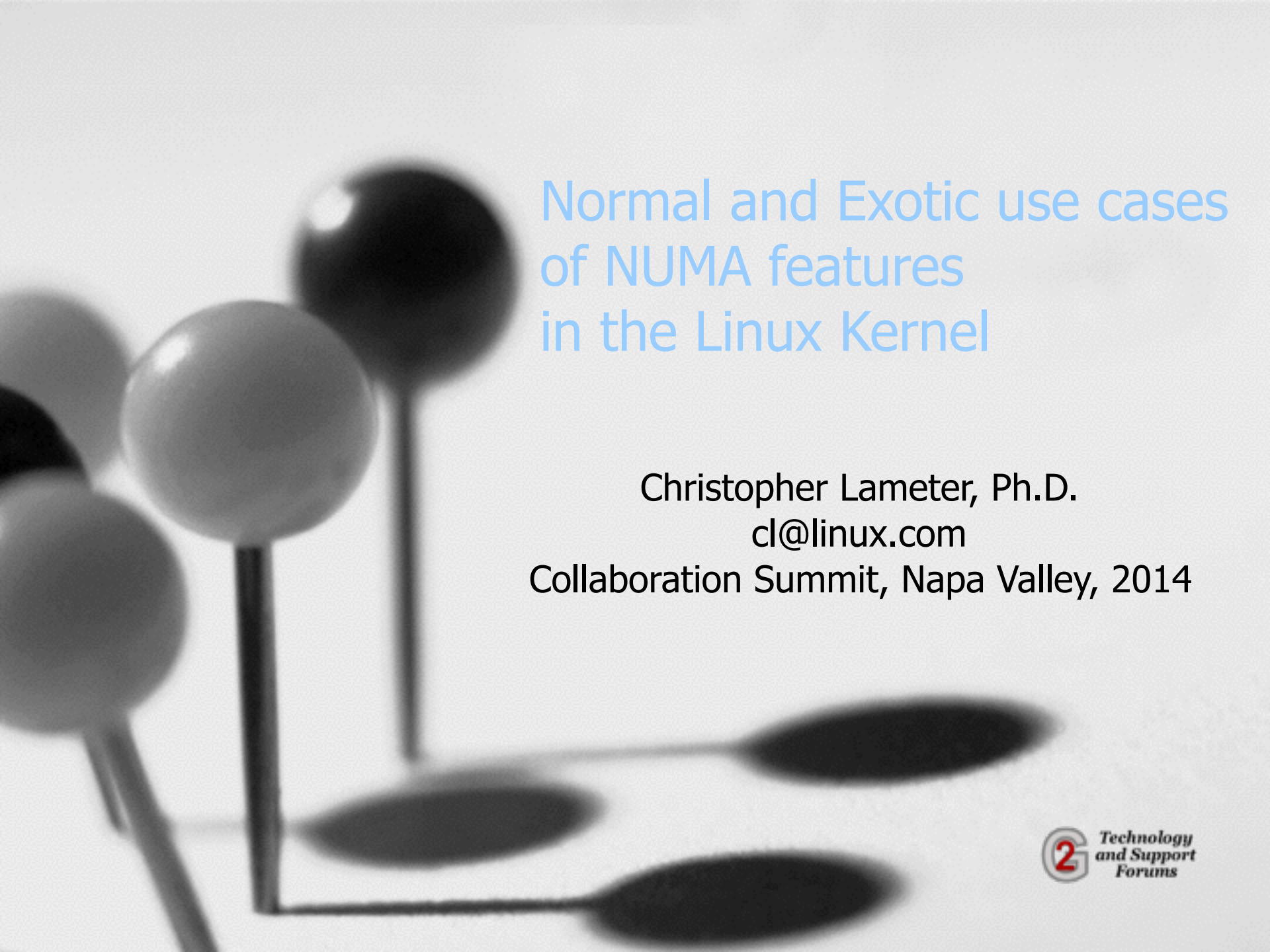


Normal and Exotic use cases of NUMA features in the Linux Kernel

Christopher Lameter, Ph.D.
cl@linux.com
Collaboration Summit, Napa Valley, 2014

Technology and Support Forums

Non Uniform Memory access in the Linux Kernel

- Memory is segmented into nodes so that applications and the operating system can **"place"** memory there.
- Regular uses

