Linux
Plumbers
Conference | Richmond, VA | Nov. 13-15, 2023

Using Hardware Hints for Optimal Page Placement

Bharata B Rao

Agenda

- Page access hints
- Hardware hints driven NUMA Balancing
- Experiments and results
- Opportunities and possibilities

Page access hints

- Page access information is useful for
  - moving page closer to where it is accessed
  - discarding/demoting unused/cold pages
  - promoting hot pages to faster memory
- In-kernel users of page access information
  - NUMA Balancing
  - Hot page promotion
  - Access-bit based reclaim/LRU lists
  - Idle page tracking
  - DAMON monitoring
- User space assistance in moving pages
  - migrate_pages(), move_pages()

Hardware-provided access hints

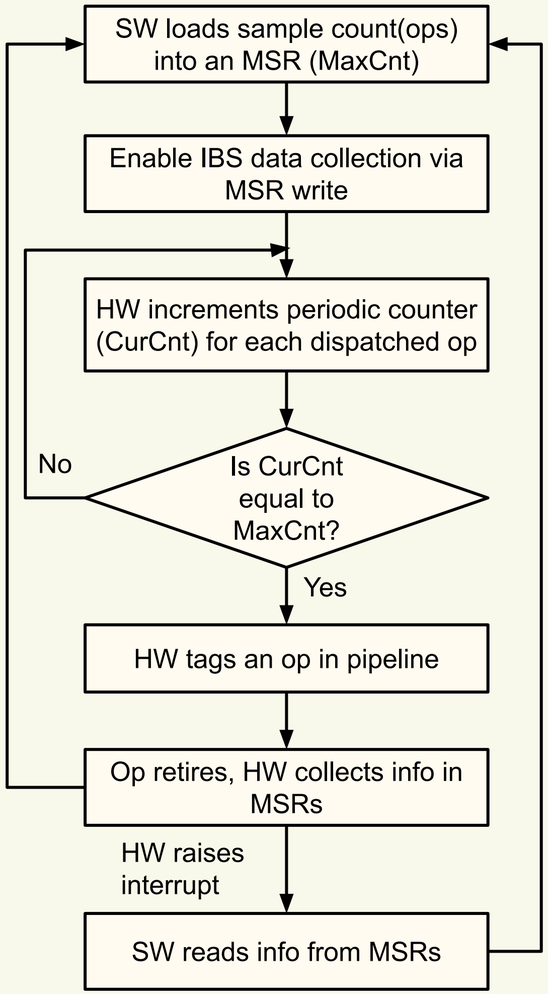
- Newer platforms are providing memory access info that OS can use
  - Target address of memory load/store
  - Data source information
  - Cache information
- Effectiveness of hardware hints matters
  - Actionable information
  - Ease of consumption
  - Low overhead
- Examples of platform provided hinting mechanisms
  - Instruction Based Sampling (IBS) in AMD
  - Processor Event Based Sampling (PEBS) in Intel
  - Hot-Cold Affinity (HCA) in IBM Power

Instruction Based Sampling - IBS

- Hardware facility in AMD processors to gather metrics related to instruction fetch and execution
  - Separate and independent Fetch(front end) and Op (back end/execution) sampling
  - Independent of PMU
  - One instance per hardware thread
  - No overhead during fetch or execution, only overhead in reporting
  - Hardware collects data by tagging a selected micro-op
  - Programmable sampling interval (ops or clock cycles)
  - When tagged op is retired, execution info collected in MSRs is reported via an interrupt
- More info - Sec 13.3 of APM vol2, Sec 2.1.15.4 of PPR vol1
- Has been traditionally used for performance profiling
  - Linux perf supports IBS based precise event profiling
- *Execution sampling can be used for memory access profiling too*

Memory access profiling using IBS

IBS workflow



Sample filtering

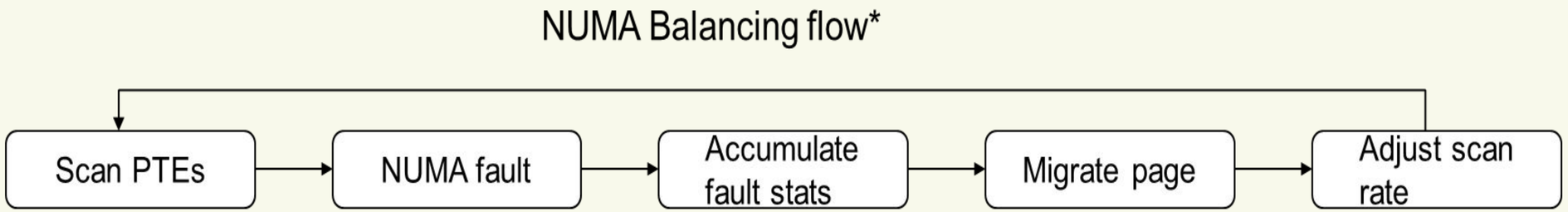
- Load/store op filtering in software
- L3 Miss filtering by hardware

Data provided for each load/store sample

- Precise Instruction Pointer
- Linear and physical address of data operand
- Data source information for Cache, DRAM, Extension memory (CXL), MMIO, IO accesses (valid for load)
- Remote node indication
- Cache miss info
- DTLB hit/miss info

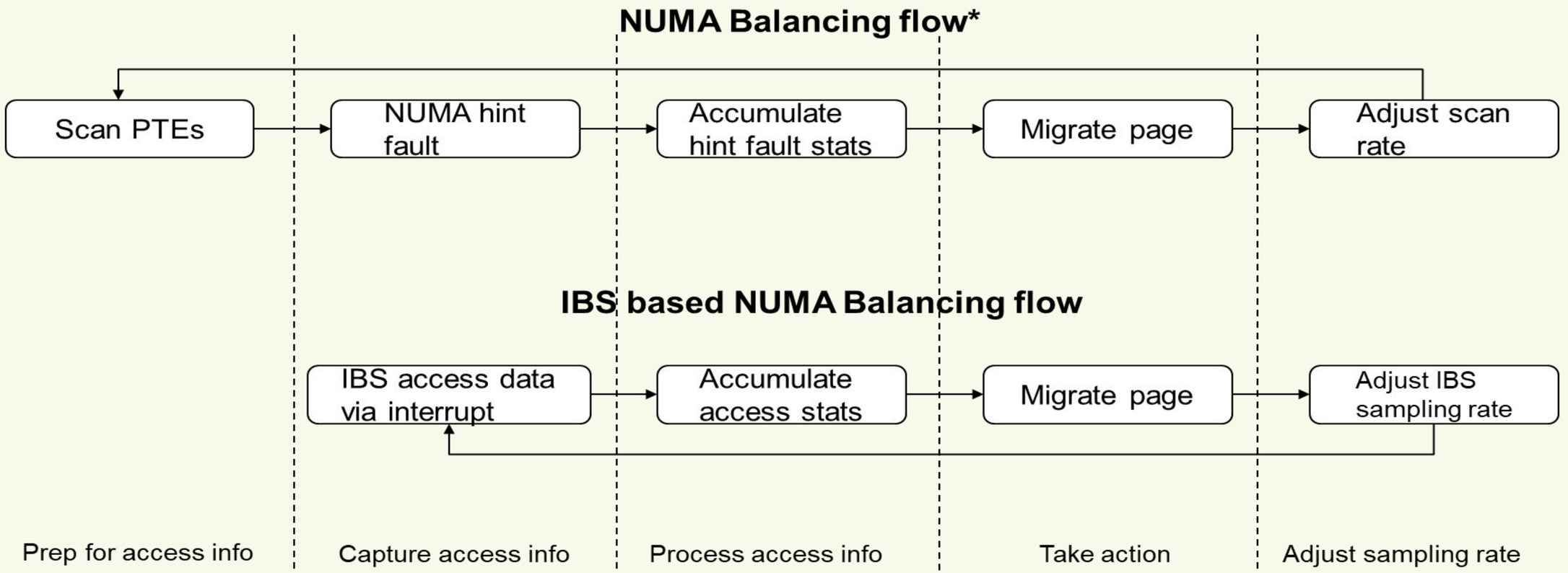
NUMA Balancing

- Compute follows memory and/or memory follows compute
- Scanning process address space periodically to introduce NUMA hint fault
  - Marking PTE as PROT_NONE
- Page access info is obtained from NUMA hint fault
  - Local, remote, private and shared faults
- Pages and tasks are migrated to appropriate nodes
- Scan rate is adjusted based on how the accesses converge
- Also drives hot page promotion in tiered memory systems



- Simplified view

IBS driven NUMA Balancing



- Simplified view

Implementation

- Per-task access profiling
  - Enabled with NUMA_BALANCING_NORMAL and NUMA_BALANCING_MEMORY_TIERING modes
  - Per-task sampling interval that varies based on convergence (similar to numa_scan_period)
- Profiling is disabled on kernel entry and enabled on kernel exit
  - Programming IBS on/off at kernel exit/entry points through an MSR write
- Task work is queued on each IBS interrupt (NMI)
  - Relevant IBS MSRs are read to get information about memory access
  - Non-actionable samples are discarded
- Feed access info to NUMA balancing as “faults”
  - NUMA Balancing is driven from task work context
- Hot page promotion
  - Time difference between successive accesses to the page is treated as hint fault latency equivalent
- RFC patchset

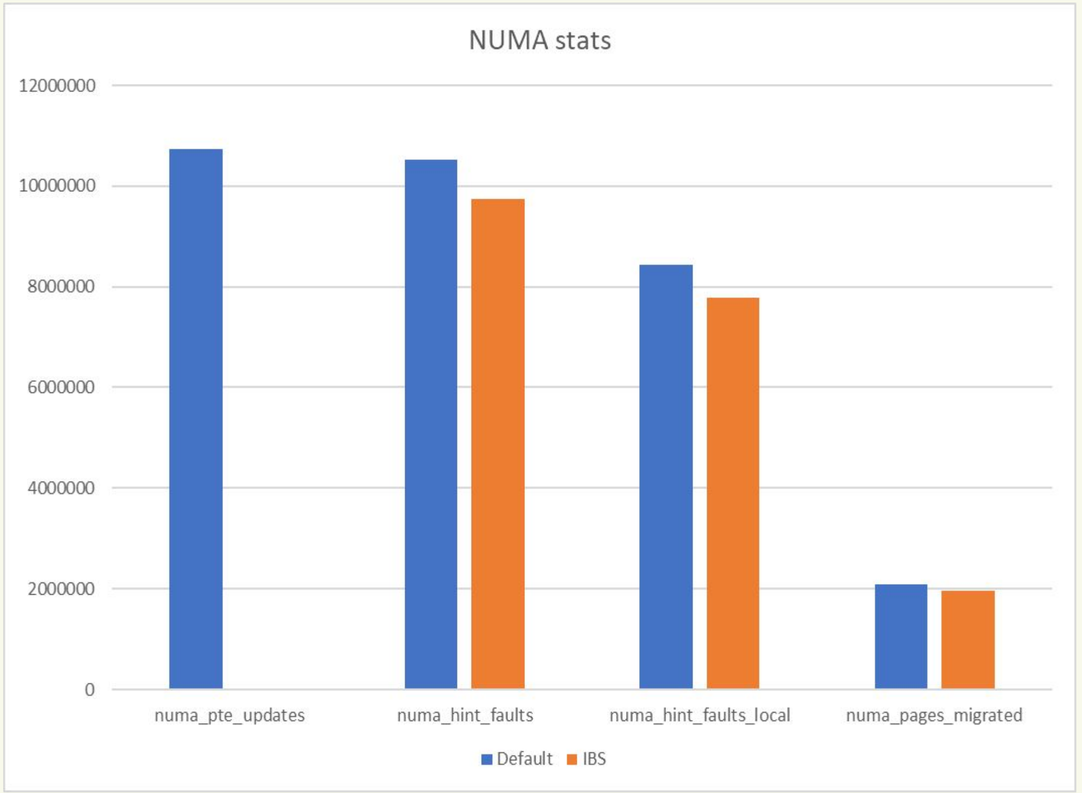
Test setup

- Test system
  - 2P 3-node AMD Zen4 system with 2 regular nodes and 1 CXL node
  - 256 CPUs on each regular node
  - 128GB memory on each node
- Micro-benchmark
  - Threaded application that provisions memory(8G) on remote node (Node 1 or 2)
  - Multiple threads (64 on Node 0) concurrently write to random bytes in the allocated range
    - 1073741824 accesses per thread
    - 100us delay after 1000 accesses
  - Only 4K pages, no THP
  - Benchmark score is the time taken to complete the fixed number of accesses
    - NUMA Balancing is expected to migrate the remote pages to local node
- Evaluation criteria
  - Can hardware hints based NUMA Balancing match the default method?
  - Is the overhead tolerable?

