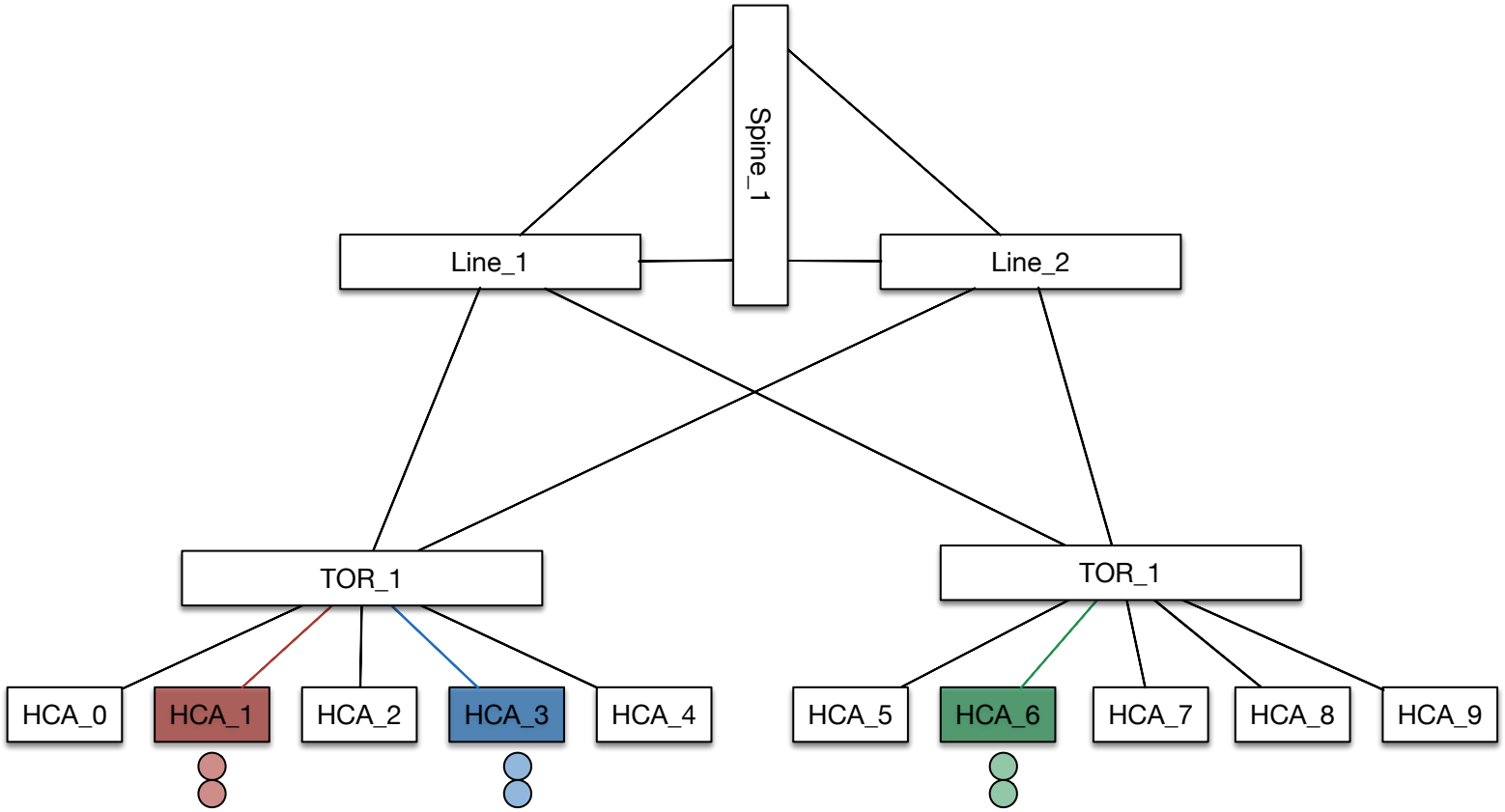
Balanced perfqueries

1. Ports that are the direct neighbor of a node will always get assigned to that node for querying.
2. All link endpoints that make up paths between the all nodes part of the current job are identified. The number of links identified at this step relates to the number of nodes in the job, but grows very quickly nearing the total number of links in the fabric.
3. These links are allocated to nodes that are found on in the switch's forwarding table (LFT) for that link's port on the switch. This is in the reverse direction from the perfqueries, so routes will not be completely balanced. However, the hop count will remain minimized in the reverse direction.
4. From step 3 some imbalances will arise, but are kept in check by a static threshold of a difference of 5 between the node assigned the least number of link endpoints to query and the most loaded node. If the node picked by step 3 is past the threshold, the allocation script will instead choose the node with the least number of LID/ports assigned to it.
5. The per-node list of link endpoints to query is written to a file to be read later by each node when starting the actual queries.