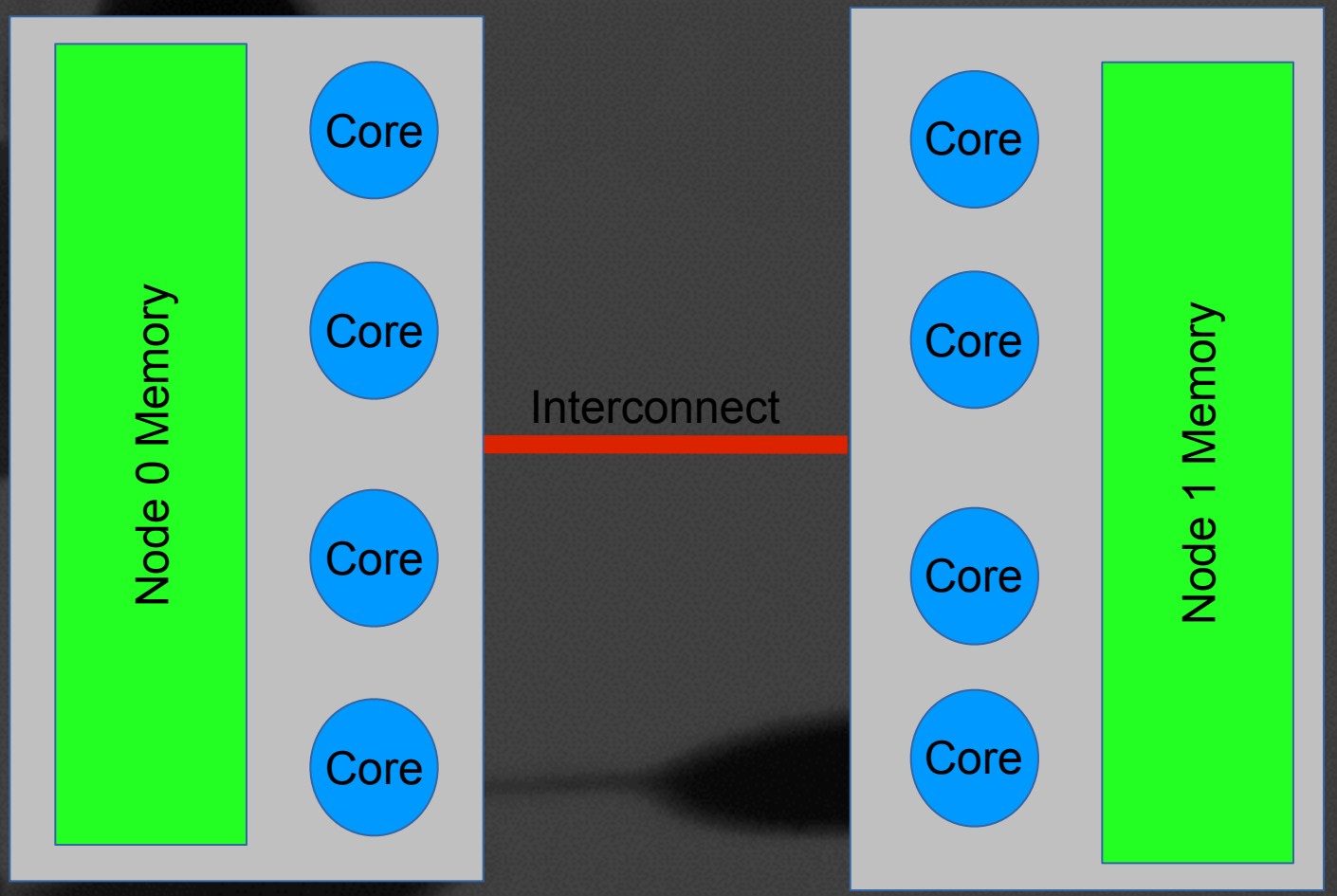
  Nodes have local processors and the system optimizes memory to be "local" to the processing. This may be automated (see the next talk on Autonuma) or manually configured through system tools and/or system calls of the application.
- Exotic uses

  Nodes may not have local processors, nodes may have other memory characteristics than just latency differences of regular memory. NVRAM, High Speed caches, Multiple nodes per real memory segments, central large node. Supports special innovative features.

