Argonne NATIONAL LABORATORY

Characterizing the Performance of "Big Memory" on Blue Gene Linux

Kazutomo Yoshii

Mathematics and Computer Science Division

Argonne National Laboratory

Kamil Iskra

P. Chris Broekema (ASTRON)

Harish Naik

Pete Beckman

ZeptoOS Project

- Our main activities:
  - System Noise Study : Selfish suite
  - I/O forwarding : ZOID(ZeptoOS I/O Daemon)
  - Memory Subsystems: Big Memory
  - Performance Analysis: Tau, Ktau
  - Linux based compute node kernel
- Project partner: University of Oregon
- External collaborators: University of Chicago, University of Delaware, ASTRON, University of Tokyo

Blue Gene/P

- Massively parallel computer developed by IBM
- 3rd in the top 500 list and 4 out of 10 (Jun09)
- Highly scalable design
  - Torus, collective and barrier
  - Single clock source
- Very low power consumption
  - 5 out of 10 in the green 500 (Jun09)

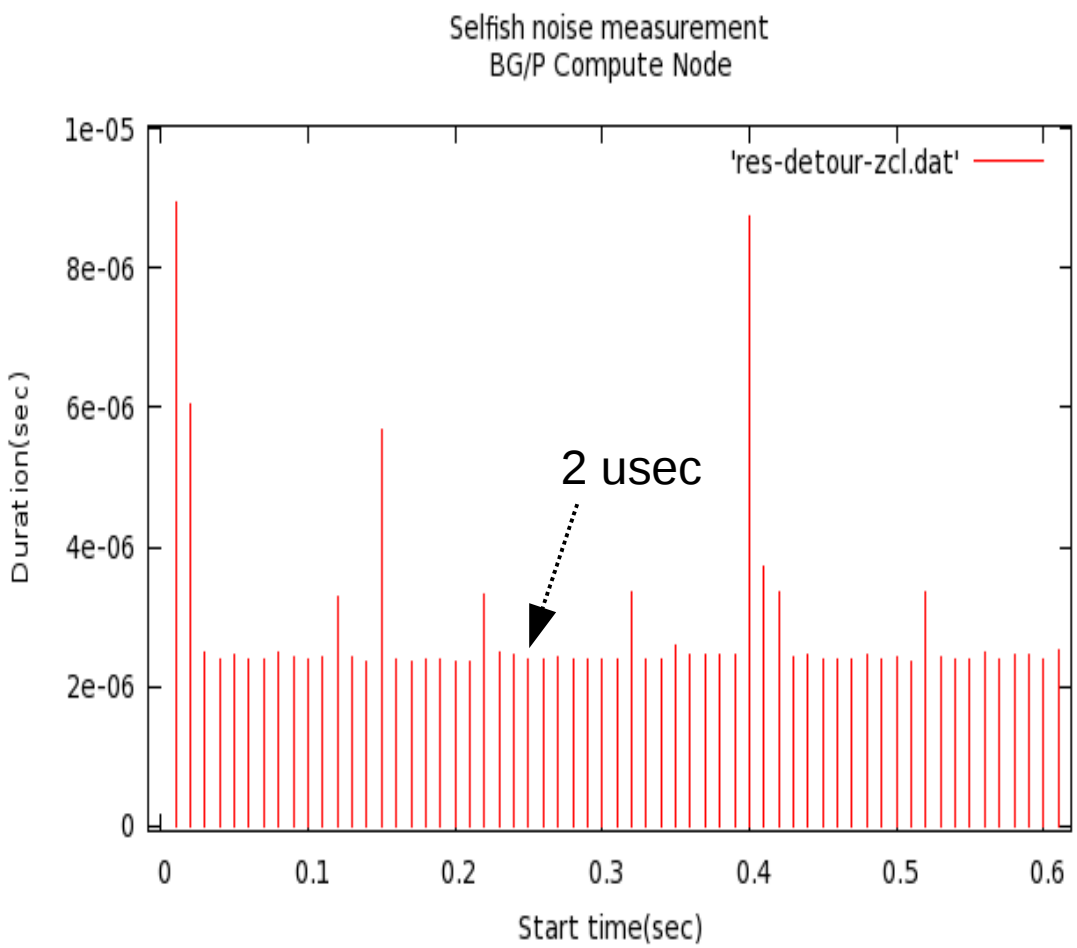
Blue Gene/P Compute Node

- PowerPC 450
  - 32-bit 4-way SMP runs at 850MHz
  - Peak: 3.4 Gflops/core (2 * 2 * 850)
- Compute Node Kernel(CNK) develop by IBM
  - Noise free
  - Thread per core, single user
  - No additional capabilities: remote login, VFS, ...