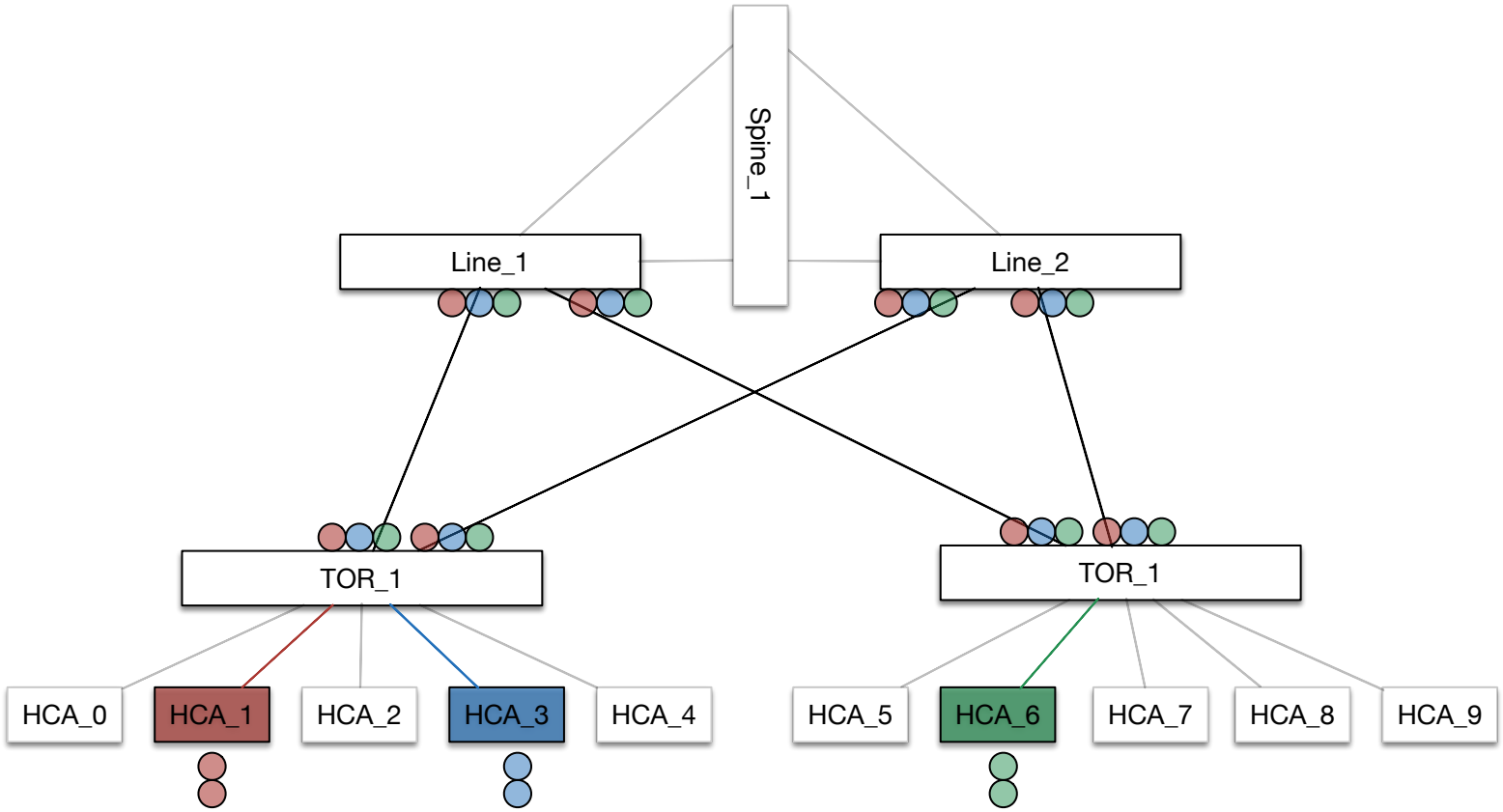
Distribution of queries