Characteristics of Regular NUMA

- Memory is placed local to the processes accessing the data in order to improve performance.
- If computations or data requirements are beyond the capacity of a single NUMA node then data and processing need to be spread out.
- Interleaving to spread memory structures
- Balancing access to data over the available processors on multiple nodes.
- Automated by the NUMA autobalancer in newer Linux kernels otherwise manual control of affinities and memory policies is necessary for proper performance.

A minimal and typical 2 node NUMA system

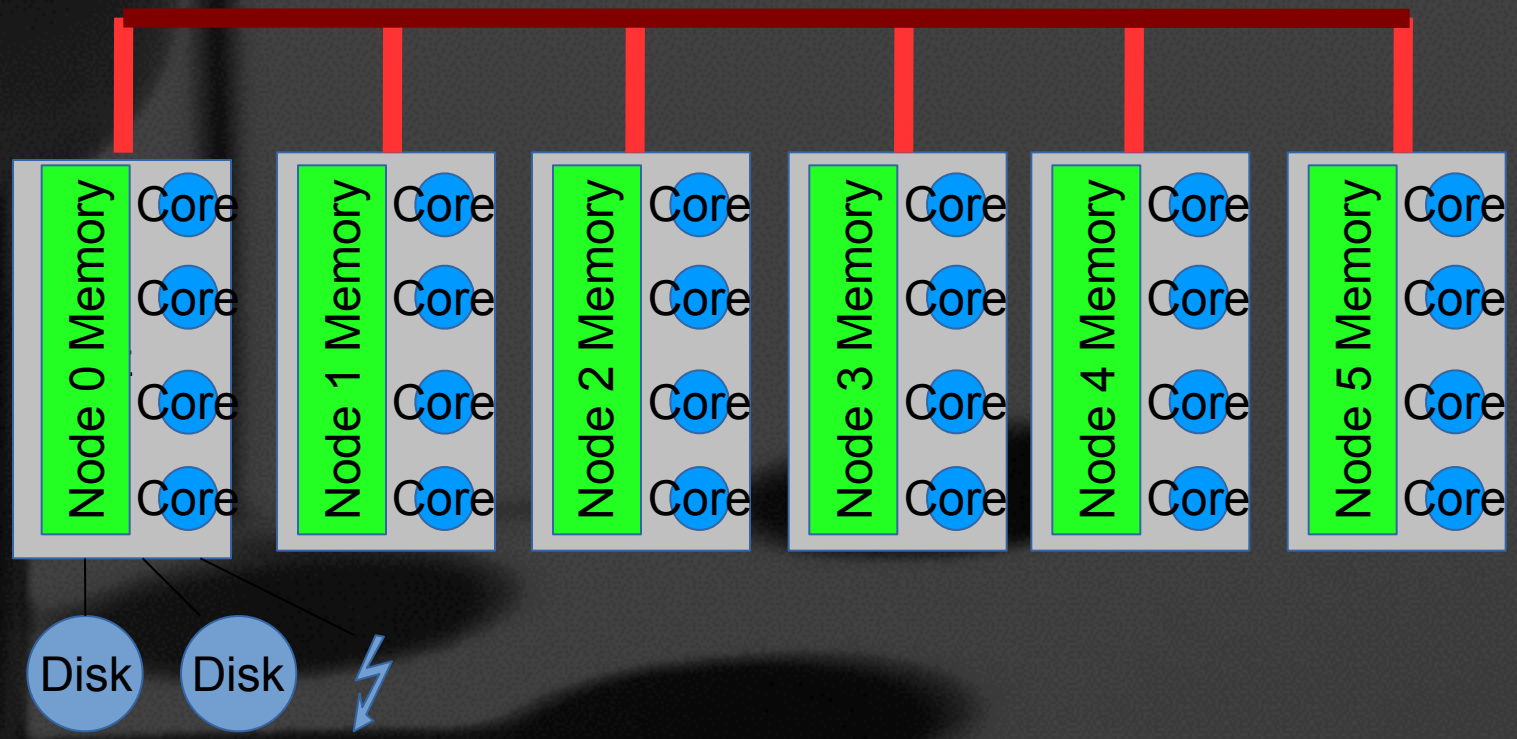


Regular NUMA use cases

- To compensate for memory latency effects due to some memory being closer to the processor than others.
- Useful to scale the VM for large amounts of memory. Memory reclaim and other processor occurs local to a node concurrent with the same actions on other nodes.
- I/O optimizations due to device NUMA effects
- Works best with a symmetric setup
  - Equal sizes of memory in each node
  - Equal number of processors
- Linux is optimized for regular NUMA and works quite well with a symmetric configuration.

