NUMA-aware scheduling

Red Hat, Inc.

Petr Holášek <pholasek@redhat.com>

7th Feb 2014

redhat.

Author

- Software Engineer in Kernel Generalists team at Red Hat
- z-stream kernel maintainer
- interested in the latest news in sched/MM development
- credits to Jirka Hladky for providing many benchmark outputs

redhat.

Section 1

Introduce to NUMA

redhat.

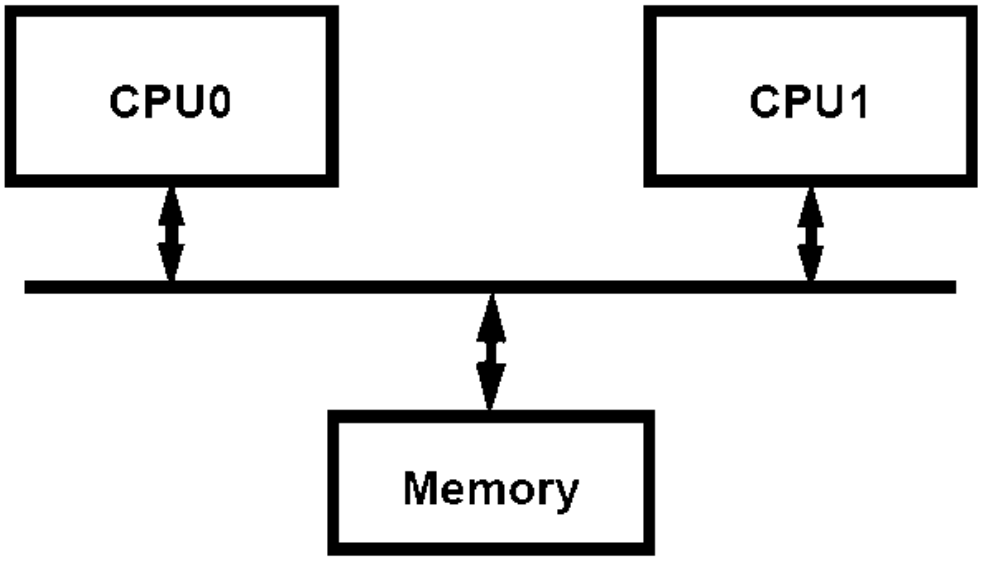
What is NUMA?

- non-uniform memory access
- supported by most of modern OS
- logical follower of UMA architecture using SMP
- heavily used in HPC world

redhat.

UMA

- uniform memory access
- typically based on SMP architecture with one bus
- memory bus is a bottleneck