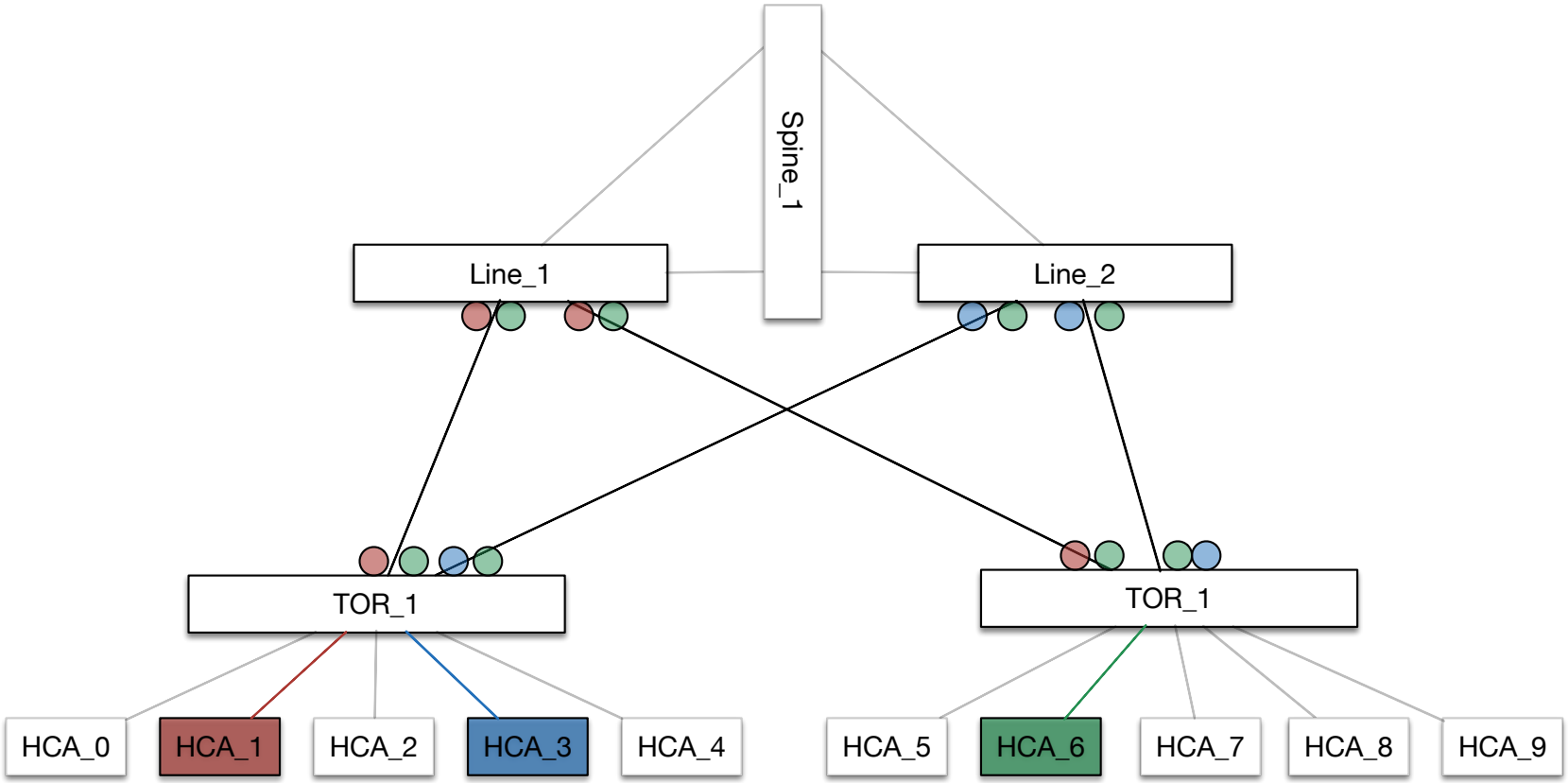
Start with local port and direct neighbor