Large Scale Regular NUMA configurations

- Scale out by adding more nodes with processors and memory
- I/O is often only connected to the first node or nodes
- An element of assymmetry even in a normal NUMA hardware configuration.

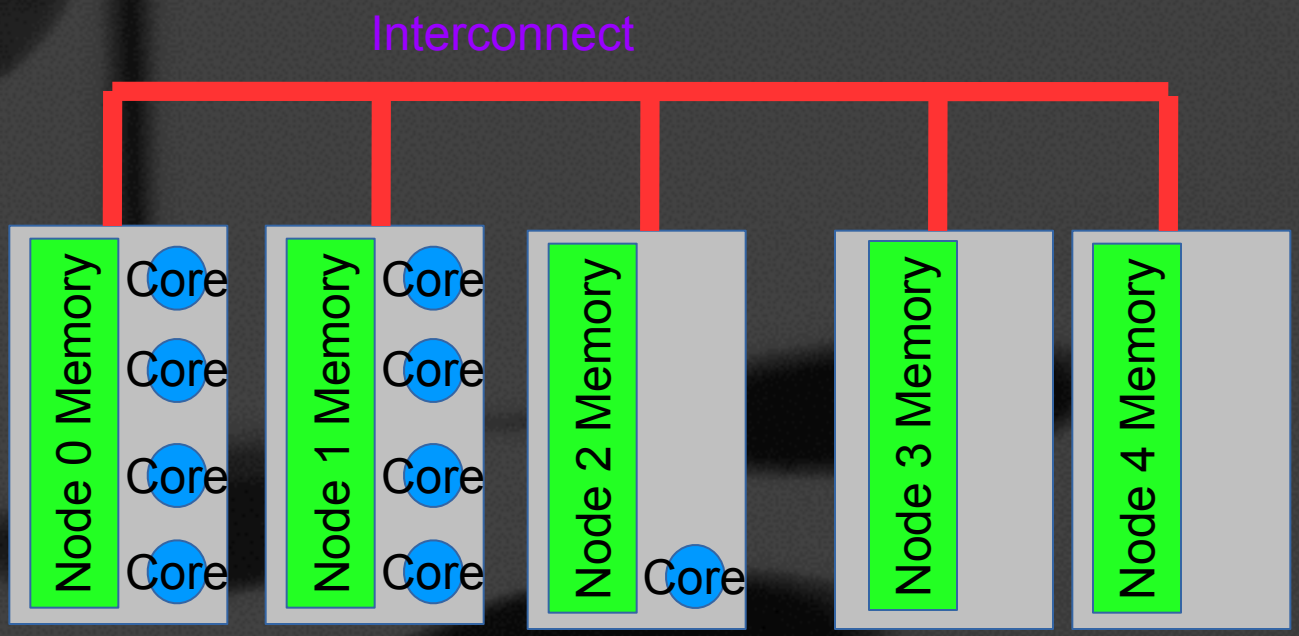


Exotic NUMA configurations

- And then people started trying to use NUMA for different things which resulted in additional challenges and IMHO strange unintended effects.
- Memory nodes. Add memory capacity without having to pay for expensive processors.
- Memoryless nodes. Add processing capacity without memory.
- Unbalanced node setups (some nodes have lots of memory, some nodes have more processors than others etc)
- Central Large Memory node. Applications all allocate from that. Only processors are grouped into NUMA nodes.
- Special High Speed nodes for data that needs to be available fast.
- Memory Segmentation for resource allocation (Google, precursor to cgroups).
- Memory hot add, hot remove