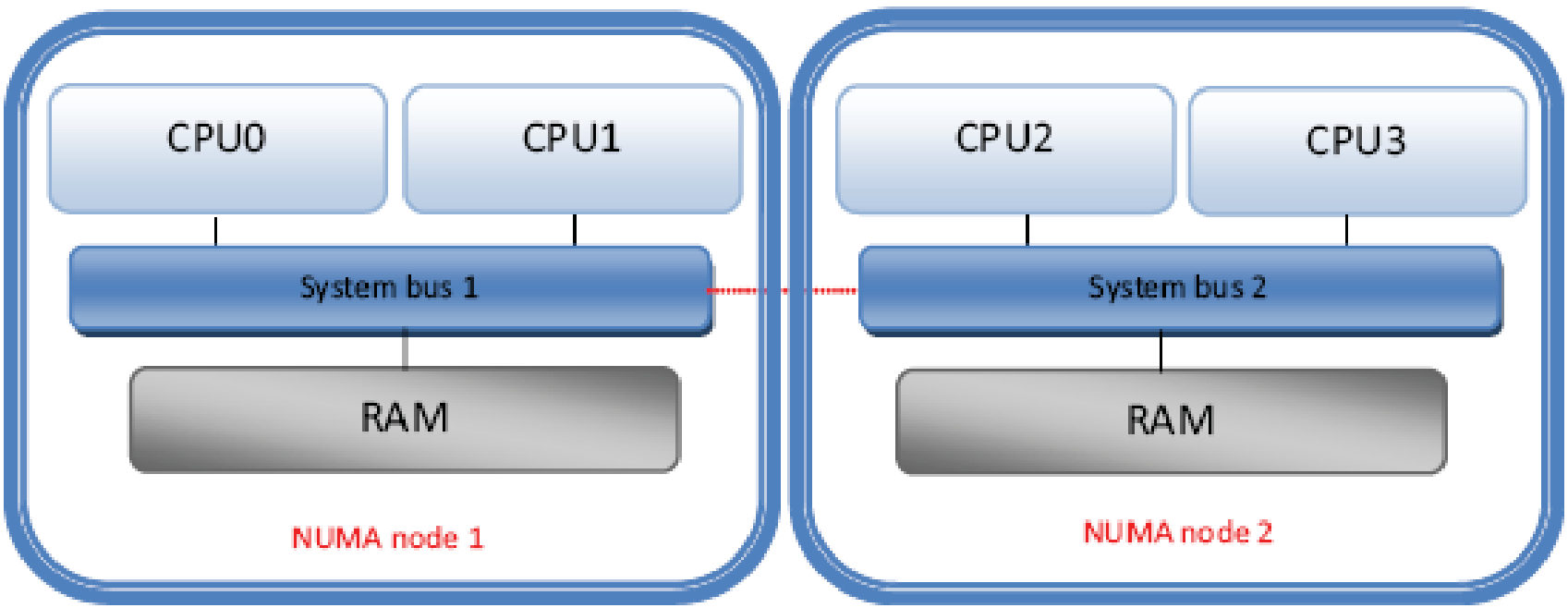


Source: http://ixbtlabs.com/articles2/cpu/rmma-numa.html

redhat.

NUMA

- CPUs are divided into groups
- NUMA reduces number of CPUs competing for shared memory bus
- better parallelism

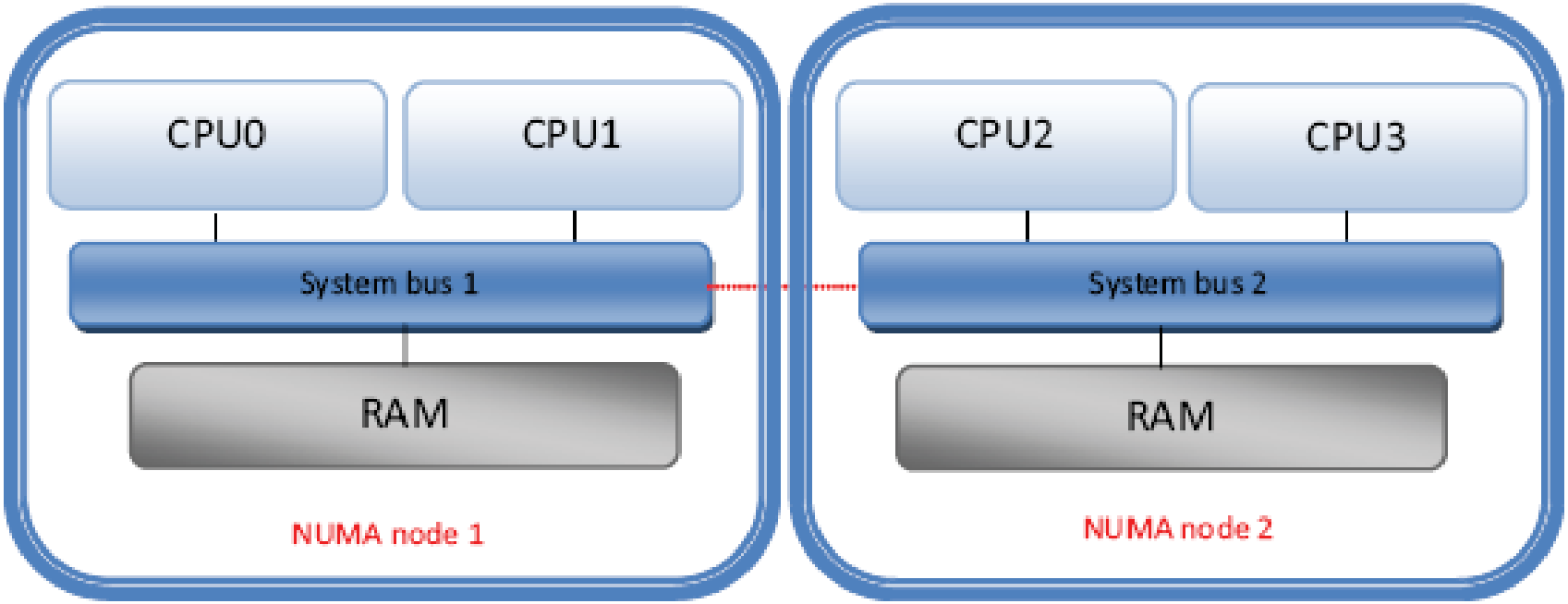


Source: http://www.networkcomputing.com/virtualization/harnessing-vsphere-performance-benefits/2295022

redhat.