Can we run Linux on CN?

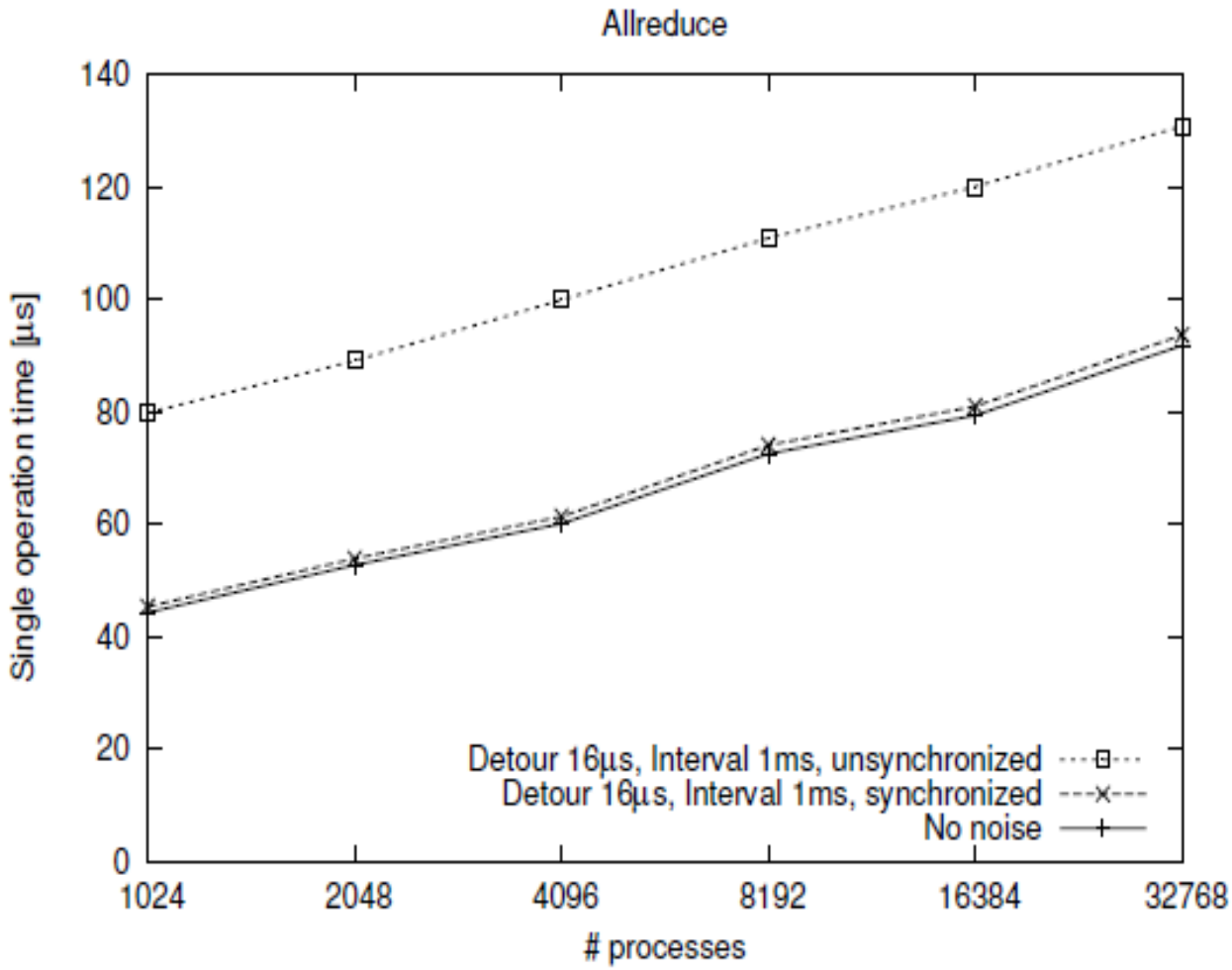
- Very popular operating system
- Linux basically boots on CN
  - although no I/O, device drivers
- Questions:
  - Node level performance?
  - Scalability?