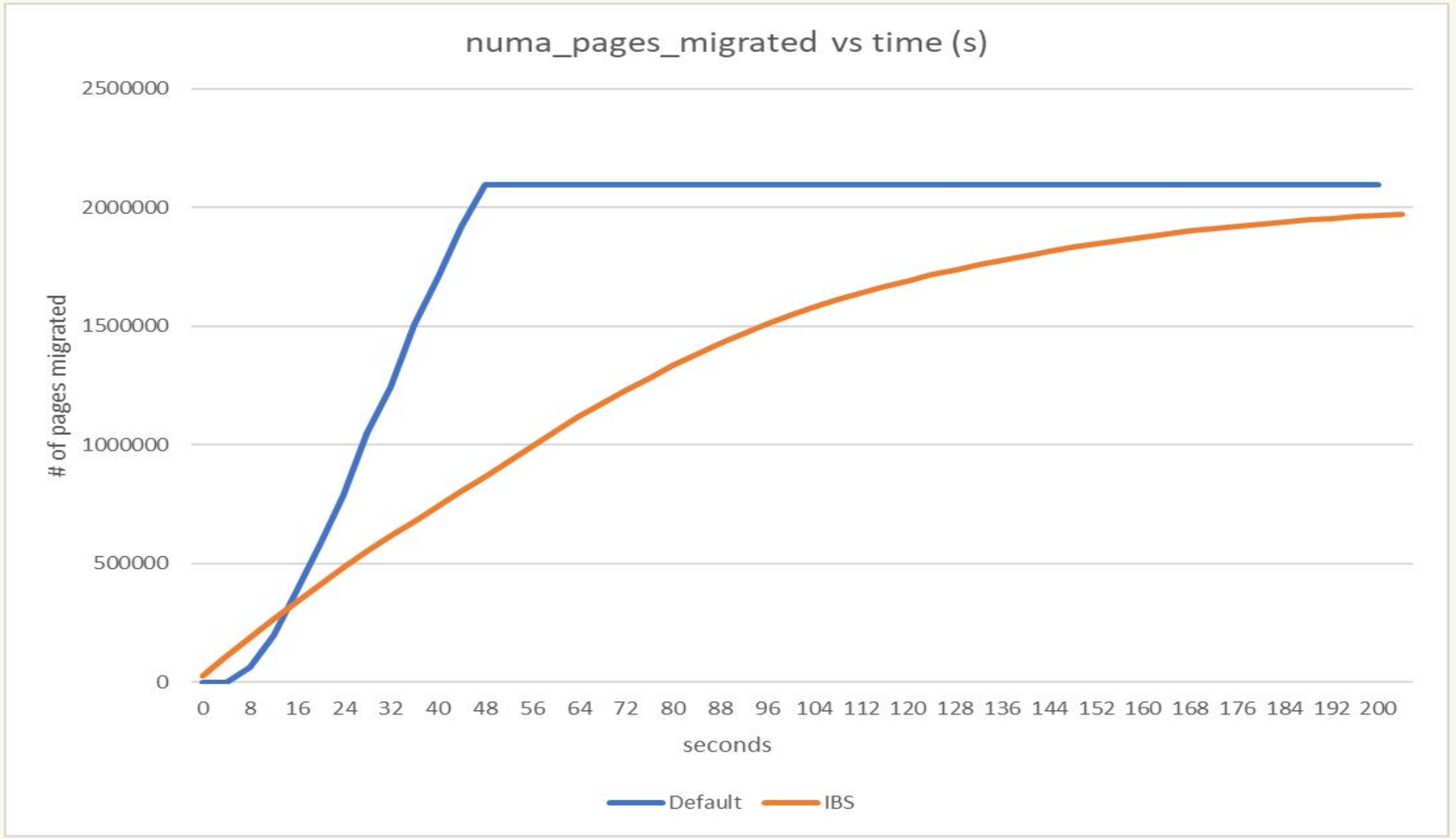
```
node distances:
node   0   1   2
  0:  10  32  60
  1:  32  10  50
  2: 255 255 10
```

Node1 Regular, numa_balancing=1

	Default	IBS
Benchmark score (s) (Lower is better)	**203.4**	**205.7**
numa_pte_updates	10742345	0
numa_hint_faults	10742345	9753187
numa_hint_faults_local	8429232	7782259
numa_pages_migrated	2097278	1970928
ibs_nr_events		9821313
ibs_useful_samples		9753199

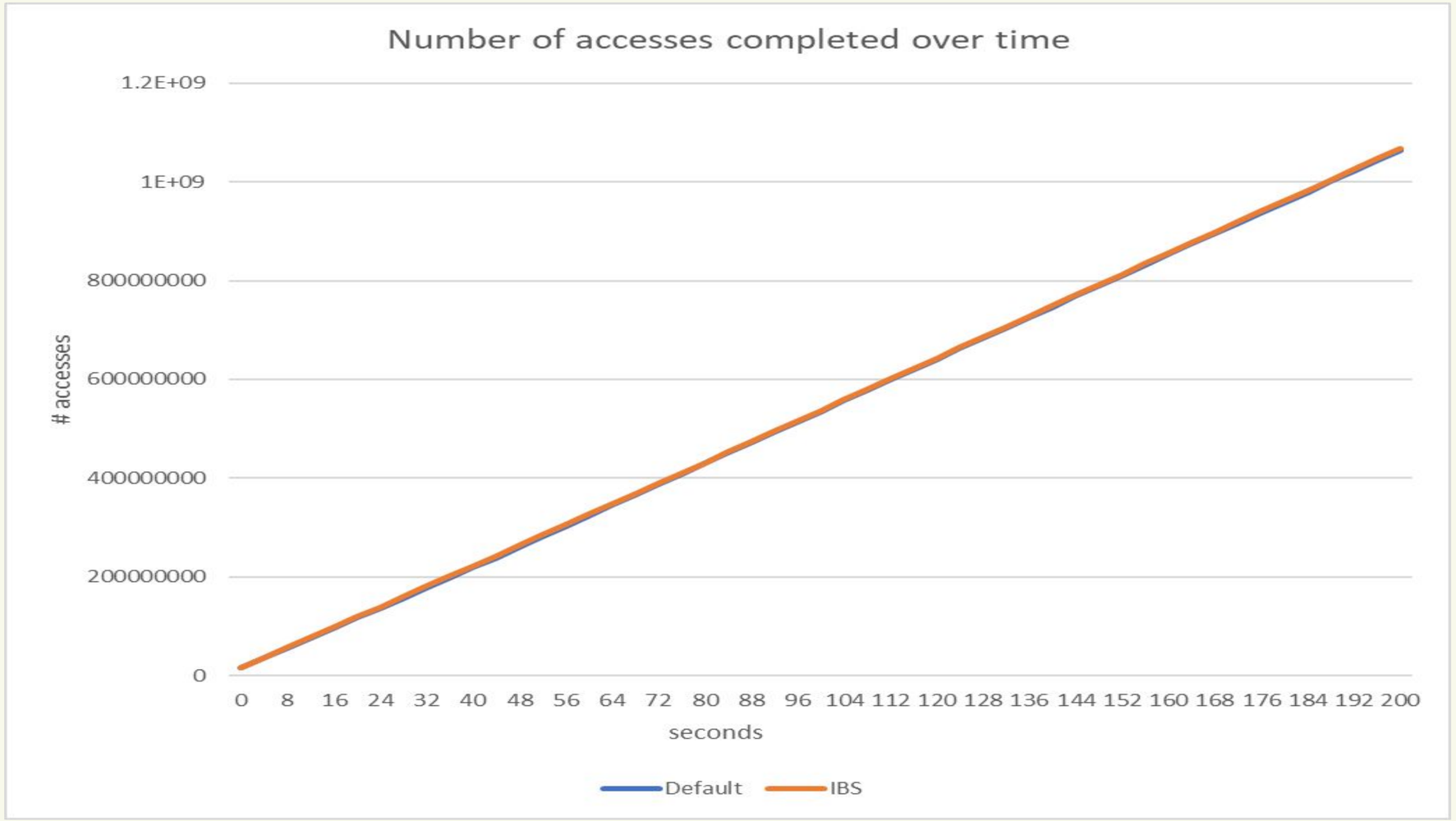


Node1 Regular, numa_balancing=1



- Default case detects and migrates all remote pages quite early
- IBS remote access samples get reported more gradually