Disadvantages of NUMA

- local memory access vs. remote memory access
- remote node access is ~2 times slower
- placement of tasks for balanced load



Source: http://www.networkcomputing.com/virtualization/harnessing-vsphere-performance-benefits/2295022

Section 2

Recent NUMA-aware scheduling changes

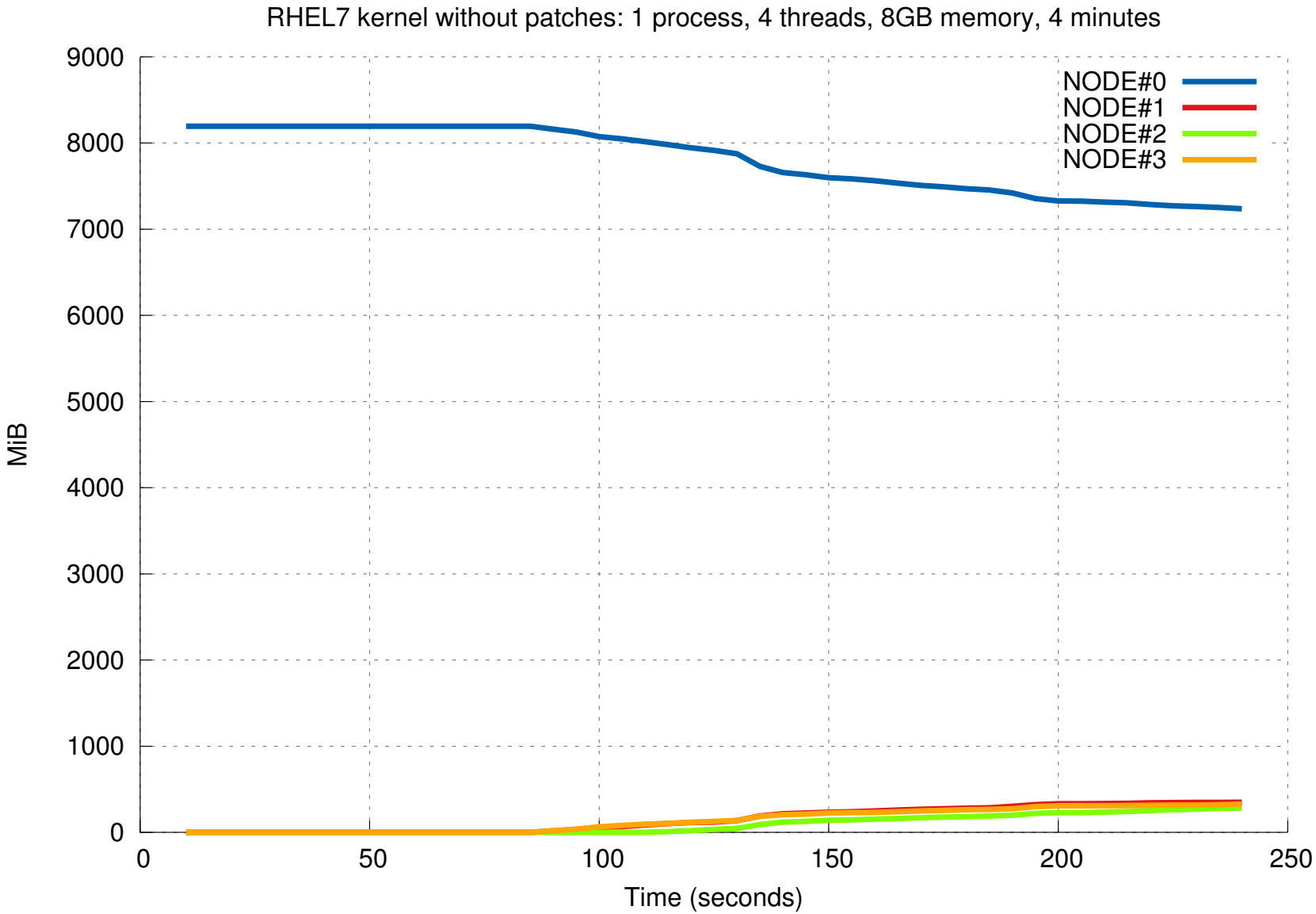
redhat.