OS Noise (single node)



- Schedule tick 100HZ
- Kernel spends only 0.027%
  - 99.963% for user
- FPU benchmark
  - 3.397 Gflops
  - 99.97% of peak

Noise influence on collective

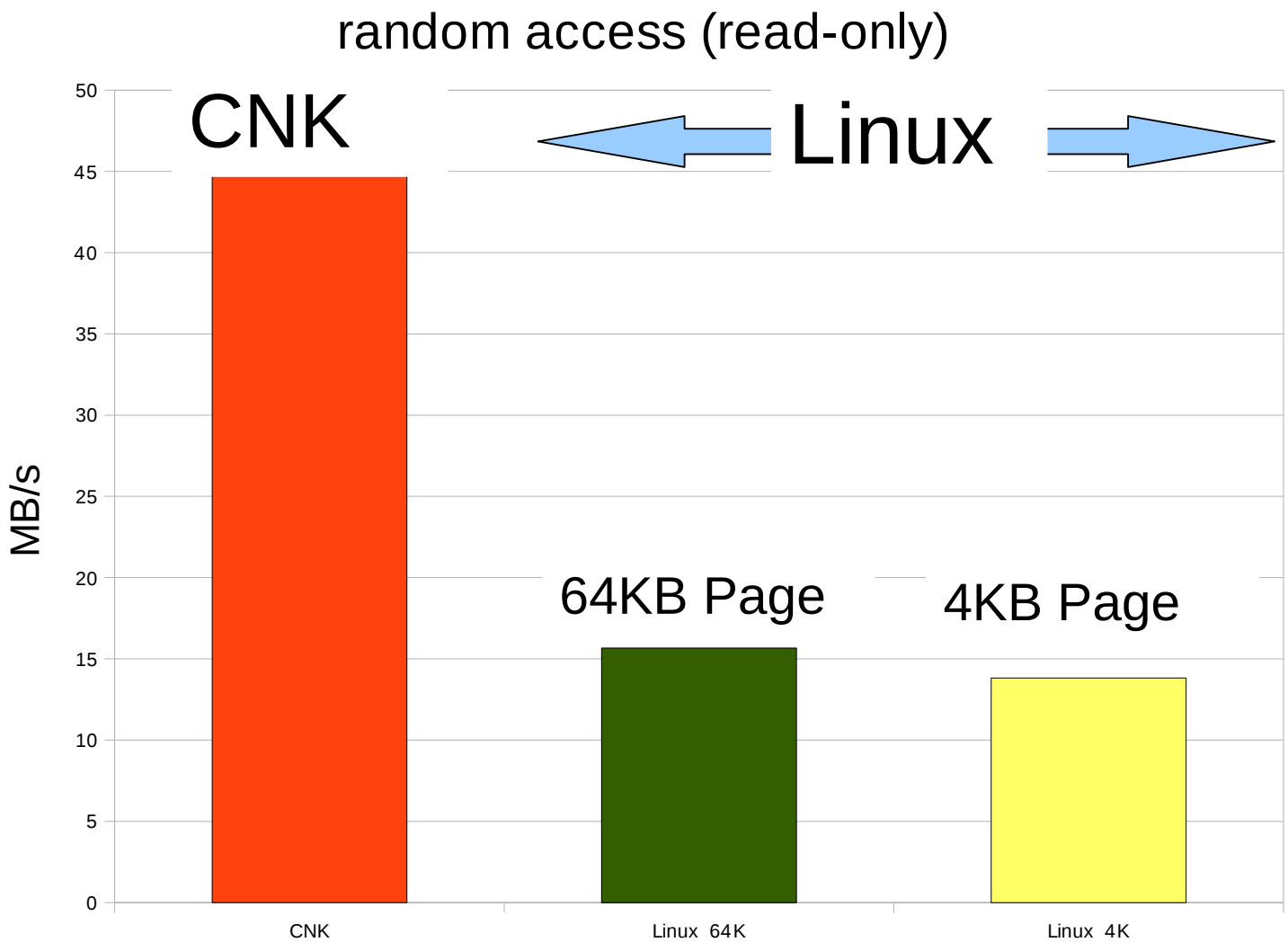


Experimental on BG/L CNK

Injected artificial noise:
16 usec detour every 1ms

1.6% of CPU time
(Linux spends 0.027%)

Memory Benchmark



Why is Linux so slow?