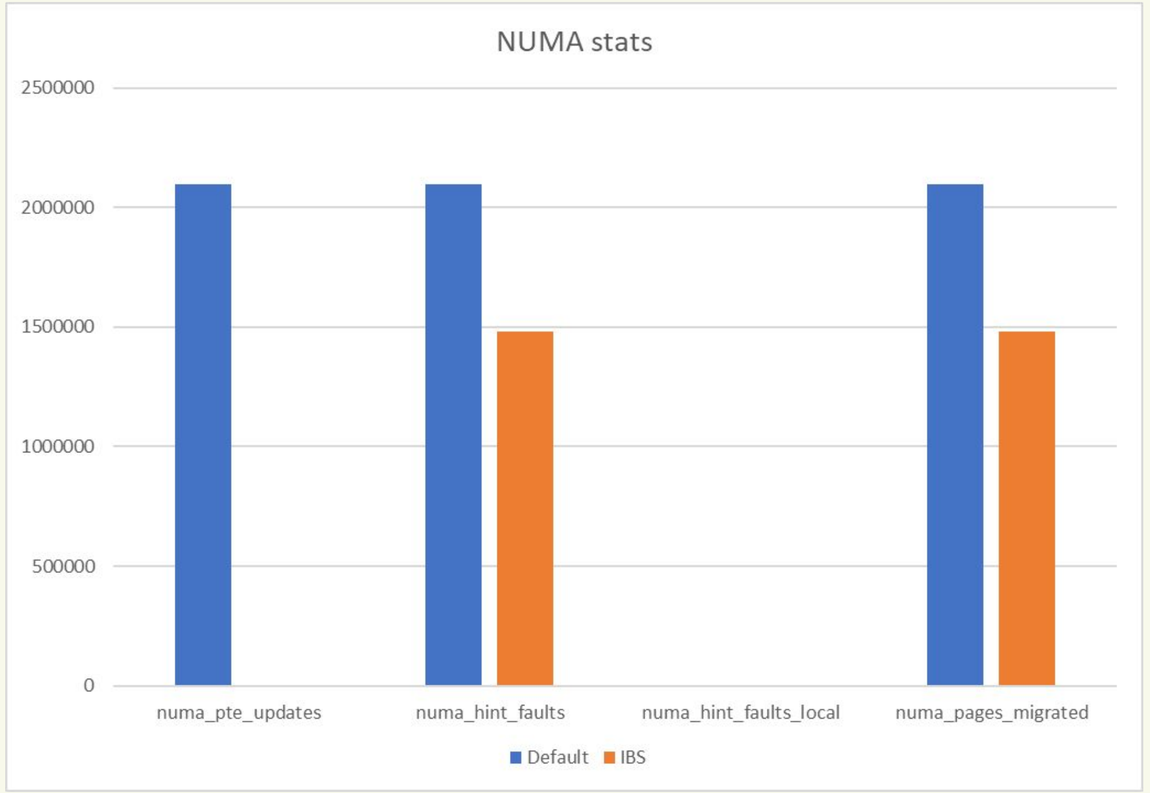
Node1 Regular, numa_balancing=1



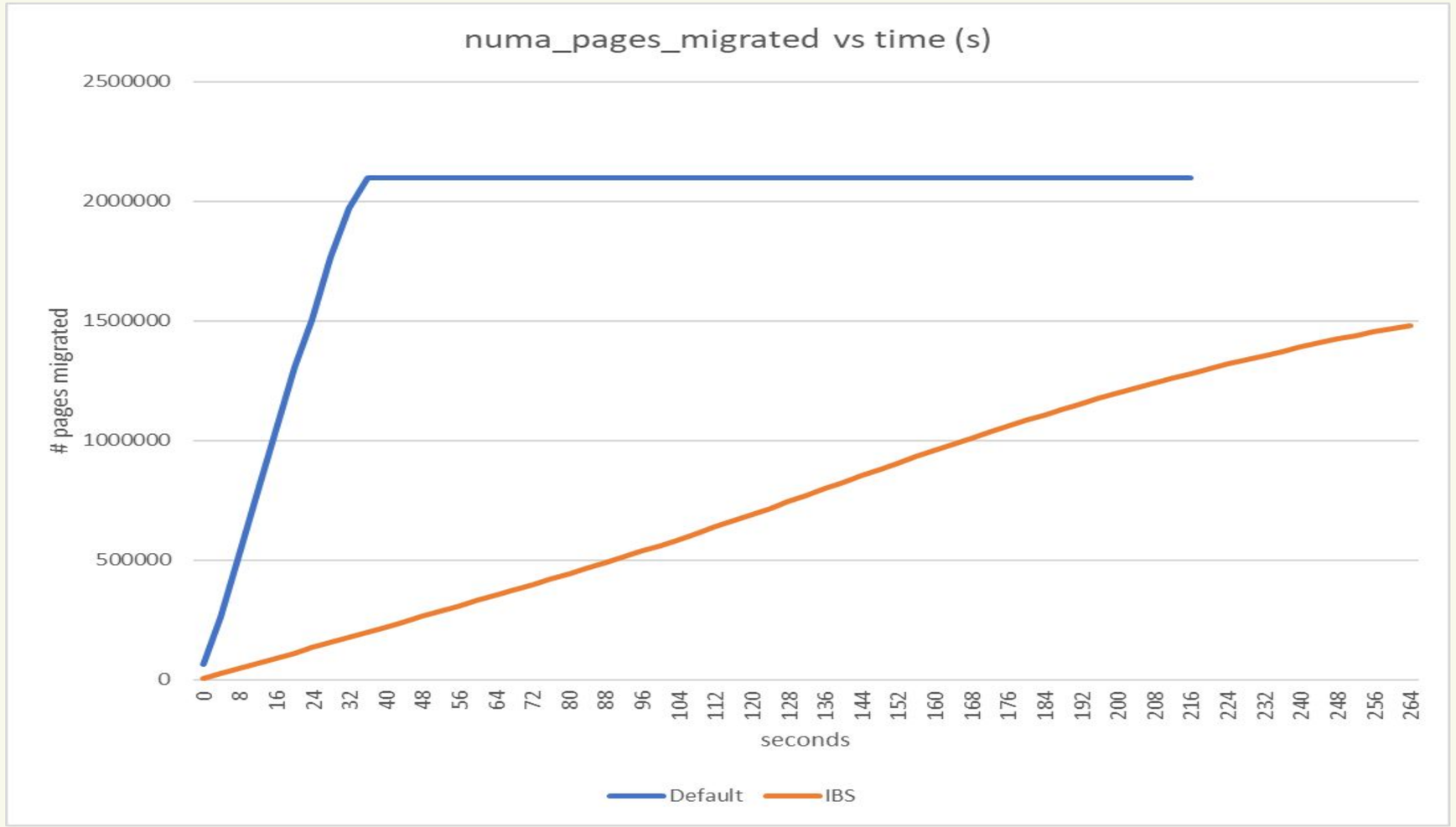
- Benchmark progresses similarly in both the cases

Node2 CXL, numa_balancing=2

	Default	IBS
Benchmark score (s) (Lower is better)	**216.9**	**266.7**
numa_pte_updates	2097153	0
numa_hint_faults	2098401	1480993
numa_hint_faults_local	0	0
numa_pages_migrated	2097153	1480993
ibs_nr_events		5122716
ibs_useful_samples		1480994

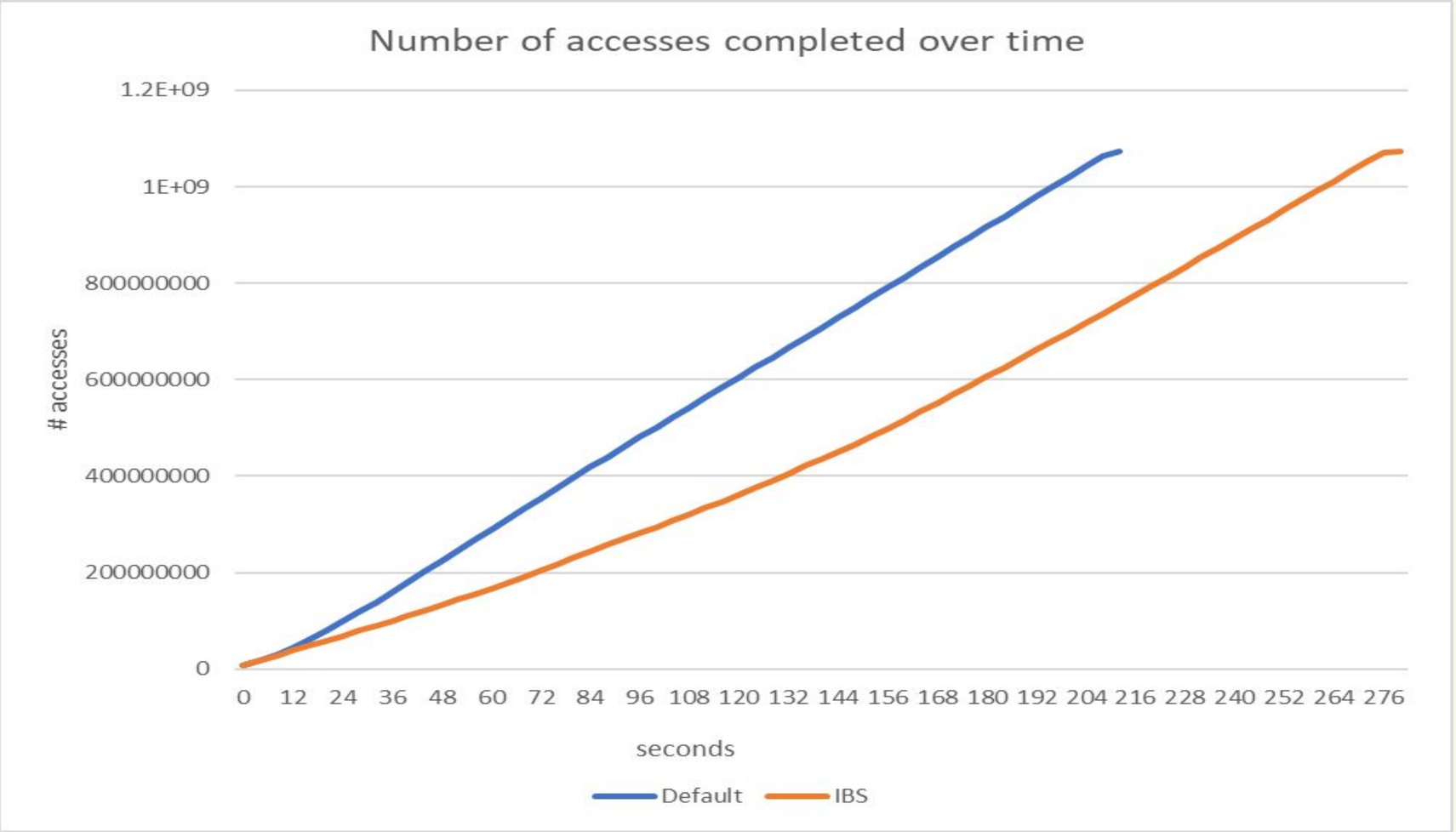


Node2 CXL, numa_balancing=2



- Default case detects and migrates all remote pages quite early
- IBS remote access samples get reported more gradually
- Number of remote samples that IBS reports is lower than desirable resulting in less migration

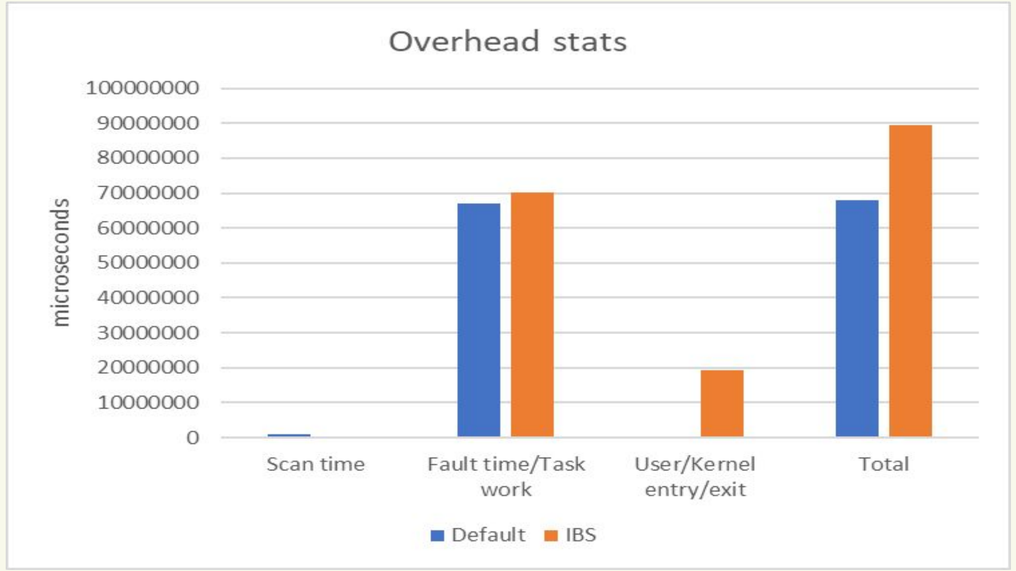
Node2 CXL, numa_balancing=2

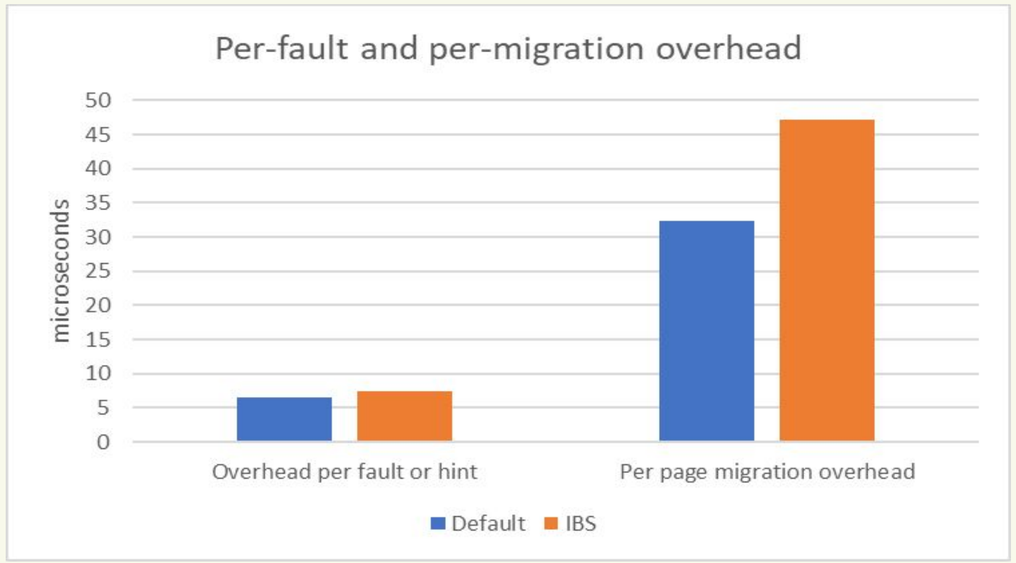


- Slower and gradual migration of remote CXL pages slows down the IBS case
- CXL latency matters