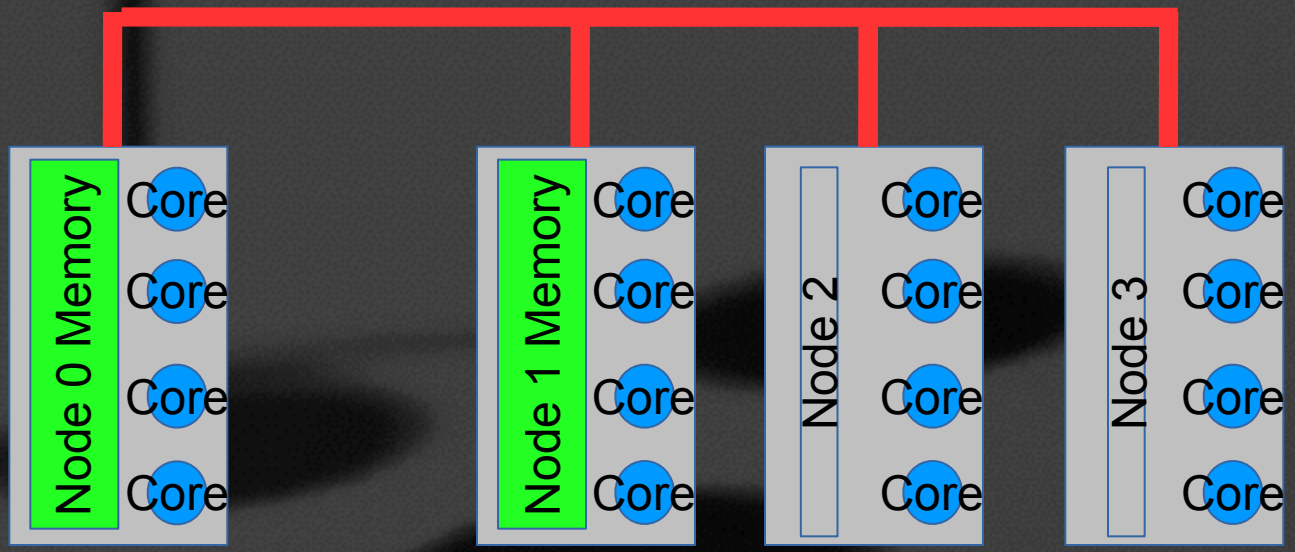
A Memory Farm to expand system capacity

- Nodes not populated with processors
- Expand the amount of memory the system supports
- The need to do reclaim and memory management remotely
- Ability to target allocations to Memory nodes
- Has been used to support extremely large memory capacities



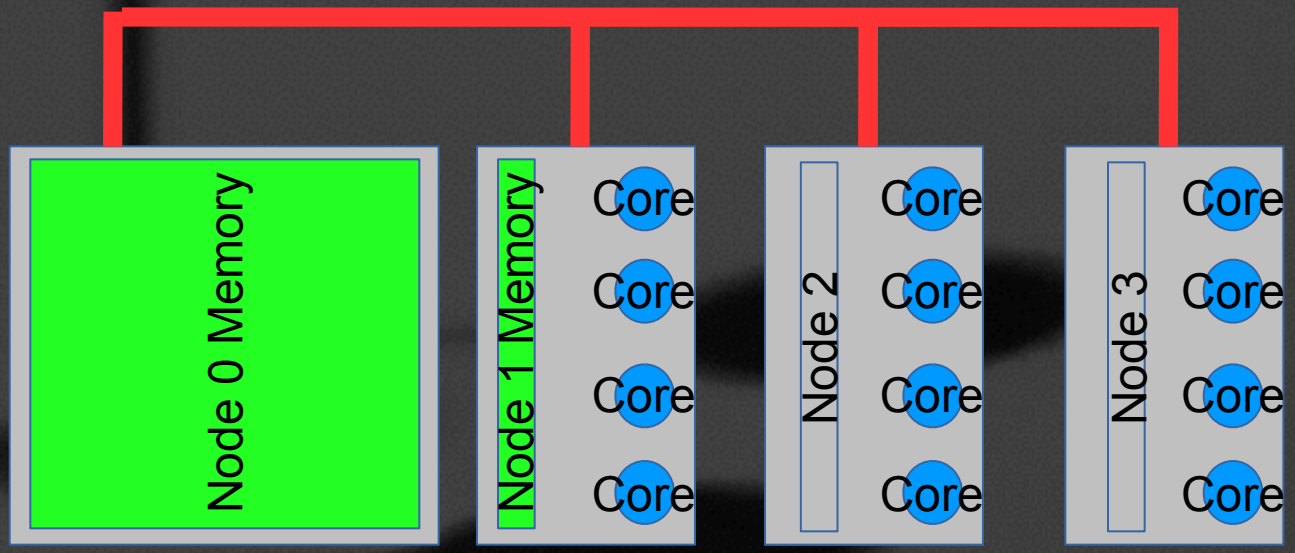
Memoryless Nodes

- Processors without memory
- Increase processing power
- Ineffective since there is no local memory