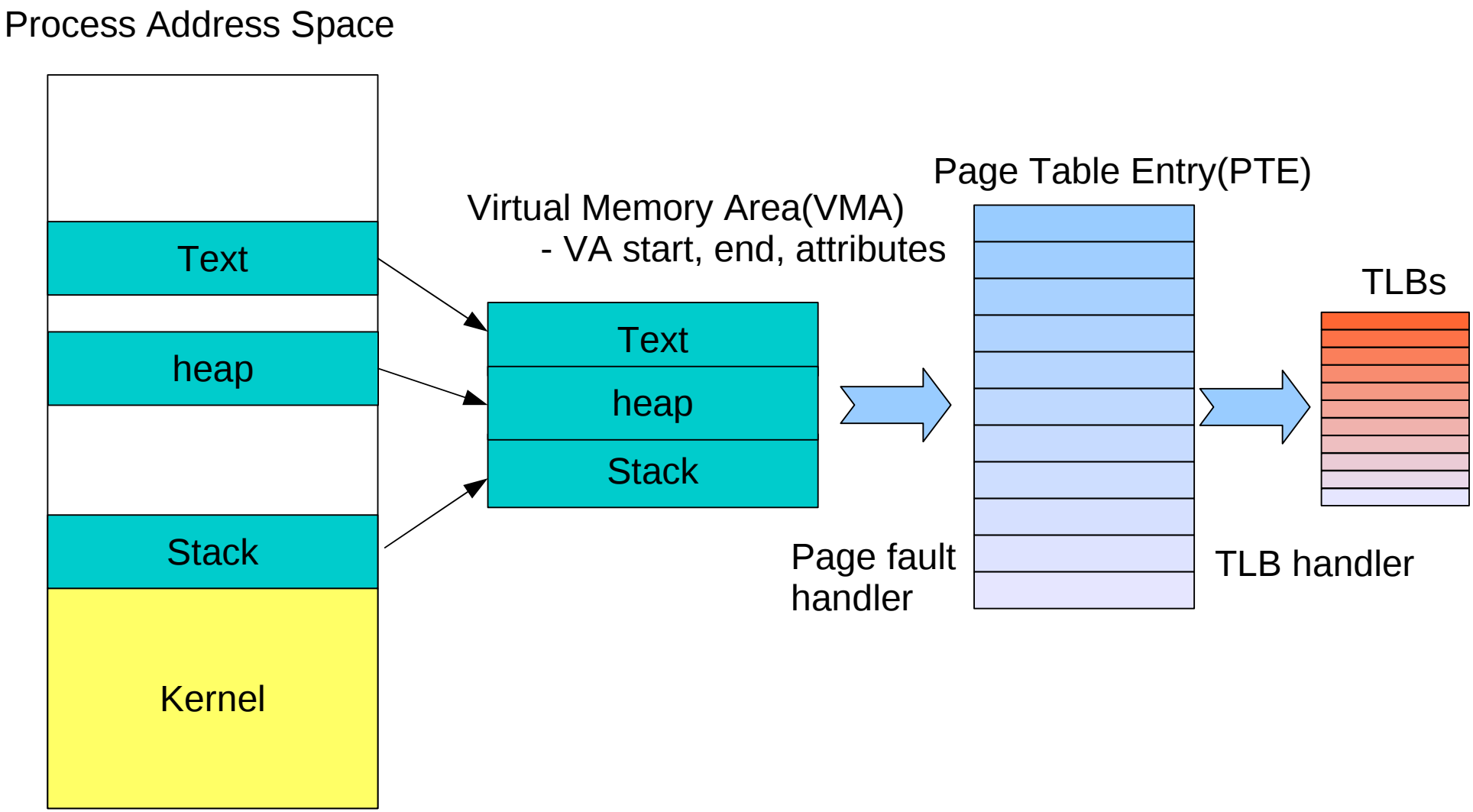
- OS noise is not an issue
- TLB miss is the source of all evil!
  - It costs approx. 0.3 usec
    - TLB exception handler reads PTE from memory to fill TLB
  - 64 TLBs per core
    - Can cover 4MB if 64KB page
  - Impact on a random or stride access pattern
    - Less impact on a streaming access