IBS hinting overhead data(us) for an example run

	Default	**IBS**
Scan time	986000	0
Fault or task work time	66868000	70142000
Kernel entry/exit IBS enable/disable time	0	19246000
Total (us)	67854000	89388000
Nr faults or hints	10528130	12155554
numa_pages_migrated	2097278	1891114
Overhead per fault or access	6.44	7.35
Per-page migration overhead	32.35	47.26

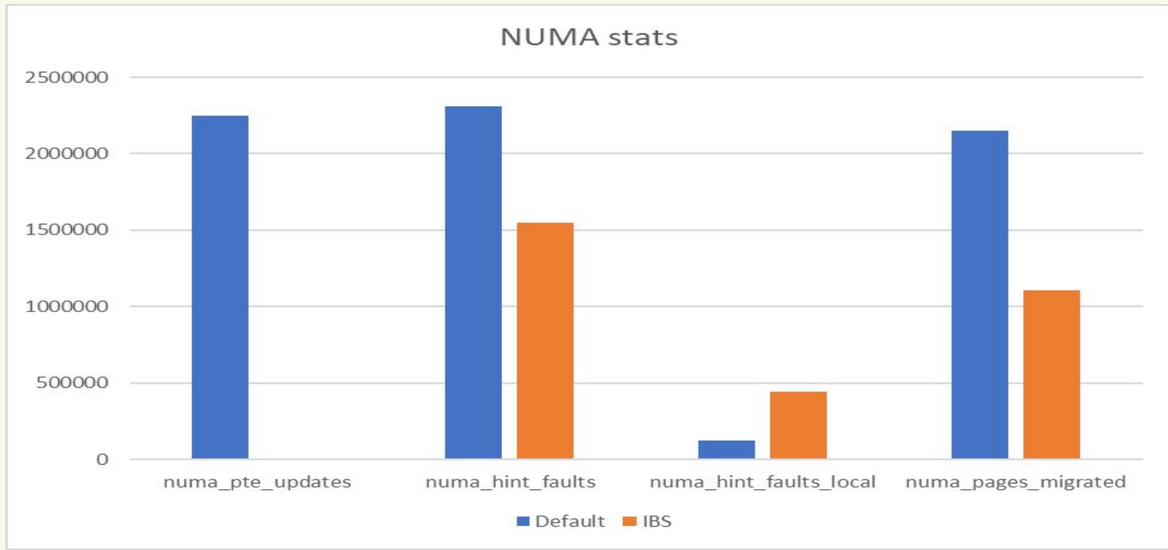




Virtualization

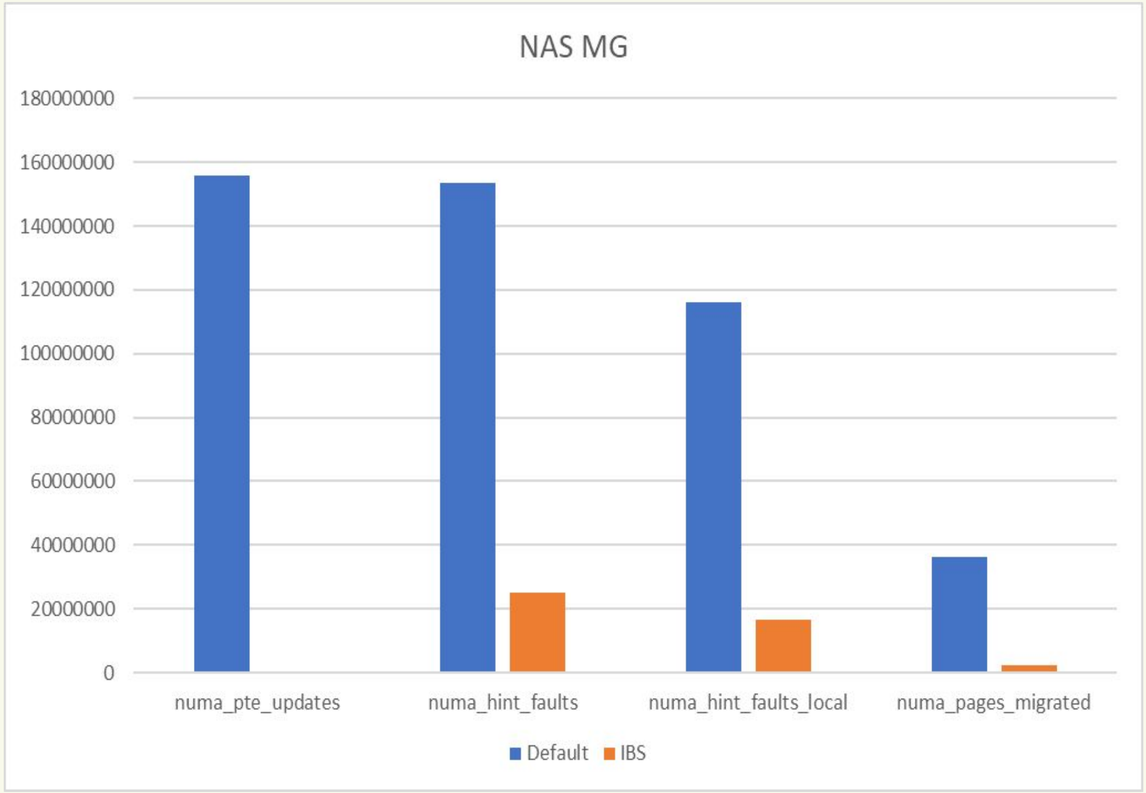
- IBS provides guest virtual and host physical addresses for accesses originating from guest
  - Guest pages in the host can be balanced transparently to the guest
- Experiment
  - KVM guest with its vCPUs pinned to host Node 0
  - Entire guest RAM is manually moved to Node 1
  - Memory accessing Micro-benchmark is run inside the guest
  - NUMA Balancing in the host will detect the remote node accesses and move the pages to local node

	Default	**IBS**
Benchmark score (s) (Lower is better)	**296**	**242**
numa_pte_updates	2251064	0
numa_hint_faults	2309422	1547451
numa_hint_faults_local	124829	444026
numa_pages_migrated	2150435	1103425
ibs_nr_events		2746003
ibs_useful_samples		2585031



NAS MG

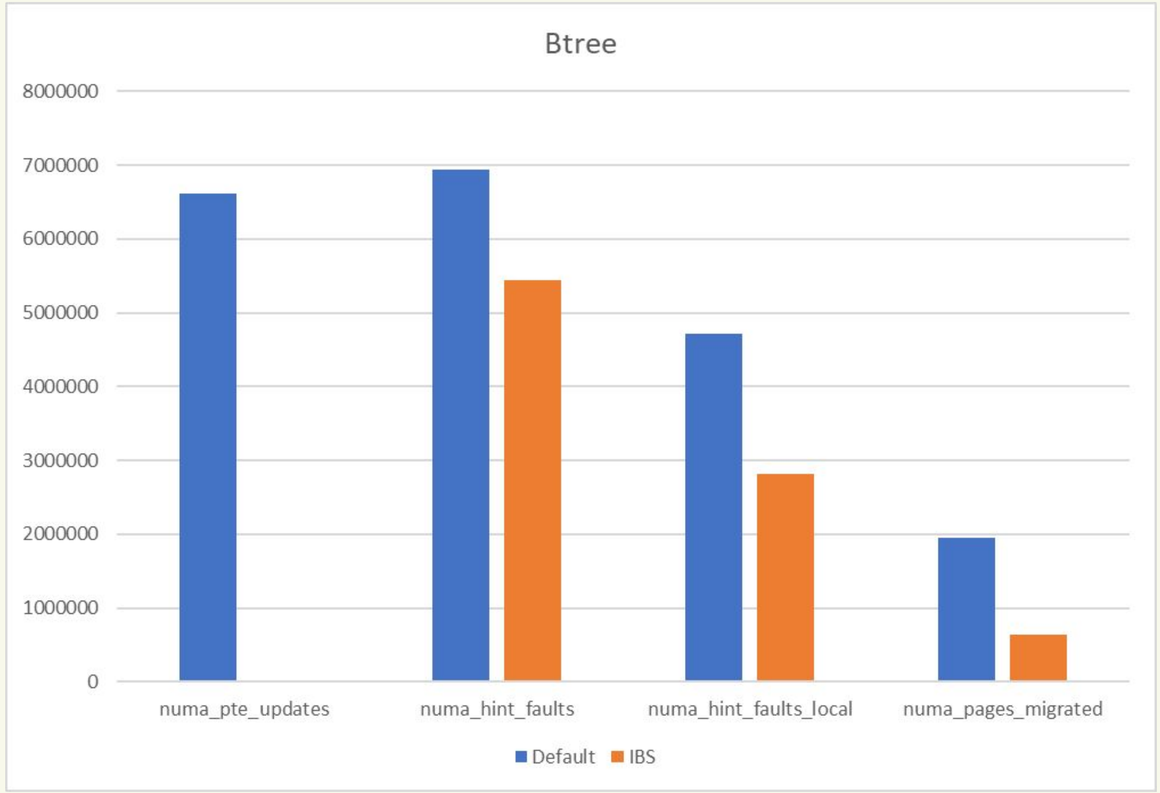
	Default	IBS
Throughput (Mop/s) (Higher is better)	**64692.17**	**67092.95**
numa_pte_updates	155653446	0
numa_hint_faults	153607660	25083212
numa_hint_faults_local	116226917	16717155
numa_pages_migrated	36078044	2185338
ibs_nr_events		25140283
ibs_useful_samples		25083376



Resources: 512 threads, 210G

Btree

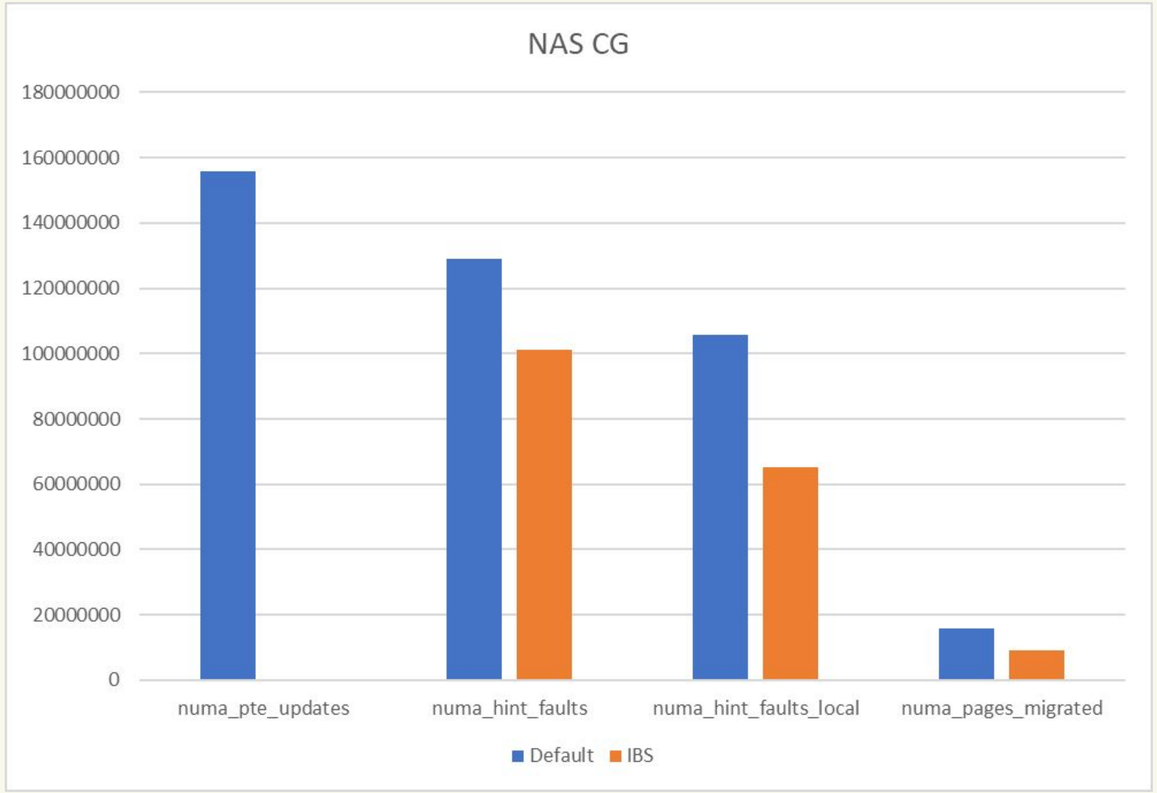
	Default	IBS
Time taken (s) (Lower is better)	**356**	**345**
numa_pte_updates	6619519	0
numa_hint_faults	6931171	5437892
numa_hint_faults_local	4718461	2818380
numa_pages_migrated	1952959	646025
ibs_nr_events		5469507
ibs_useful_samples		5438296