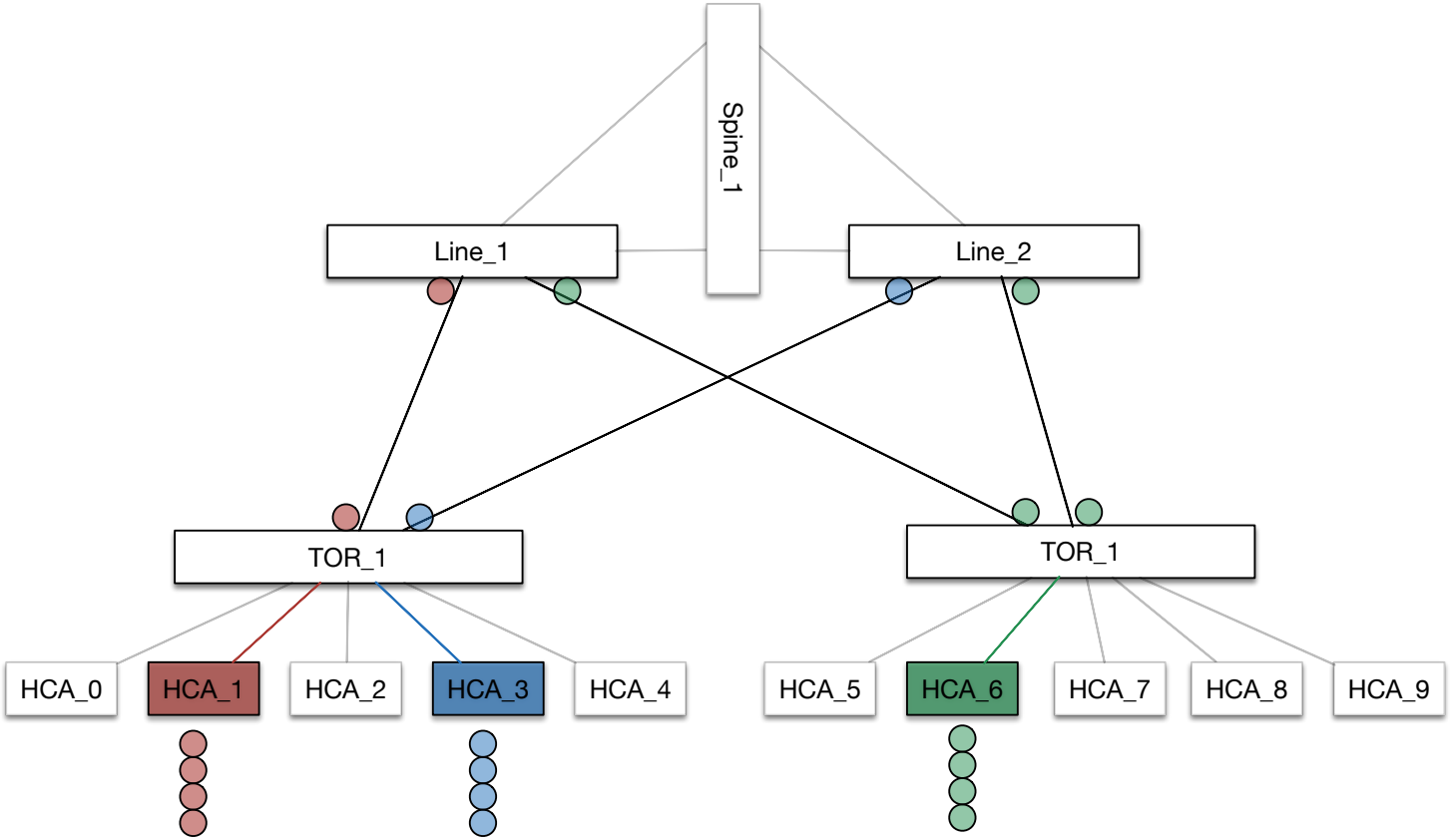
Identify what else needs to be queried



Use lft to prune “far” HCAs



Take per-node list and run perfquery



Summary

- InfiniBand monitoring tools on github:

  https://github.com/bacaldwell

  1. scalable-monitoring/monitor_ib_health
  2. Ib_d3viz
  3. balanced-perfquery

- Please contribute! ☺