NAS Serial Benchmarks

	CNK (Mop/s)	Linux 64KB (Mop/s)	Loss(%)
BT	315.30	298.21	5.42
CG	37.15	37.88	-1.97
EP	3.23	3.18	1.55
FT	236.35	218.60	7.51
IS	23.51	5.86	75.07
LU	371.02	334.24	9.91
MG	254.54	250.20	1.71
SP	224.86	217.34	3.34

NOTE:
NAS version 3.3
IBM XL Compiler

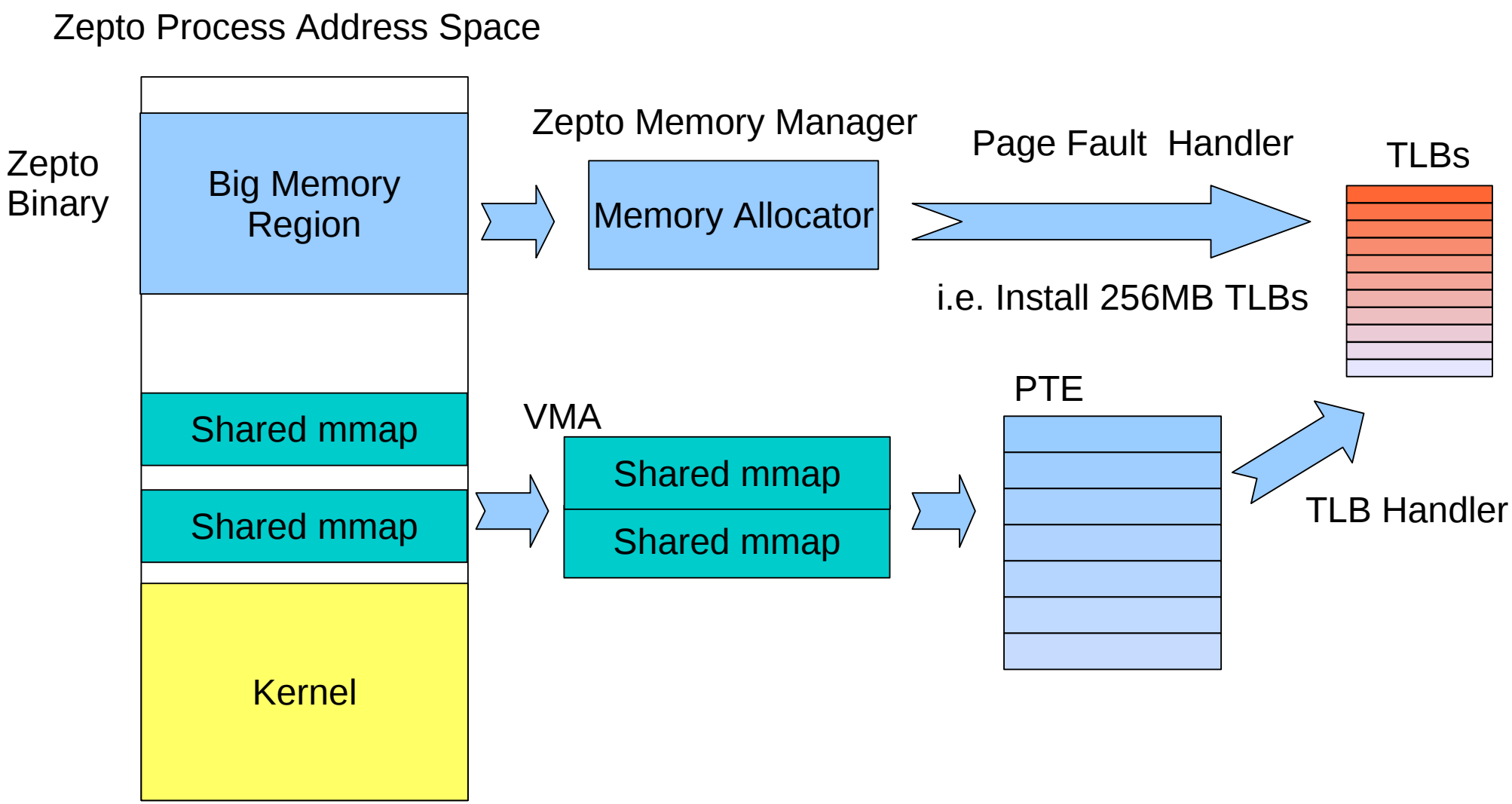
Paged Virtual Memory



Compute Node Environment

- It's not general purpose environment
- Computational process is main character
  - Monopolize CPU resources
  - Context switch is not preferable
  - One thread per core is best
  - Pin down memory for network devices
- Paged Virtual Memory is not appropriate