Resources: 512 threads, 150G

NAS CG

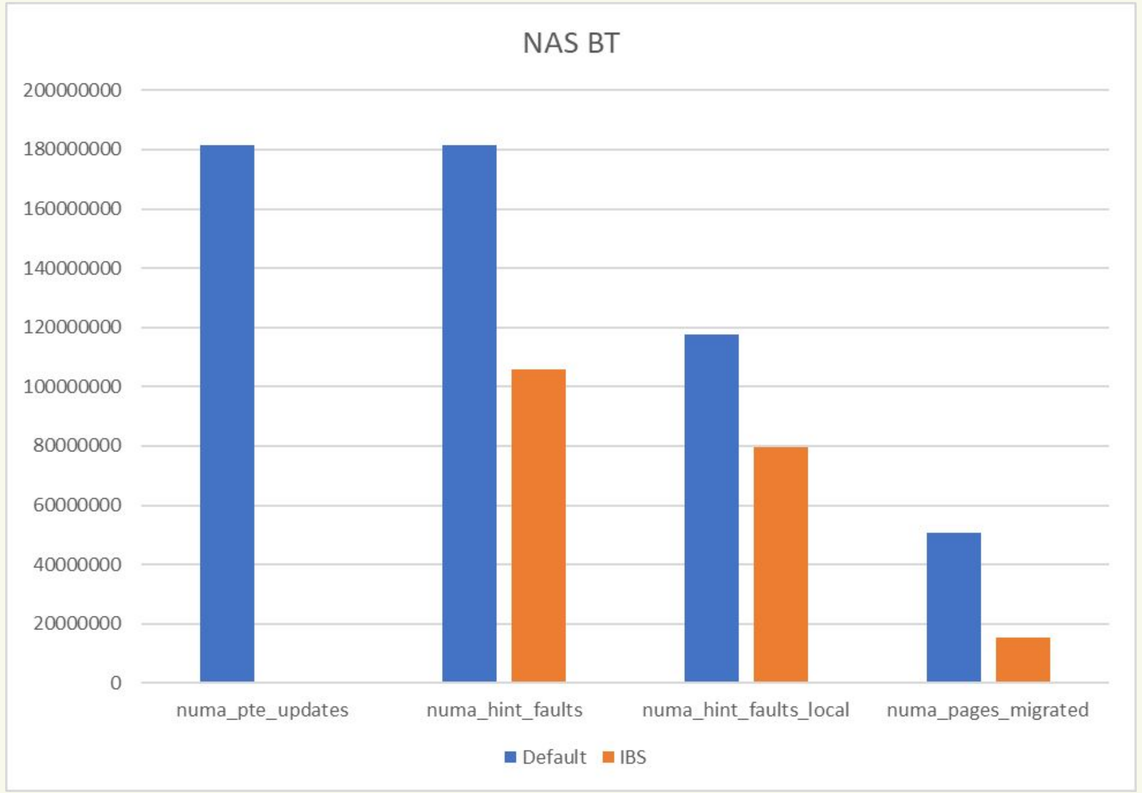
	Default	IBS
Throughput (Mop/s) (Higher is better)	**5521.96**	**5308.83**
numa_pte_updates	155763925	0
numa_hint_faults	129089135	101027118
numa_hint_faults_local	105795831	65278913
numa_pages_migrated	15787876	9209945
ibs_nr_events		101239466
ibs_useful_samples		101091808



Resources: 512 threads, 200G

NAS BT

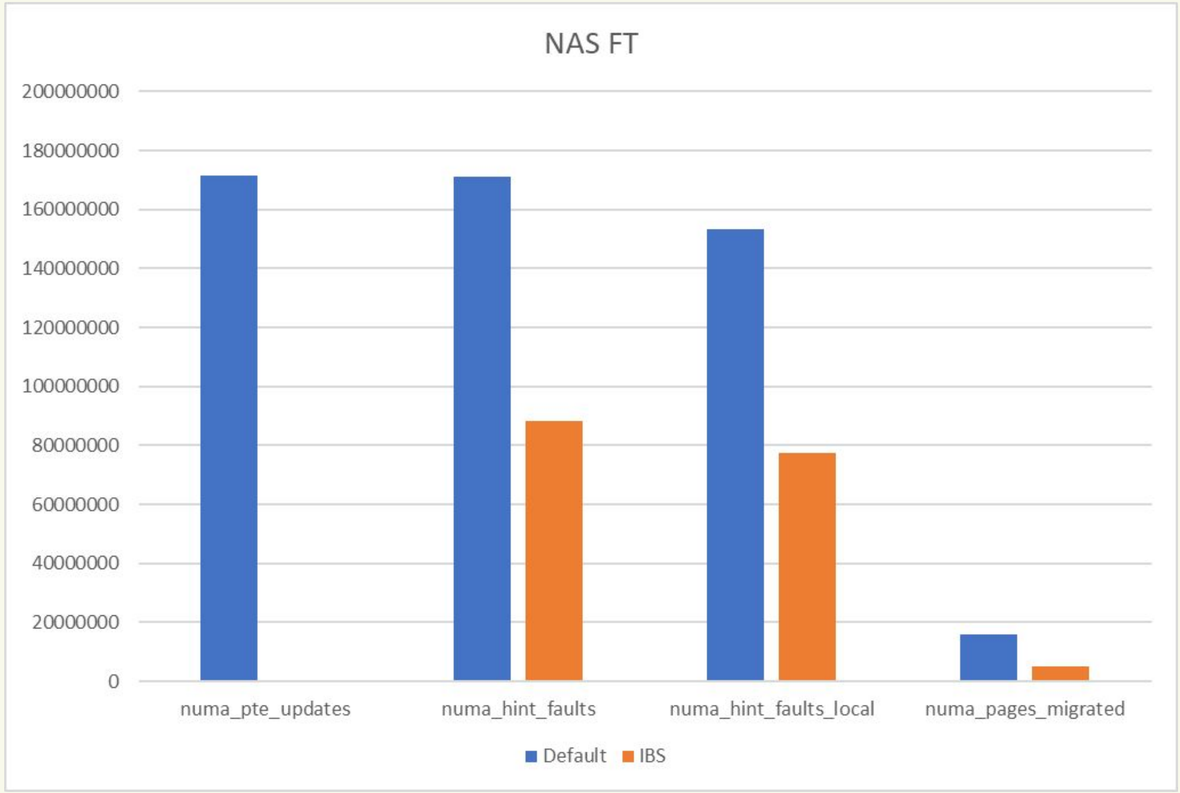
	Default	IBS
Throughput (Mop/s) (Higher is better)	**162448.18**	**161023.92**
numa_pte_updates	181316500	0
numa_hint_faults	181338494	105713521
numa_hint_faults_local	117733504	79722227
numa_pages_migrated	50556489	15331160
ibs_nr_events		106100376
ibs_useful_samples		105713711



Resources: 512 threads, 160G

NAS FT

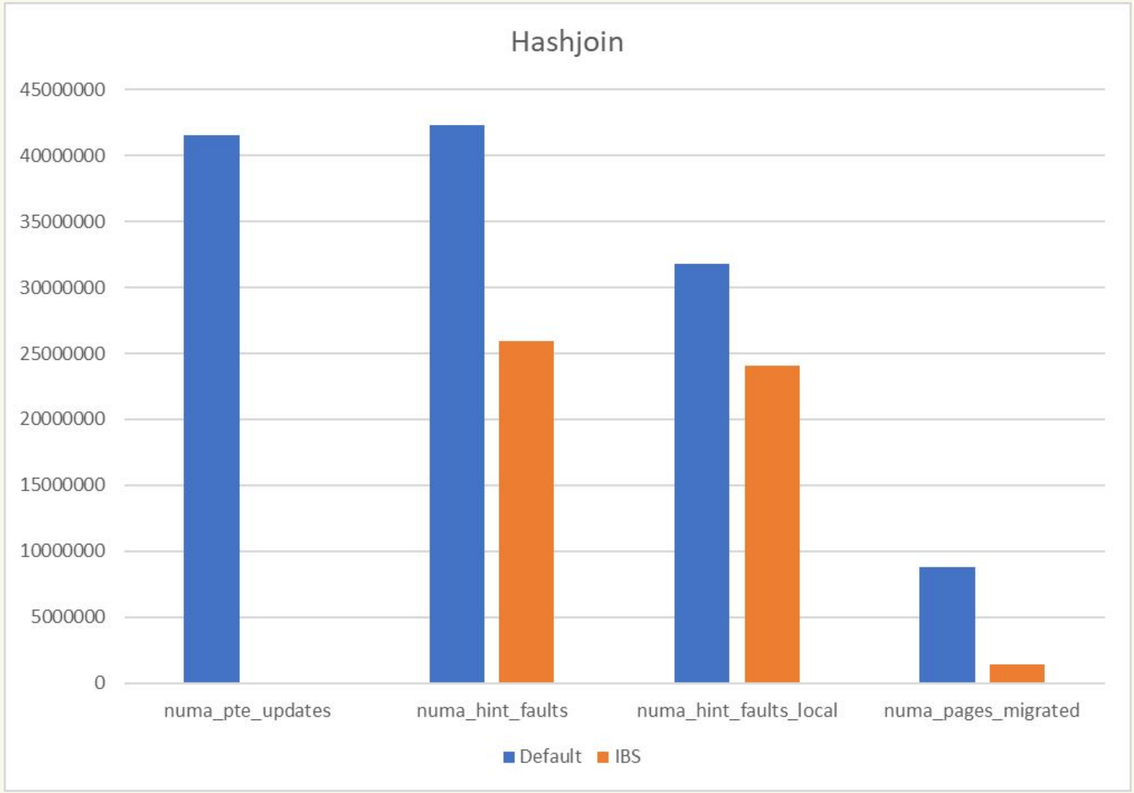
	Default	IBS
Throughput (Mop/s) (Higher is better)	**85178.26**	**82300.77**
numa_pte_updates	171613512	0
numa_hint_faults	171219386	88078256
numa_hint_faults_local	153103000	77589776
numa_pages_migrated	15784076	4982272
ibs_nr_events		88331818
ibs_useful_samples		88081626



