ALICE

HLT NETWORK

PROGRESS PLAN

1. PHYSICAL — Move the switches physically.

2. LOGICAL — Re-design the IP-layout, using subnets and VLANs.

3. CONFIGURATION — Configure the equipment properly. Redundancy.

4. MONITORING — Monitor the network to find errors quickly.

5. DOCUMENTATION — Document the network thoroughly.

PHYSICAL

- Move the switches physically to the racks where the nodes are (just as the CHARM/ IPMI-switches). This will make it much easier to troubleshoot physical network-problems (follow the cable from the node; voila, you're at the switch — no cable-mess).

- This removes a source of error; the patch-panels & patch-cables.

- 2 racks share 1 switch.

- 2 switches in spare.

- Again; no cable-mess!

LOGICAL

- We have been assigned 10.162.0.0/16, which is 2^{16} IP-addresses. It's enough.

- Divide that address-space into more suitable networks, one for each function of the network. We're currently at 10 subnets — we probably need more. Make them large enough to grow in (f.ex. 2x the size of what we *currently* need).

- Each subnet has it's own VLAN. Use the layer 3 functionality of the switches to route packets between the VLANs.

- Allocate IP-addresses on-the-fly (no static setup)

CONFIGURATION

• Upgrade firmware — the ones currently on the switches is from 2007. Latest firmware is from October 2010.

• One VLAN for each subnet. Use trunks to distribute these to all edge-switches.

• Use ACLs to control the traffic between the VLANs (security).

• Redundant and high-speed paths between edge- and core-switches (using link-aggregation).

CONFIGURATION

- Extra redundant and extra-high-speed paths between all the core switches (using link-aggregation and OSPF) — speed can be scaled easily by just adding cables.

- Per VLAN spanning tree. Works much more efficiently compared to the current setup, so potential loops are closed much faster.

- QoS. Prioritize important stuff (management, SSH, etc).

core0

core1

4x

2x

2x

4x

4x

ipmi/charm

2x

core2
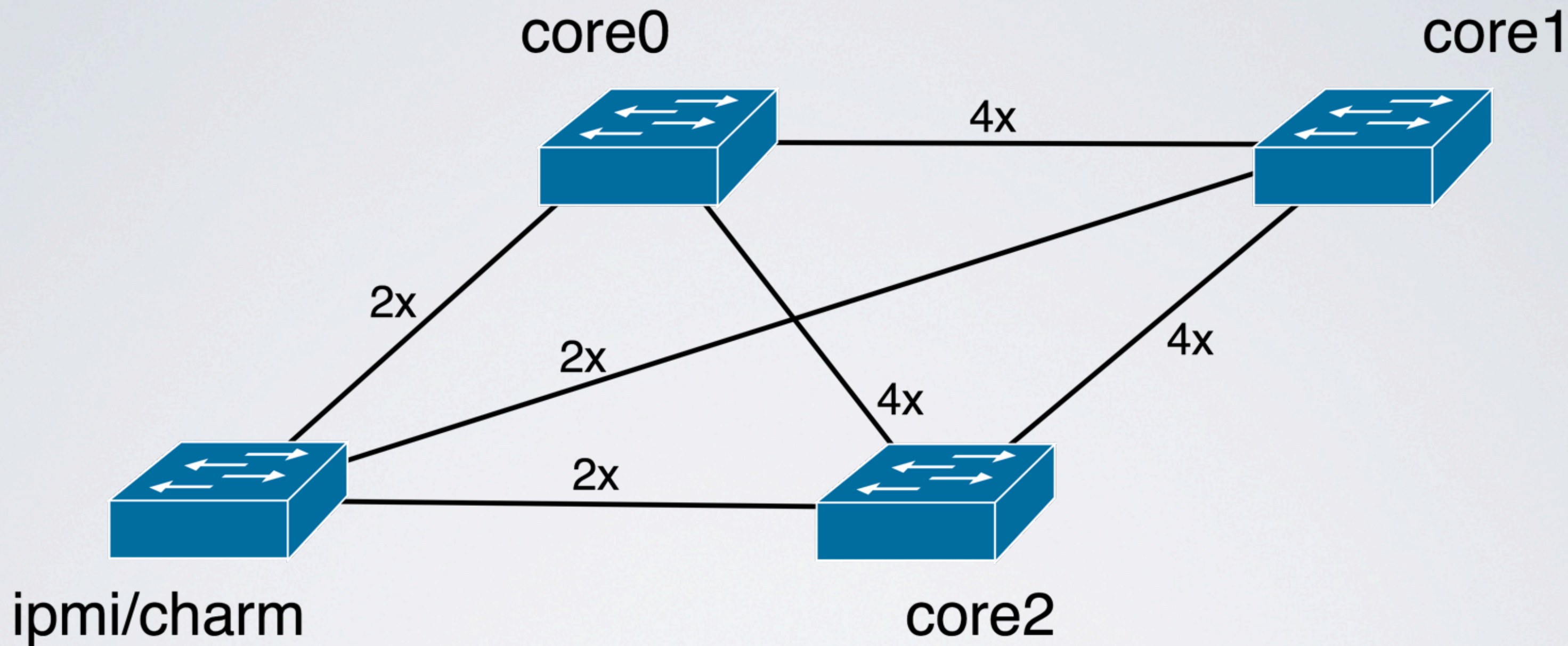
- Redundancy (multiple paths)
- Scalability
- Stability
- Quick convergency

MONITORING

- Setup a proper monitoring-system that works.

- Errors and performance-issues can be discovered fast.

- Performance-tweaks can be done based on these monitoring-utilities.

- We all ♥ graphs

DOCUMENTATION

- Important!

- Makes troubleshooting & maintenance *much* more easier for people who's not familiar with the network.

- Hopefully it prevents, to some degree, that someone messes up the network while working on it.

- Saves time (compared to the time people would scratch their heads if they had *no* documentation at all).