Development

- first patchset merged in Oct 2013 - kernel 3.12
- based on work of many people during years 2012 and 2013
- still under active development
- this feature is fully automatic and requires only NUMA enabled in BIOS

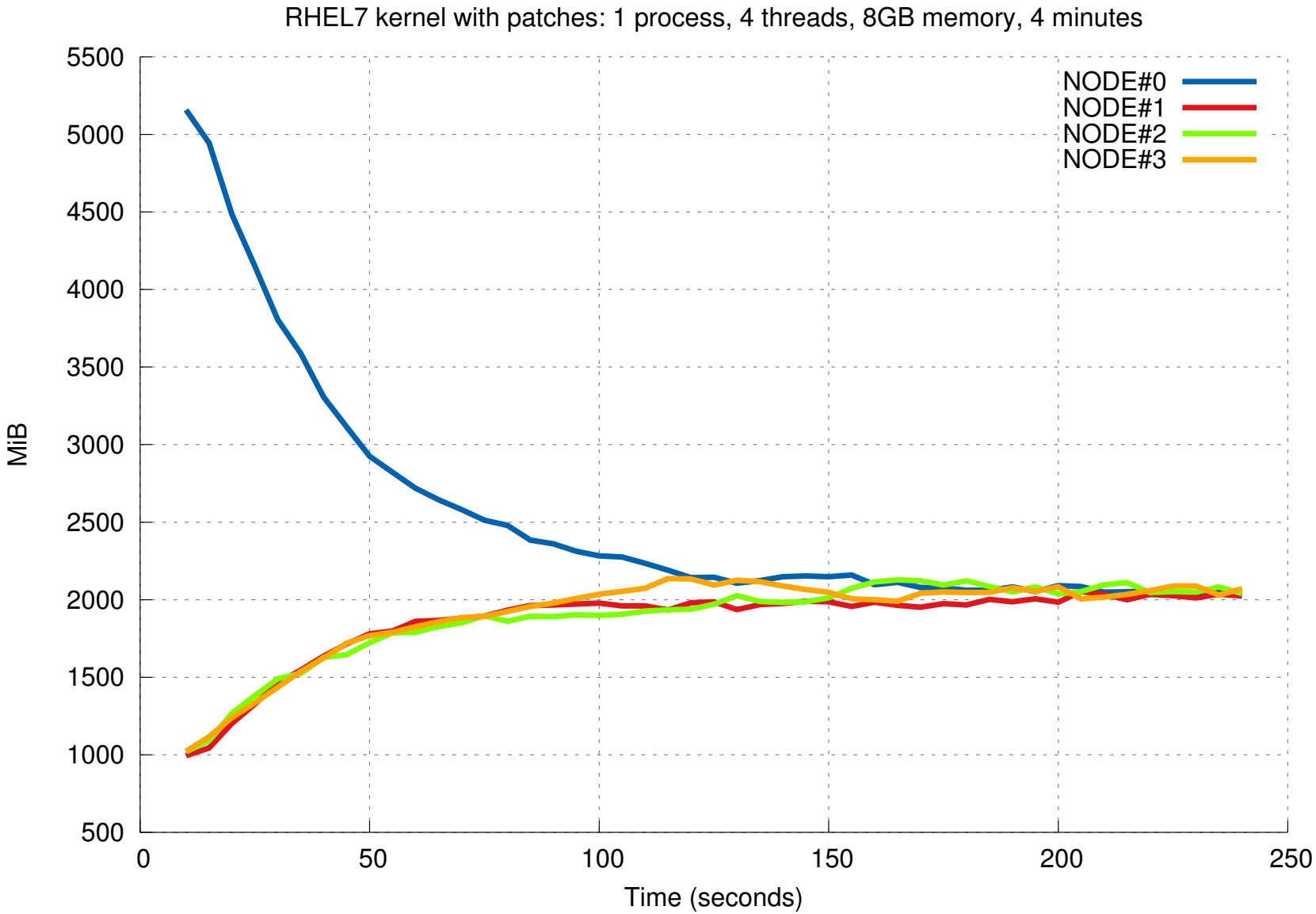
redhat.