Resources: 512 threads, 80G

Hashjoin

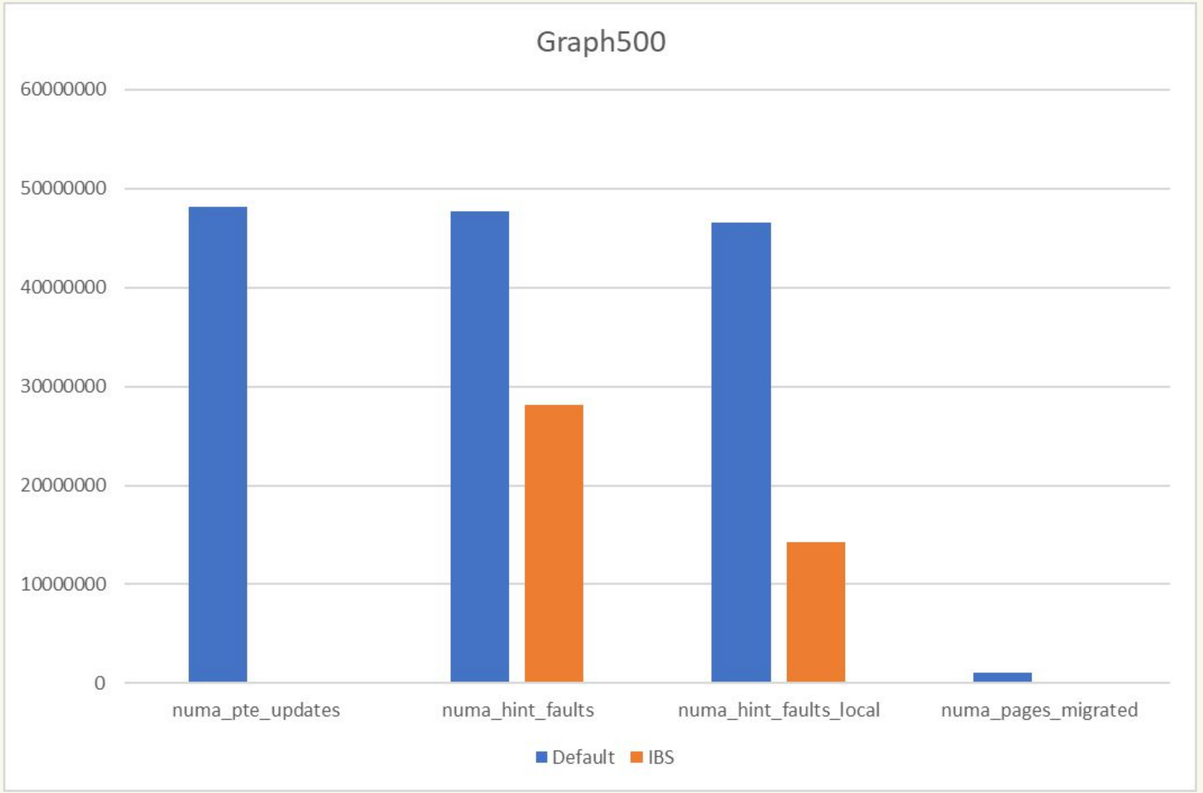
	Default	IBS
Time taken (s) (Lower is better)	**342**	**378**
numa_pte_updates	41549803	0
numa_hint_faults	42337102	25940529
numa_hint_faults_local	31741454	24052924
numa_pages_migrated	8786556	1436730
ibs_nr_events		26142158
ibs_useful_samples		25942835



Resources: 512 threads, 80G

Graph500

	Default	IBS
harmonic_mean (TEPS) (Higher is better)	**5795293274.03165**	**5707201722.19172**
numa_pte_updates	48172395	0
numa_hint_faults	47737148	28151422
numa_hint_faults_local	46530639	14240376
numa_pages_migrated	1022593	120058
ibs_nr_events		28208376
ibs_useful_samples		28167569



Resources: 512 threads, 70G

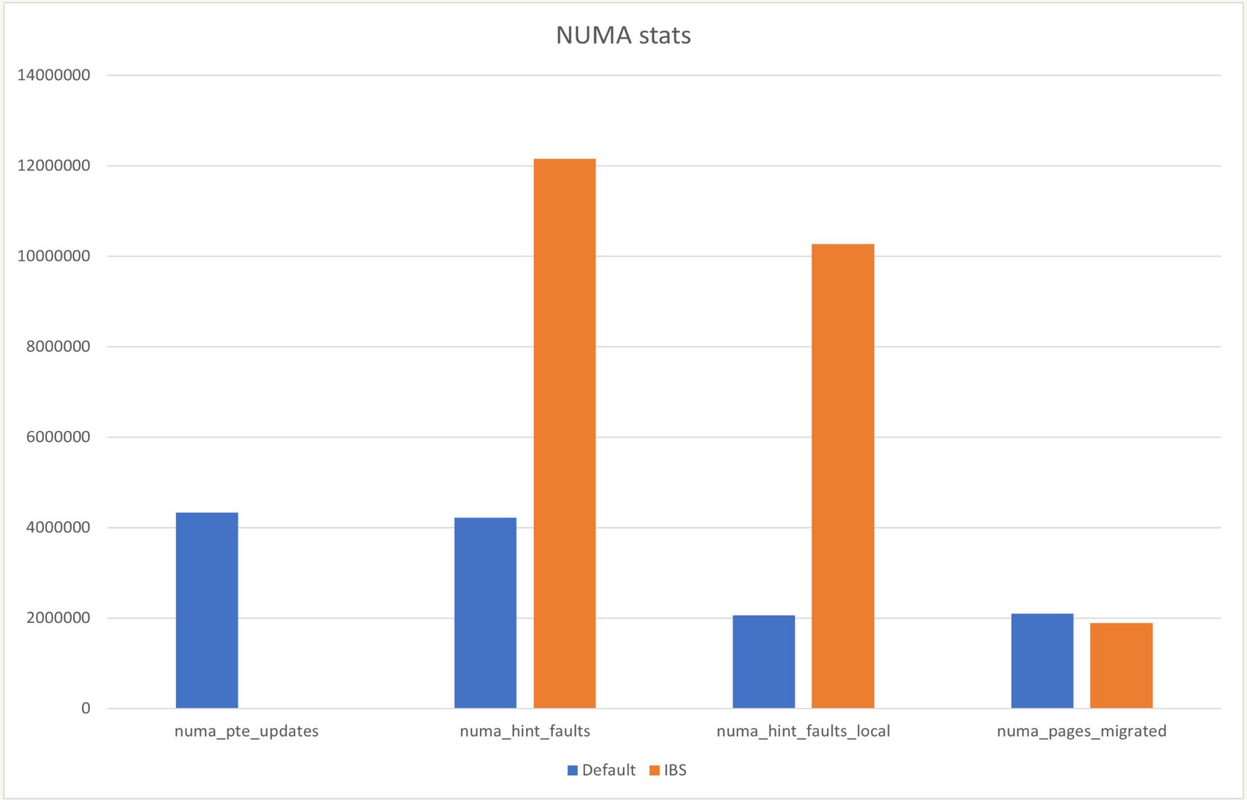
Summary and looking forward

- Sampling based HW hinting will not be able to capture all accesses
  - Lesser number of samples hurt some benchmarks but benefit some others
  - Missing of samples due to HW limitations
- Workload coverage
  - Other interesting benchmarks that should be tried out
- Other subsystems that benefit
  - Reclaim/LRU aging - Alternative to PTE Accessed bit?
  - DAMON - Physical and Virtual address space monitoring using HW hints?
- Better overhead management of HW hinting mechanism
- Separate subsystem to collect HW hints from which other parts of the kernel can consume?

Thanks!

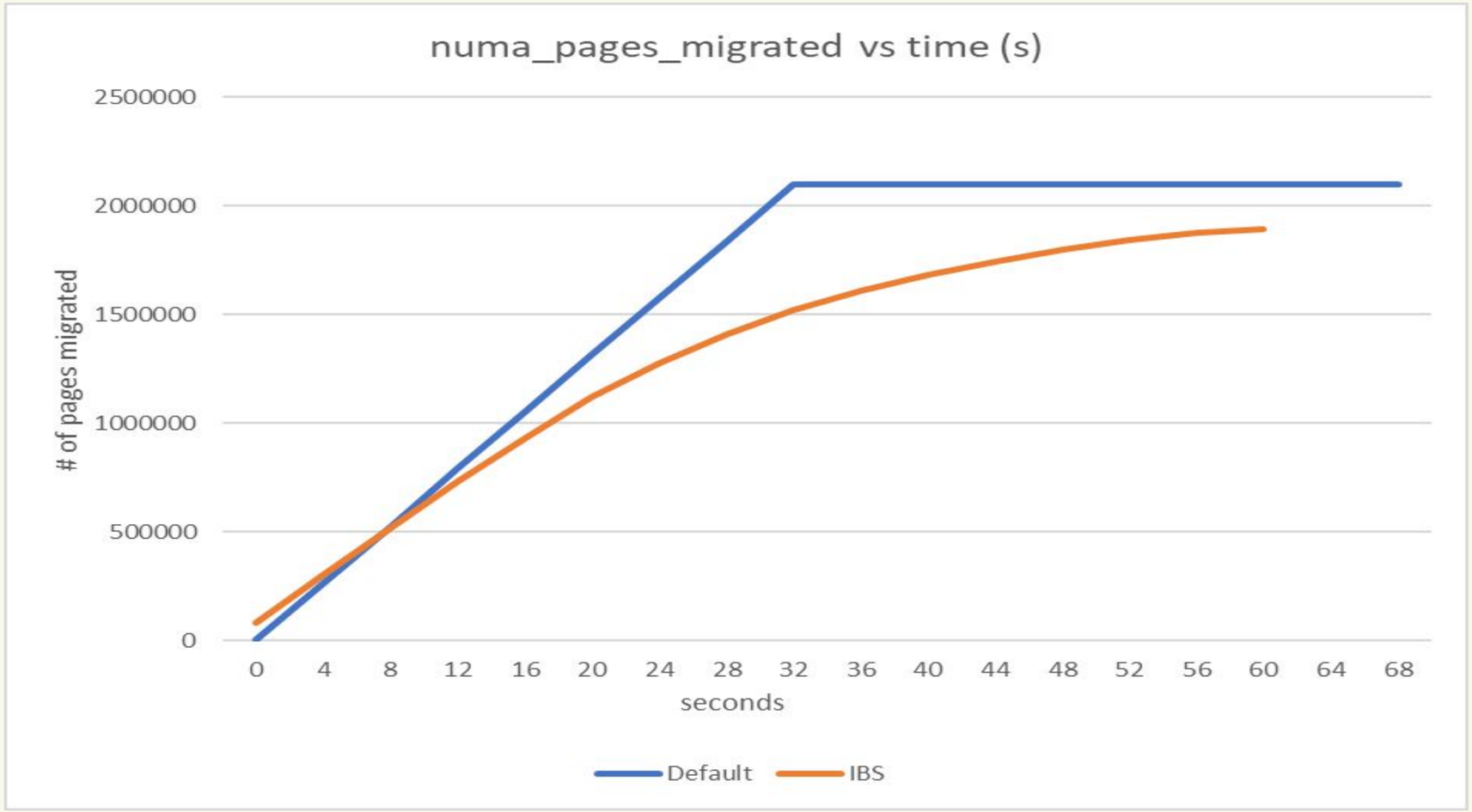
Node1 Regular, numa_balancing=1, delay=0*

	Default	IBS
Benchmark score (s)	71.5	62.1
numa_pte_updates	4328651	0
numa_hint_faults	4224734	12155554
numa_hint_faults_local	2062753	10264439
numa_pages_migrated	2097325	1891114
ibs_nr_events		12186918
ibs_useful_samples		12155614



*No delay b/n accesses, unlike the previous case where there was 100us delay between 1000 accesses.