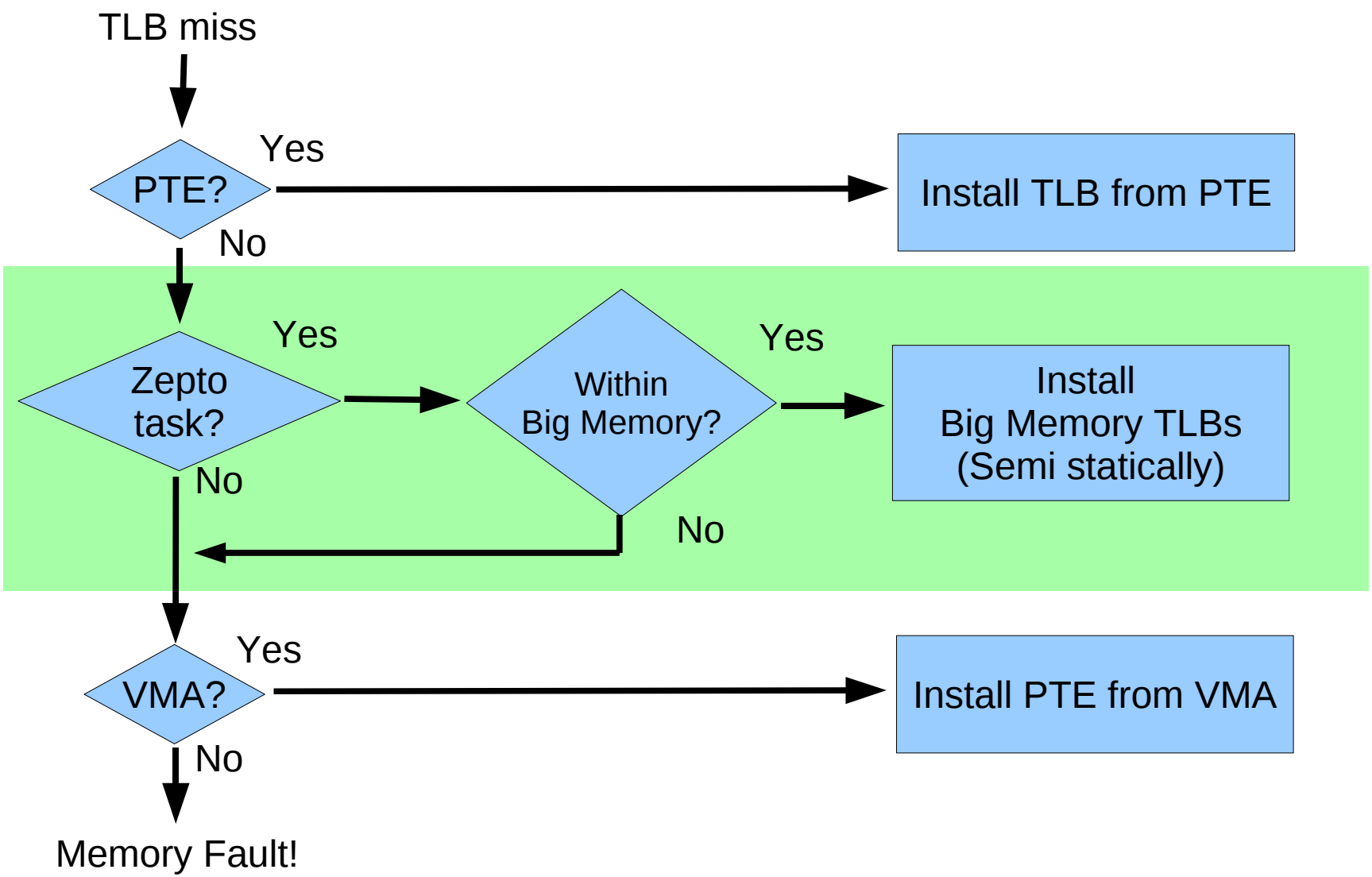
Big Memory


Zepto Process Address Space
Zepto Binary
Big Memory Region
Shared mmap
Shared mmap
Kernel
Zepto Memory Manager
Memory Allocator
Page Fault Handler
i.e. Install 256MB TLBs
TLBs
VMA
Shared mmap
Shared mmap
PTE
TLB Handler

