Simplified Interface to Complex Memory

Sean Williams[1]

Michael Lang[2], Latchesar Ionkov[2]

[1]New Mexico Consortium, [2]Los Alamos National Laboratory

LA-UR-17-27387

Emerging technologies

- Intel Xeon Phi
- NVLink
- Gen-Z
- 3D XPoint

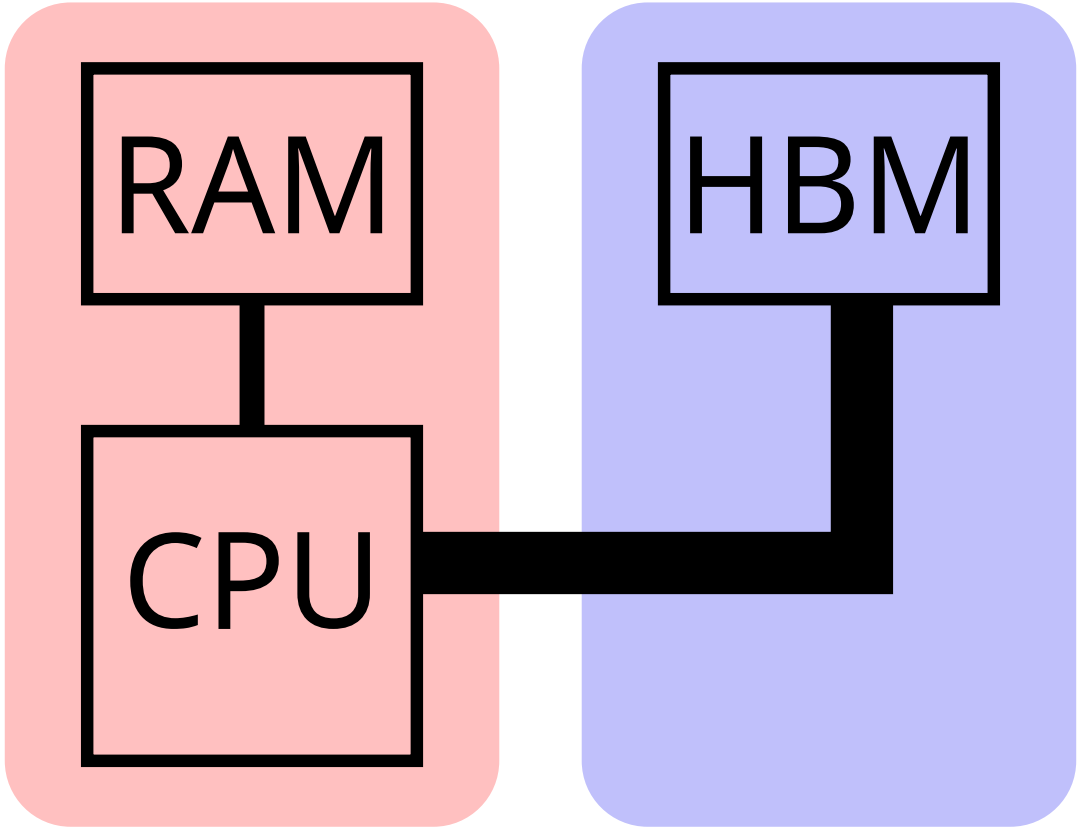
Problems

- Exposure of heterogeneous memory
  - NUMA?
  - Block devices?
  - Exotic buses?
- Coordination between processes and threads
- Portability

Portability

- Allocate
- Deallocate
- Migrate
- Introspect
- Arbitration/coordination

NUMA on KNL



NUMA on KNL

- HBM “far away” from CPUs
- Cache mode for automatic handling
  - Can be fine
- `preferred` memory policy for manual handling
  - Works because only two memory systems
- Not exactly “distance”

New memory policy

Configurable distances between memory pairs, configured via sysfs:

```
$ cat /sys/devices/system/node/node1/ordering1
1 0 2 3
```

New memory policy

- Orderings become policies:
  - Ordering that prefers bandwidth
  - Policy that follows bandwidth ordering
  - Ordering that prefers latency
  - Policy that follows latency ordering
  - ...

Deciding the ordering

- Have hardware define multiple distances
  - Distance as measured by bandwidth
  - Distance as measured by latency
  - ...
- Have hardware define specs
  - Spec bandwidth
  - Spec latency
  - ...
- Derive specs empirically
  - Measure bandwidth on boot
  - Measure latency on boot
  - ...

Balancing allocations?

- Kernel solution?
  - Per-process, per-node caps on pages
  - Set via system calls
  - Orchestrated globally, e.g., MPI math
- User solution?
  - Custom heap allocator
  - Uses shared memory to coordinate
  - Bookkeeping can be tricky

Block devices?

- At present, data is mirrored in memory
- Basically just swap
- Swap improvements?

Files?

- Memory mapped via `/dev`
  - Currently used for shared memory
- Possible future implementation
- This is a userspace problem
- May require shared-memory allocator

Conclusion

- Need portable, simplified approachs to heterogeneous memory
- Memory policy for NUMA heterogeneity
  - Policies to implement custom orderings
  - User side is still libnuma
  - Compatible with memkind
- Shared heap allocator
  - Arbitrate between threads/processes
  - Manage memory on non-NUMA devices
- Swap improvements may be coming

Conclusion