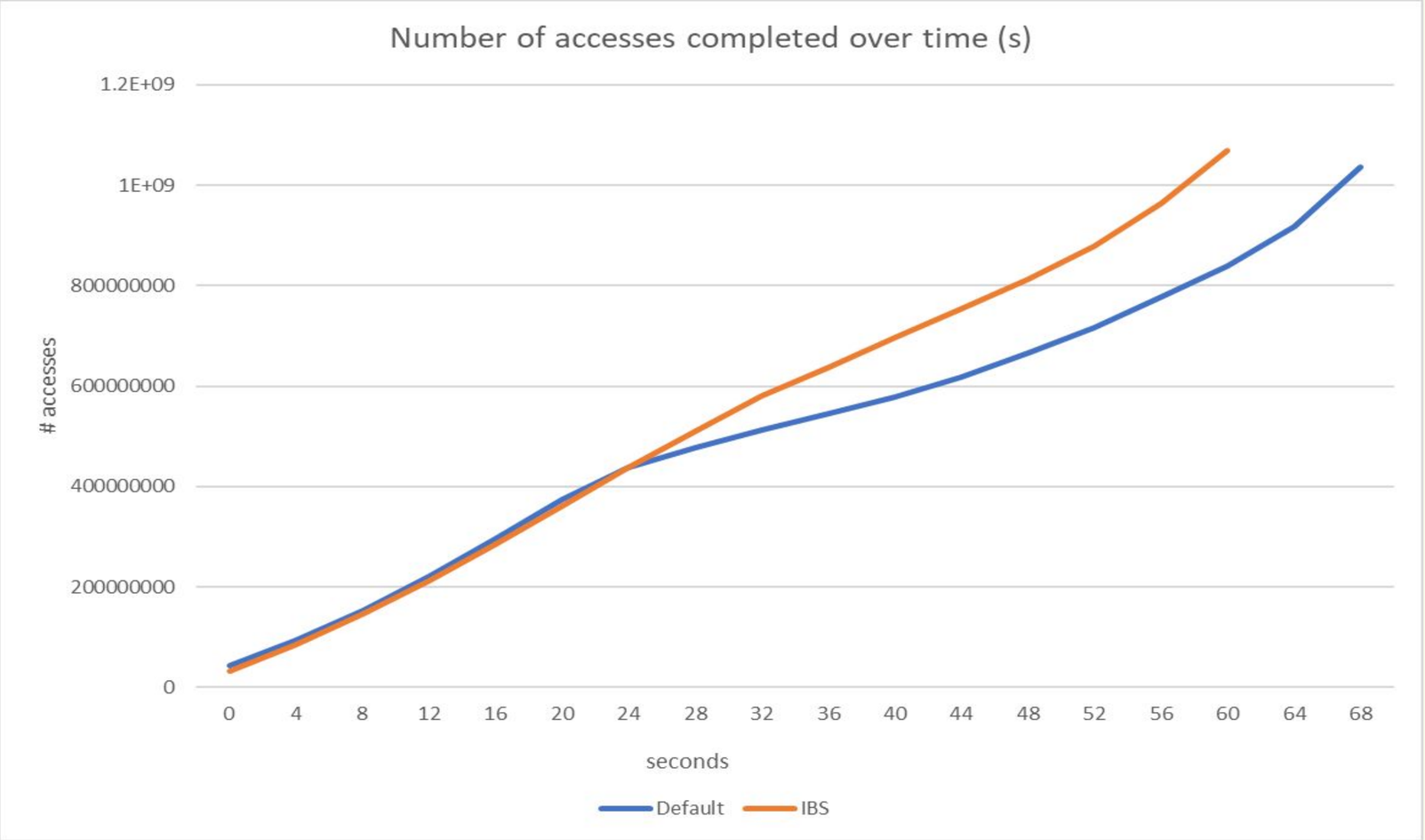
Node1 Regular, numa_balancing=1, delay=0*



- Default case detects and migrates all remote pages quite early
- IBS remote access samples get reported more gradually

* No delay b/n accesses

Node1 Regular, numa_balancing=1, delay =0*

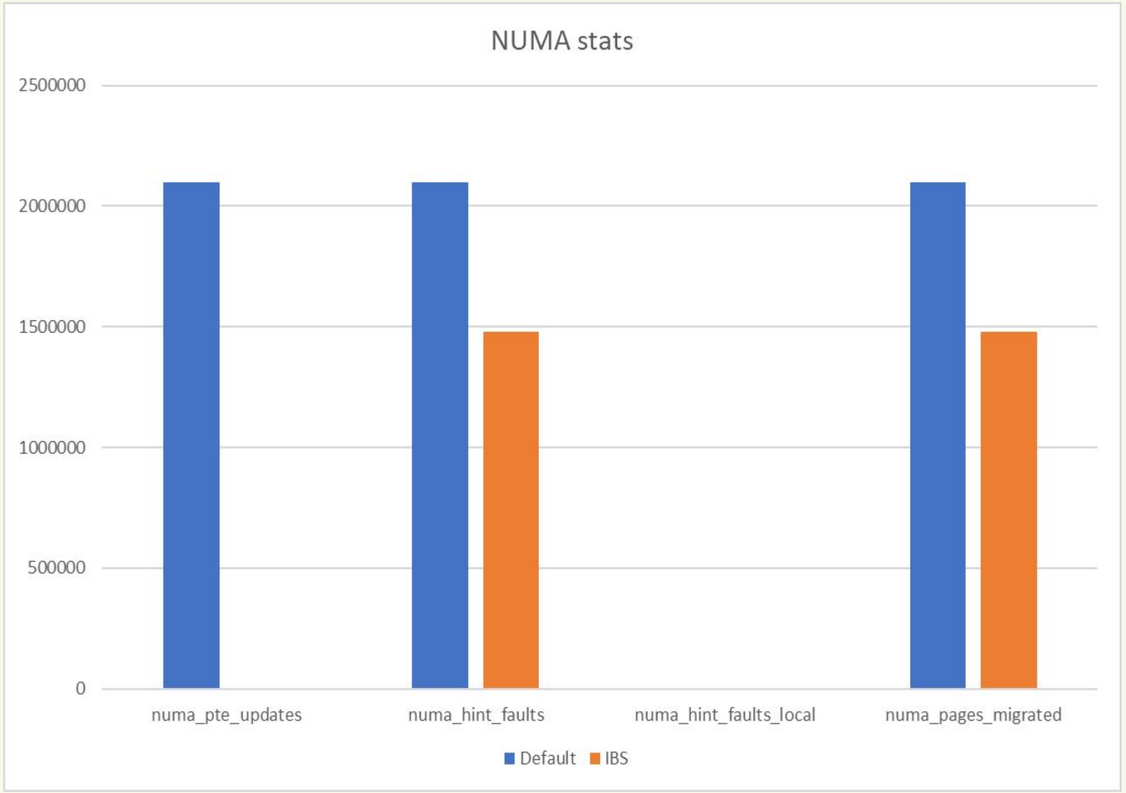


- The progress of default case slows down in the middle
- Too much of migration too fast can slow down the forward progress

* No delay b/n accesses

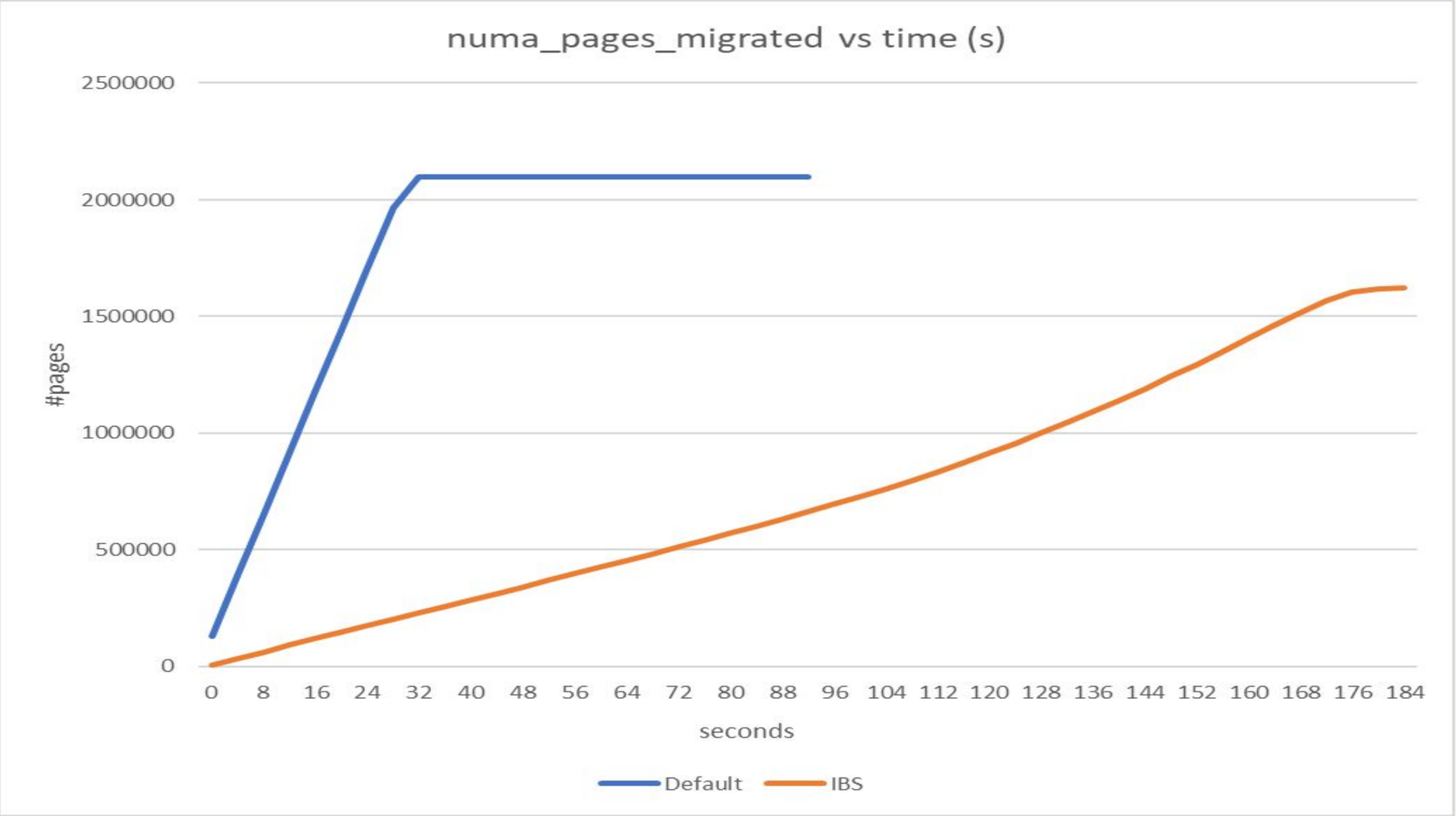
Node2 CXL, numa_balancing=2, delay=0*

	Default	IBS
Benchmark score (s)	97	183.5
numa_pte_updates	2097153	0
numa_hint_faults	2125596	1622043
numa_hint_faults_local	0	0
numa_pages_migrated	2097153	1622037
ibs_nr_events		8384602
ibs_useful_samples		1622053