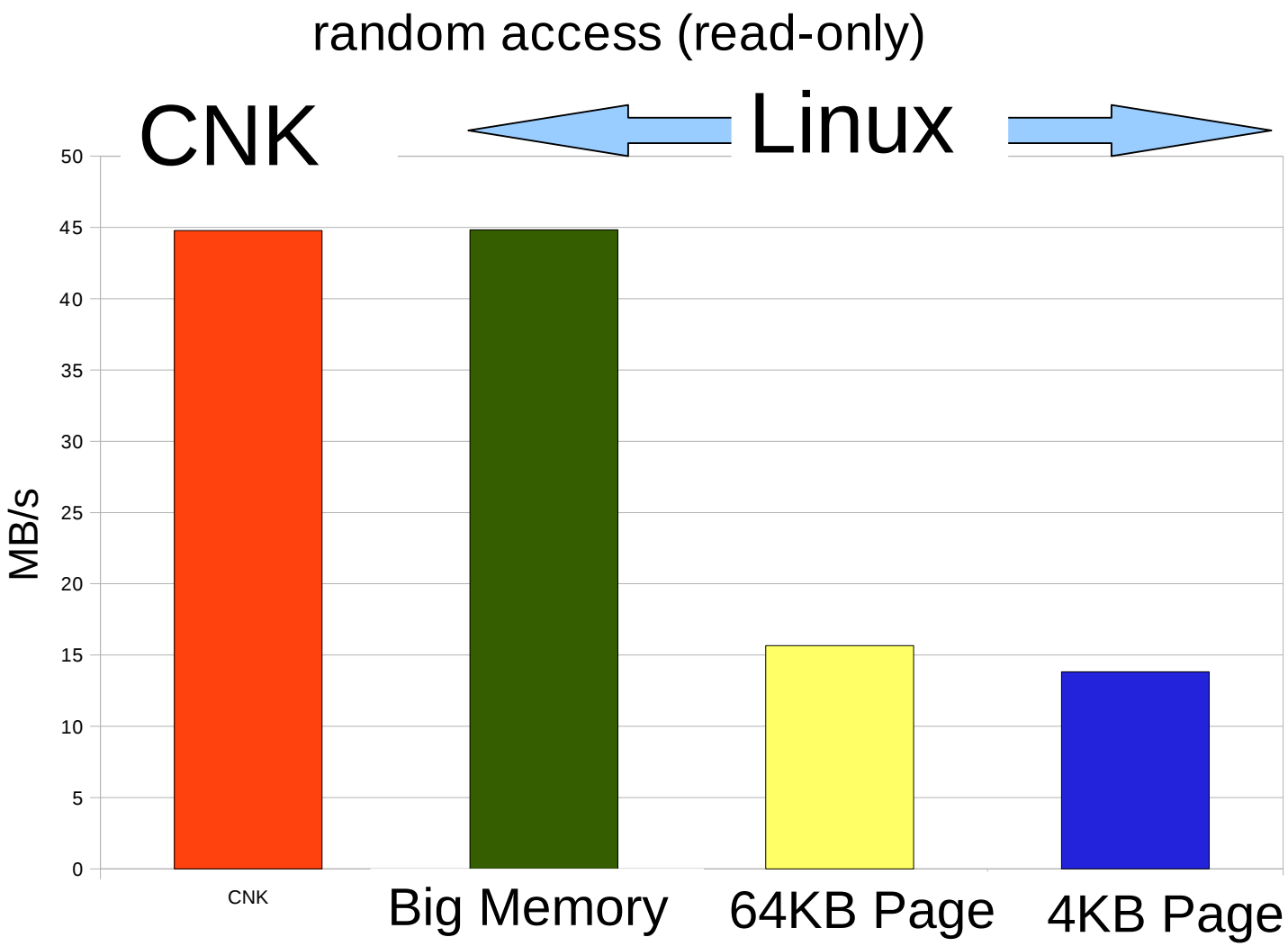
Zepto Binary

- It is a regular ELF binary except ELF header
  - A processor specific flag(e_flags) in the ELF header is altered
  - Zepto kernel checks see if Zepto Binary or not
    - Initialize Big Memory and load the text, data and initial stack into Big Memory
    - Set the personality field in task_struct
- Transparent!
  - No explicit mmap()
  - No recompilation , no dynamic linker trick!

