RED HAT®
ENTERPRISE LINUX®
DEVELOPERS GROUP

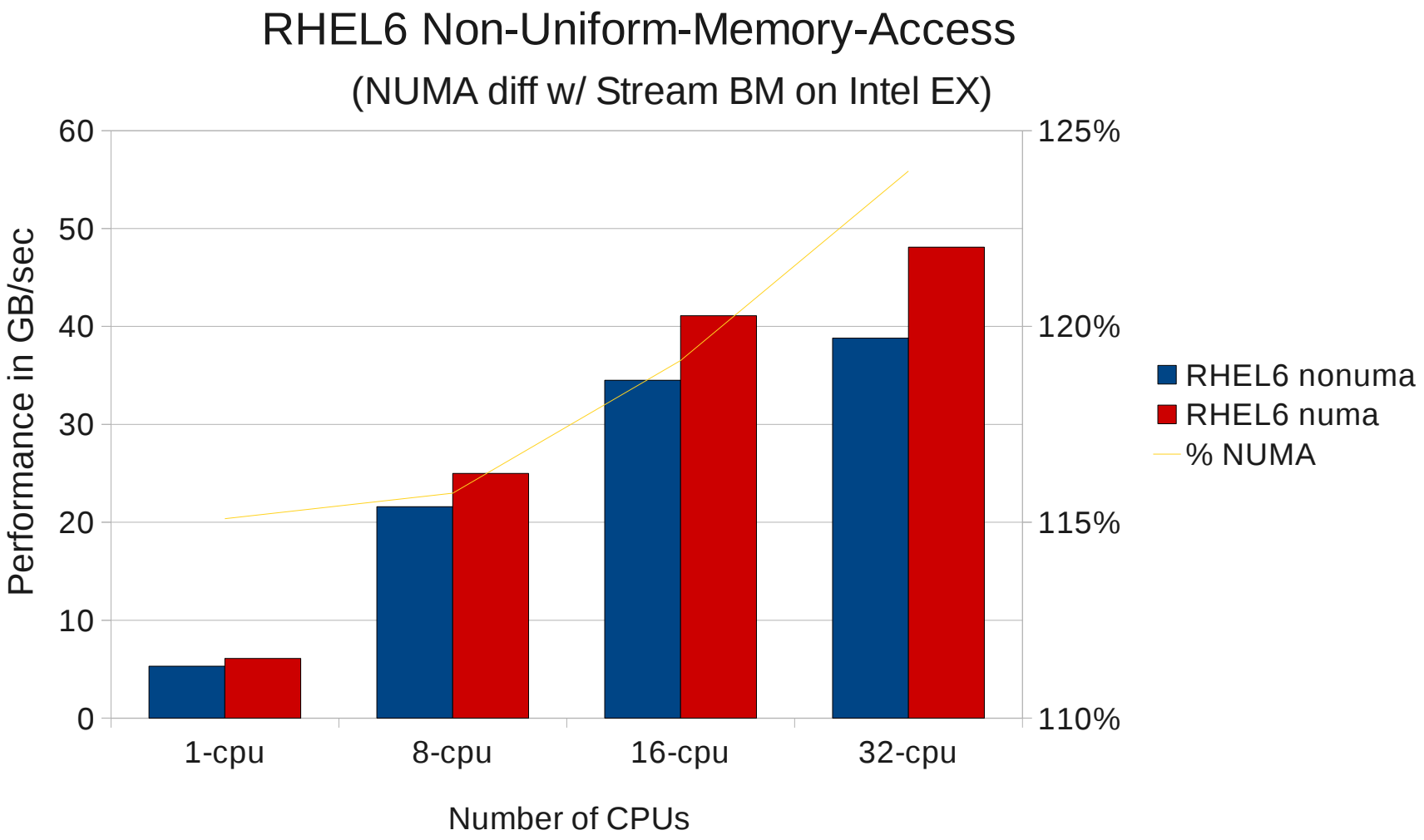
Using NUMA on RHEL 6

George Hacker
Curriculum Manager, Red Hat
06.26.12

What Is NUMA?

- UMA vs. NUMA
- What a node is
- Types of NUMA policy
  - Local (default)
  - Bound to specific memory nodes
  - Interleave
  - Preferred

RHEL 6 Scalability with NUMA



CLI Support for NUMA

- numactl(8)
- Provided by numactl package
- Examples
  - numactl --interleave=all program -opts args
  - numactl --cpubind=0 --membind=0,1 program -opts args
  - numactl --preferred=1
  - numactl --localalloc /dev/shm/file
  - numactl --show

CLI Support for NUMA (cont.)

- numastat
- /proc/sys/devices/system/node/*/meminfo
- cpuset cgroup
  - cpuset.cpus and cpuset.mems tunables

System Calls that Support NUMA

- Get/set memory policy
  - get_mempolicy(2), set_mempolicy(2)
- Get/set CPU affinity
  - sched_getaffinity(2), sched_setaffinity(2)
- Manipulate regions of memory
  - mbind(2), migrate_pages(2), move_pages(2)
- Typically system calls should not be used directly
  - Use functions provided by the NUMA library - libnuma

Introducing libnuma

- Provided by numactl-devel package
- Finer-grained control than numactl(8)
  - Per thread
  - Per memory address region

libnuma – How to Use the Library

- C source code must include numa.h header file
  - #include <numa.h>
- Link with the libnuma library
  - gcc -o program program.c -lnuma

libnuma – Getting Started in a C Program

- Confirm NUMA support before you do anything else

```
if (numa_available() < 0) {
        printf("numa_* functions unavailable\n");
        return 1;
}
```

- All other libnuma functionality is undefined when numa_available() returns an error

libnuma – General Information Functions

- How many memory nodes are there?
  - int numa_num_possible_nodes()
  - int numa_max_possible_node()
  - int numa_num_configured_nodes()
- How many CPUs?
  - int numa_num_configured_cpus()
- How large is a memory node?
  - long numa_node_size(int node, long *free)
  - long long numa_node_size64(int node, long long *free)

libnuma – General Information Functions (cont.)