* No delay b/n accesses

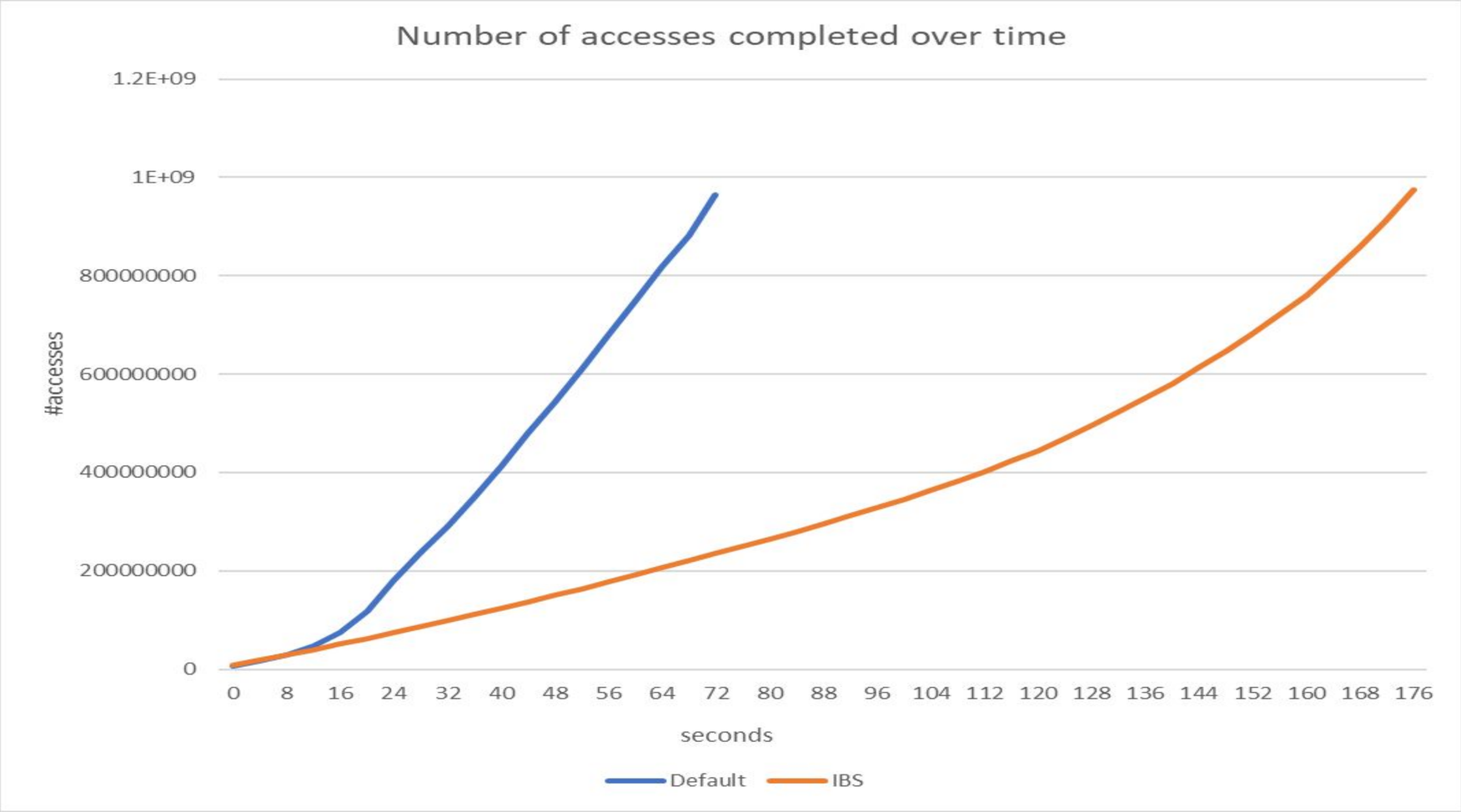
Node2 CXL, numa_balancing=2, delay=0*



- Default case detects and migrates all remote pages quite early
- IBS remote access samples get reported more gradually
- Number of remote samples that IBS reports is lower than desirable, resulting in less migration

* No delay b/n accesses

Node2 CXL, numa_balancing=2, delay=0*

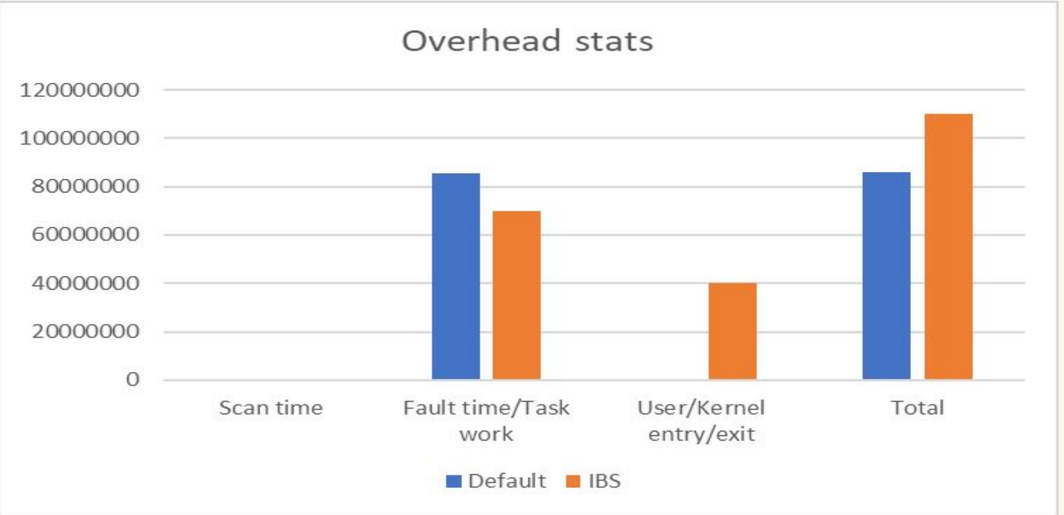


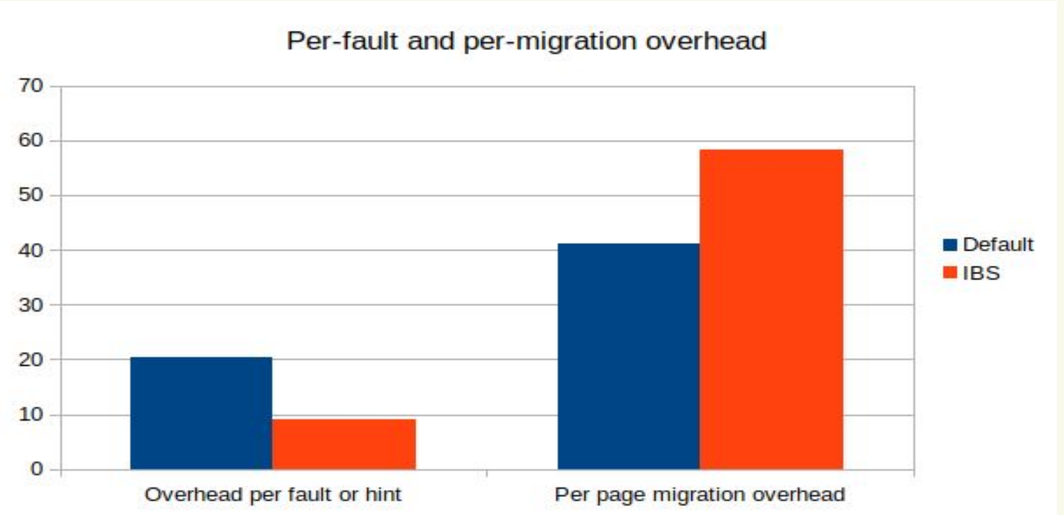
- Slower and gradual migration of remote CXL pages slows down the IBS case
- Despite migration overhead, default case is quicker as CXL latency matters

* No delay b/n accesses

IBS overhead data (in us) for a run with delay=0*

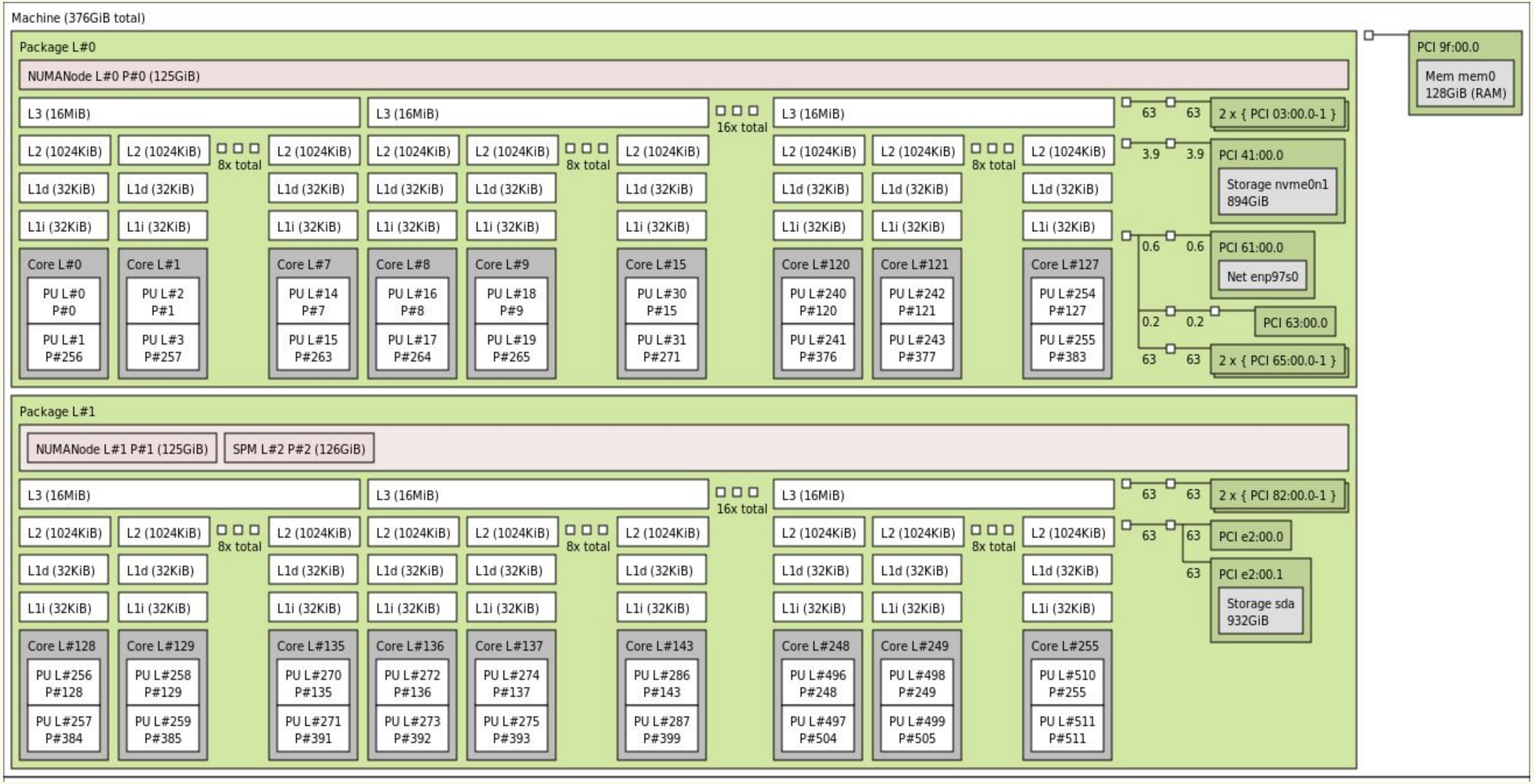
	Default	IBS
Scan time	521000	0
Fault or task work time	85508000	69880000
Kernel entry/exit IBS enable/disable time	0	40077000
Total	86029000	109957000
Nr faults or hints	4224734	12155554
numa_pages_migrated	2097325	1891114
Overhead per fault or access	20.363	9.045
Per-page migration overhead	41.018	58.144

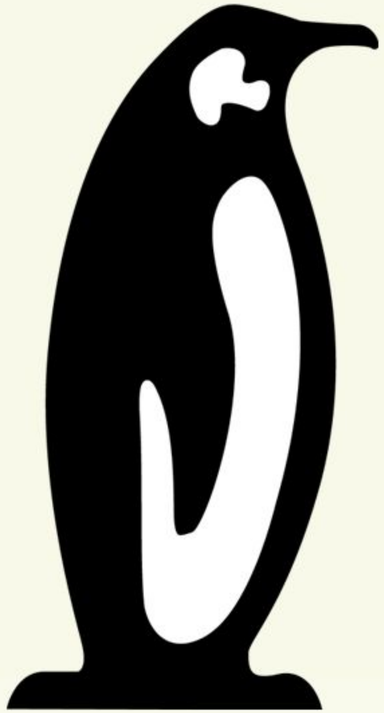




* No delay b/n accesses

Test system topology





Linux Plumbers Conference

Richmond, Virginia | November 13-15, 2023