Big Memory Fault-Handling Flow



Single Node Memory Benchmark



NAS Serial Benchmarks

Performance loss(%) against CNK

	Linux 4KB	Linux 64KB	Big Memory
BT	12.09	5.42	0.22
CG	2.53	-1.97	0.08
EP	6.19	1.55	0.31
FT	13.93	7.51	0.06
IS	76.74	75.07	0.26
LU	22.37	9.91	0.09
MG	6.32	1.71	0.21
SP	14.80	3.34	-0.07

NAS Parallel Benchmarks

Big Memory Performance loss(%) against CNK

	1024 Nodes	4096 Nodes
BT	1.17	0.30
CG	0.18	0.28
EP	1.32	1.32
FT	0.34	0.09
LU	0.40	0.65
MG	1.08	0.76
SP	0.44	0.20

NOTE:
NPB 3.3 / MPICH 1.0.7 / DCMF 1.0
IBM XL Compiler
SMP mode

Parallel Ocean Program(POP)

	CNK(sec)	Zepto (sec)	Loss(%)
64	196.62	197.26	0.33
128	105.69	105.59	-0.09
256	57.37	57.00	-0.64
512	34.98	34.39	-1.69
1024	22.37	21.87	-2.24
2048	16.74	16.32	-2.51
4096	14.54	14.10	-3.03

NOTE:
POP 2.0.1 / X1 benchmark data set
IBM XL compiler
SMP mode

System call overhead

- The gettimeofday() system call takes 3.91 usec on CNK while 0.51 usec on Linux
- Profiling adds more overhead
  - With Tau enabled, POP took 131 sec on CNK while 120 sec on Linux at 128 node

LOFAR node processing (I/O node)

	Stock – 16 bit	Zepto – 16 bit	Zepto – 4 bit
Receive UDP/IP packets	1.44	1.44	1.22
Copy data to ring buffer	1.80	0.27	0.37
Send ring buffer to CN	1.40	0.52	0.61
Receive data from CN	1.00	0.10	0.35
Send results to storage	0.40	0.32	0.64
Total system load	151.0%	66.5%	79.7%

Choice of CPU for Supercomputer

- Based on commodity CPU cores
  - Intel Xeon, AMD Opetron, IBM Power
  - Software compatibility
  - Fewer bugs
  - Less cost
- Existing MMUs are designed for general purpose, not for supercomputer !
  - Network devices, memory subsystems, FPU are evolving while MMUs are not.

Conclusions

- Results
  - Increase memory performance
  - Porting communication library became easier
- Future works
  - Big memory on other CPU
  - Extended to DUAL, VN node mode
  - Tickless kernel
    - at least for computational process

Thank you!

NAS Parallel Benchmarks
1024 nodes - gcc

	CNK (Mop/s)	Zepto (Mop/s)	Loss(%)
IS	3.90	3.92	-0.470
CG	15.38	15.34	0.268
MG	131.79	131.23	0.426
FT	94.33	94.13	0.216
LU	39.93	39.66	0.666
EP	2.44	2.44	0.126
SP	103.52	103.23	0.283
BT	161.37	160.92	0.280

NOTE:
Total Mop/s
NPB 3.3 / MPICH 1.0.7 / DCMF 1.0 for Linux, 2.0 for CNK
SMP mode

NAS Parallel Benchmarks 1024 nodes – IBM XL compiler

	CNK(Mop/s)	Zepto (Mop/s)	Loss(%)
BT	304.03	300.48	1.17
CG	28.18	28.13	0.18
EP	3.78	3.73	1.32
FT	212.57	211.84	0.34
LU	288.99	287.84	0.40
MG	241.15	238.54	1.08
SP	149.24	148.58	0.44

NOTE:
Mop/s per proc
NPB 3.3 / MPICH 1.0.7 / DCMF 1.0
SMP mode

NAS Parallel Benchmarks 4096 nodes – IBM XL compiler

	CNK(Mop/s)	Zepto (Mop/s)	Loss(%)
BT	241.53	240.81	0.30
CG	14.35	14.31	0.28
EP	3.78	3.73	1.32
FT	90.96	90.88	0.09
LU	211.99	210.88	0.52
MG	213.86	212.23	0.76
SP	132.60	132.34	0.20

NOTE:
Mop/s per proc
NPB 3.3 / MPICH 1.0.7 / DCMF 1.0
SMP mode