Overview

- Categories
  - Default Actions
  - Simple Binding
  - Abstract Binding
- APIs/Designs
  - Default Actions
  - Simple Binding
  - Abstract Binding

Default Actions

- *CONFIG_DISCONTIGMEM*
- Memory Classes:
  - *User Memory*
  - kmalloc()/kmem_cache_alloc()
  - *Default Memory Binding*
- Replication:
  - *Kernel Text*
  - Kernel Modules
  - User Text
  - User Libraries
  - Read-Only Mappings
  - Read-Mostly Mappings?
- Per-Node kswapd
- Per-Node kreclaimd
- NUMA-Aware Scheduler!
- Kernel-Subsystem NUMAization
- Multipath I/O
- Page Migration?
- Process Migration
- I/O-Based Migration?
- NUMA-Aware Locking!
- NUMA-Aware Tools
- NUMA-Aware Development Tools
- Performance-Monitoring Tools

Simple Binding

- Topology Discovery!
- Topology Discovery: Dynamic Reconfig
- Bind Tasks to CPU(s)!
- Bind Tasks to Node(s)
- Bind Memory to Node(s)
- Bind Tasks to I/O Device(s) [later]
- Find Memory to I/O Device(s) [later]
- Tools and Commands

Abstract Binding

- Specify Memory Locality Domains
- Link Tasks Together
- Virtualize Resource Names
- Linking/Binding Hints

Topology Discovery

- Kanoj's /proc patch
  - /proc/machine
    - machine
    - machine/node0003
    - machine/node0003/cpu0007
    - machine/node0003/cpu0007/shcach0003
    - machine/node0003/cpu0006
    - machine/node0003/cpu0006/shcach0003
    - machine/node0003/memory0003
    - machine/node0003/memory0003/memsize

Topology Discovery, Cont.

- Programmatic interface:
  - int getcpu(numasubset_t *restrict)
  - int getnode(numasubset_t *restrict)
  - int cputonode(int node, numasubset_t *restrict)
  - int nodetocpu(int cpu, numasubset_t *restrict)
- numasubset_t will hopefully be closely related to Paul Jackson's struct cpumemset (perhaps identical).

Simple Binding

- C API:
  - int bindtocpu(int cpu, numasubset_t *restrict)
  - int bindtonode(int node, numasubset_t *restrict)
  - int setlaunch(numasubset_t *restrict)
  - int bindmemory(void *first, void *last, int policy)
    - MPOL_FIRST, MPOL_STRIPE, MPOL_RR
- *sh Commands:
  - runon *cpus command*
  - runonnode *nodes command*
  - Tools TBD
- /proc Interface?

Abstract Binding

- Memory Locality Domains
  - Operations TBD, overview in Paul Jackson's design document
- Link Tasks Together
  - CLONE_NODE at creation time
  - /proc to link existing tasks?
- Virtualize Resource Names
  - Addressed by Memory Locality Domains?
- Linking/Binding Hints
  - TBD