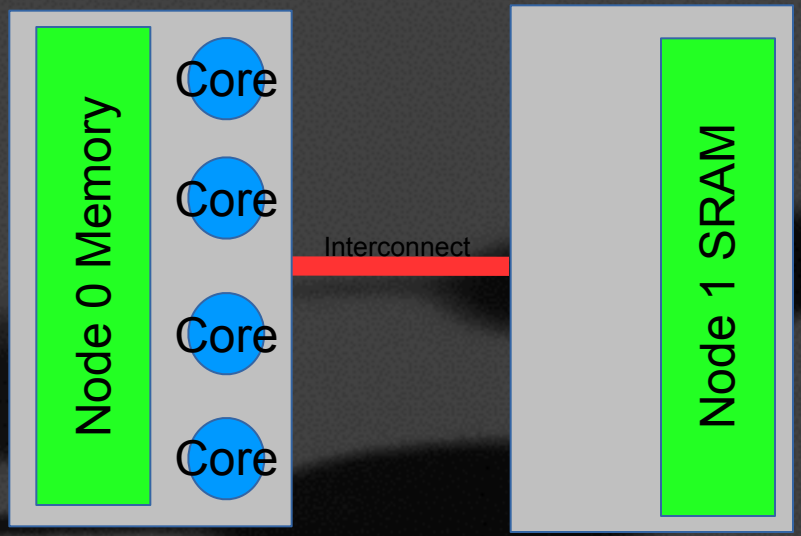
Big Memory Node Configuration

- One node with most of the memory available.
- Lots of nodes with processors
- Avoid NUMA configurations for applications. Everything on one large node.
- Maybe some small amount of memory on each node for efficiency.
- Need specialized memory configuration



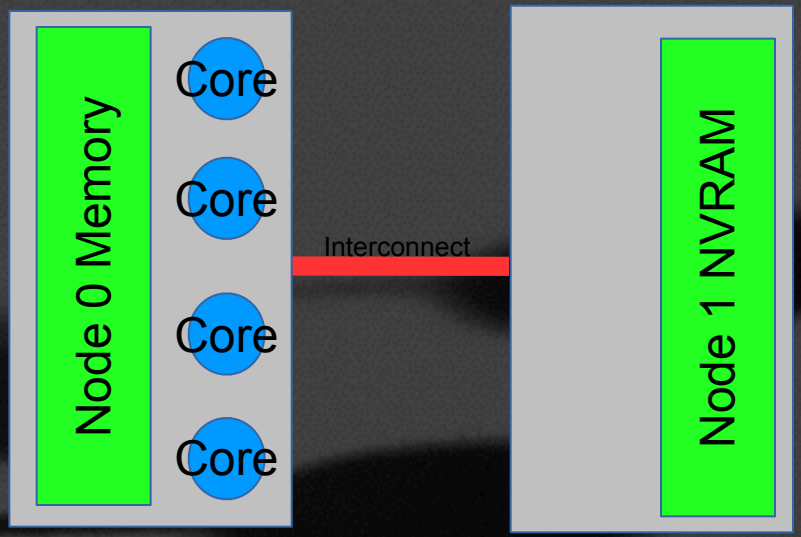
Small Fast node on the side

- For SRAM. Small capacity.
- Node is unused on boot. No local processors. Default allocs occur from Node 0.
- User manually request allocation on the SRAM node via memory policies or numactl.
- Danger of OS allocations on the node. Bring system up with the fast node down and online it when system is running. Therefore minimal OS data structures will be placed there.



NVRAM (Future?)

- A node with information that survives the reboot process.
- Node has no processors and is brought up after boot is complete.
- Problem of avoiding the OS initialization on bootup.
- Small area for bootup and then the rest could be a raw memory device?



Conclusion

- Questions
- Suggestions
- Advice
- Ideas?