- How large is a page of memory?
  - int numa_pagesize()
- Which NUMA node does a given CPU belong to?
  - int numa_node_of_cpu(int cpu)
- Which CPUs belong to a given NUMA node?
  - int numa_node_to_cpus(int node, struct bitmask *cpus)

libnuma – Bitmask Operations

- Data type: struct bitmask *
  - Size of the bitmask stored in a field of the structure
- Useful predefined bitmasks
  - numa_all_nodes_ptr
  - numa_no_nodes_ptr
  - numa_all_cpus_ptr
- Never modify the above bitmasks

libnuma – Bitmask Operations (cont.)

- How do you allocate a bitmask?
  - struct bitmask *numa_allocate_nodemask()
  - struct bitmask *numa_allocate_cpumask()
- How do you destroy a bitmask?
  - numa_free_nodemask(struct bitmask *bm)
  - numa_free_cpumask(struct bitmask *bm)
- Lower-level allocate/destroy functions
  - struct bitmask *numa_bitmask_alloc(unsigned int n)
  - numa_bitmask_free(struct bitmask *bm)

libnuma – Bitmask Operations (cont.)

- How do you initialize a bitmask?
  - numa_bitmask_clearall(struct bitmask *bm)
  - numa_bitmask_setall(struct bitmask *bm)
- How do you set and clear bits in a bitmask?
  - numa_bitmask_clearbit(struct bitmask *bm, int n)
  - numa_bitmask_setbit(struct bitmask *bm, int n)
- How do you copy a bitmask?
  - copy_bitmask_to_bitmask(bm_from, bm_to)
  - Both arguments are struct bm *

libnuma – Bitmask Operations (cont.)

- How do you check bits in a bitmask?
  - int numa_bitmask_isbitset(struct bitmask *bm, int n)
  - Returns the value of the specified bit in the bitmask
- How do you compare two bitmasks?
  - int numa_bitmask_equal(bm1, bm2)
  - Both arguments are struct bm *
  - Returns 1 when the bitmasks are equal, 0 when they are different

libnuma – NUMA Policy Operations

- Begin with struct bitmask *numa_get_mems_allowed()
- How do you set memory allocation policy to...
  - Local?
    - numa_set_localalloc()
  - Bound?
    - numa_bind(struct bitmask *nodemask)
  - Interleave?
    - numa_set_interleave_mask(struct bitmask *nodemask)
  - Preferred?
    - numa_set_preferred(int node)

libnuma – NUMA Policy Operations (cont.)

- Policy related queries
  - Bound
    - struct bitmask *numa_get_membind()
  - Interleave
    - struct bitmask *numa_get_interleave_mask()
  - Preferred
    - int numa_preferred()

libnuma – NUMA Allocation Functions

- How do you allocate memory...
  - Using the default policy?
    - void *numa_alloc(size_t size)
  - On the local node?
    - void *numa_alloc_local(size_t size)
  - On a specific node?
    - void *numa_alloc_onnode(size_t size, int node)

libnuma – NUMA Allocation Functions (cont.)

- How do you allocate memory... (cont.)
  - Interleaved on all nodes?
    - void *numa_alloc_interleaved(size_t size)
  - Interleaved on specific nodes?
    - void *numa_alloc_interleaved_subset(size_t size, struct bitmask *nodemask)
- How do you free allocated memory?
  - numa_free(void *start, size_t size)

libnuma – Other NUMA Functions

- How do you run on a specific node?
  - int numa_run_on_node(int node)
- How do you run on a group of specific nodes?
  - int numa_run_on_node_mask(struct bitmask *nodes)
- Which nodes can I run on?
  - struct bitmask *numa_get_run_node_mask()

libnuma – Other NUMA Functions (cont.)

- Put memory on the local node?
  - numa_setlocal_memory(void *start, size_t size)
- Put memory on a specific node?
  - numa_tonode_memory(void *start, size_t size, int node)
- Put memory on a specific set of nodes?
  - numa_tonodemask_memory(void *start, size_t size, struct bitmask *nodemask)
- Interleave memory across specific nodes
  - numa_interleave_memory(void *start, size_t size, struct bitmask *nodemask)

libnuma – Error Handling

- libnuma functions use return values to indicate errors
- Internal internal library warnings and errors
  - Display to the screen
  - Are not fatal, do not cause the program to exit
- Error and warning behavior can be changed
  - numa_exit_on_error
  - numa_exit_on_warn

For Further Study

- Whitepaper
  - Linux NUMA support for HP ProLiant servers
- Documentation
  - numa(8) man-page for a general overview of NUMA operation in Linux
  - numa(3) man-page for descriptions of libnuma API

LIKE US ON FACEBOOK

www.facebook.com/redhatinc

FOLLOW US ON TWITTER

www.twitter.com/redhatsummit

TWEET ABOUT IT

#redhat

READ THE BLOG

summitblog.redhat.com

GIVE US FEEDBACK

www.redhat.com/summit/survey