How older kernels migrated tasks and memory



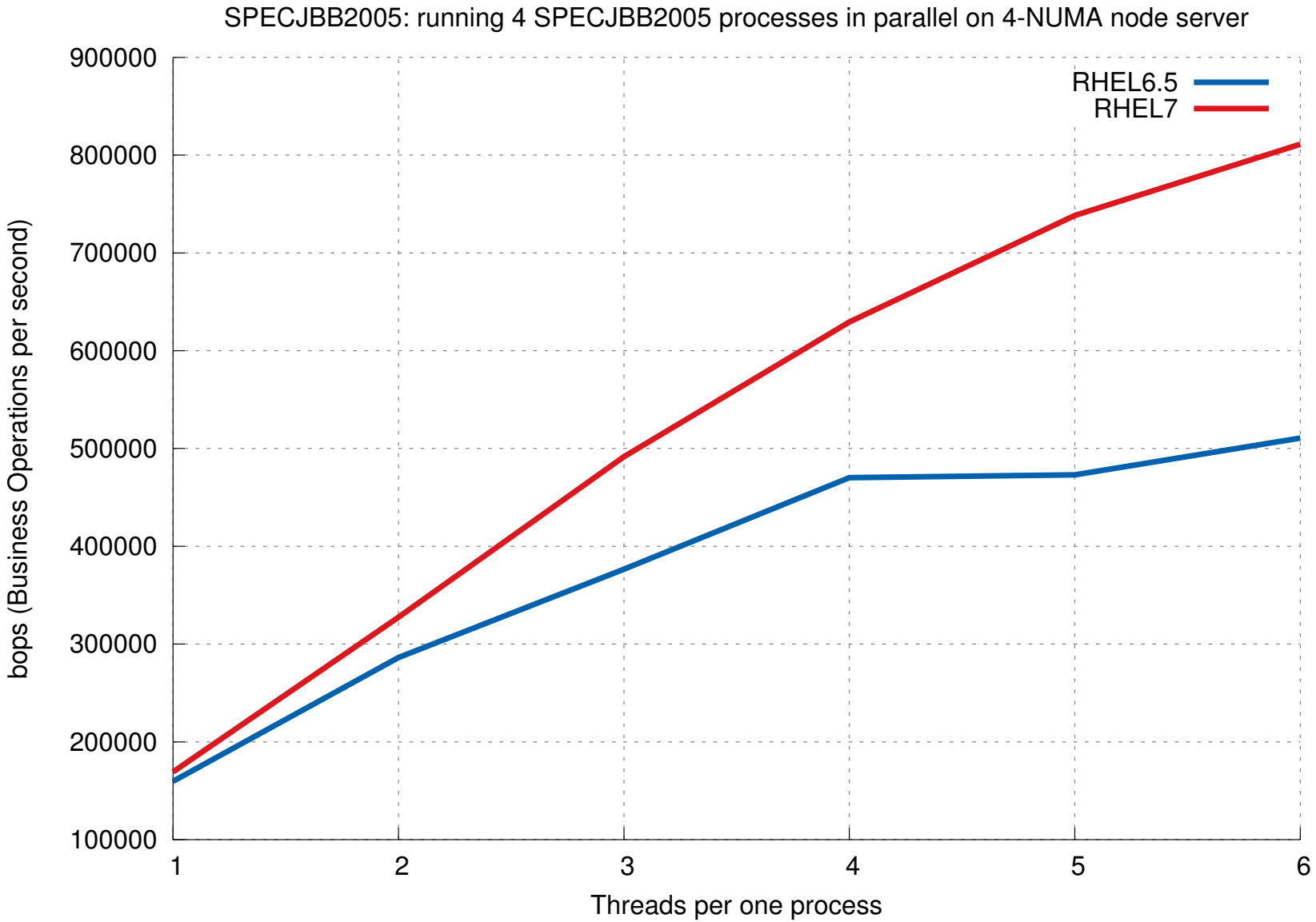
redhat.

How recent kernels migrate tasks and memory



redhat.

Results of SPECjbb2005



Section 3

Implementation

redhat.

Preferred node

- node with the most accessed memory
- every task has assigned preferred node
- kernel attempts to move task towards its preferred node

```
--- a/include/linux/sched.h
+++ b/include/linux/sched.h
@@ -1344,6 +1344,7 @@ struct task_struct {
struct callback_head numa_work;

+ unsigned long *numa_faults;
+ int numa_preferred_nid;

#endif /* CONFIG_NUMA_BALANCING */
```

redhat.

Tracking of page accesses

- kernel periodically unmaps memory of processes
- resulting pagefaults then provide access metadata
- not all types of pages are watched
- last `cpupid` encoded into `struct page` flags

redhat.

Migrating of memory

- memory is also migrated towards tasks
- kernel did it also before, but now it is better
- more conditions, e.g. tasks pinned to node

redhat.

NUMA groups

- groups of tasks accessing same shared pages
- shared page accesses are tracked at the group level
- shared pages accesses are preferred over private pages accesses

redhat.

How can I tune it?

- `cat /sys/kernel/debug/sched_features`
- `echo NO_NUMA > /sys/kernel/debug/sched_features`
- `echo NUMA > /sys/kernel/debug/sched_features`

Section 4

Summary

redhat.

Summary

- tasks and memory are migrated to each other
- performance gains upto ∼40% compared to RHEL6.5 on 4 nodes NUMA systems
- this feature **will be** included in RHEL7

redhat.

Sources

- https://lkml.org/lkml/2013/9/27/211
- http://lwn.net/Articles/568870/
- http://www.spec.org/jbb2005/
- http://kernel.org

The end.

Any questions?