PARALLELIZING ATHENA MP EVENT RECONSTRUCTION

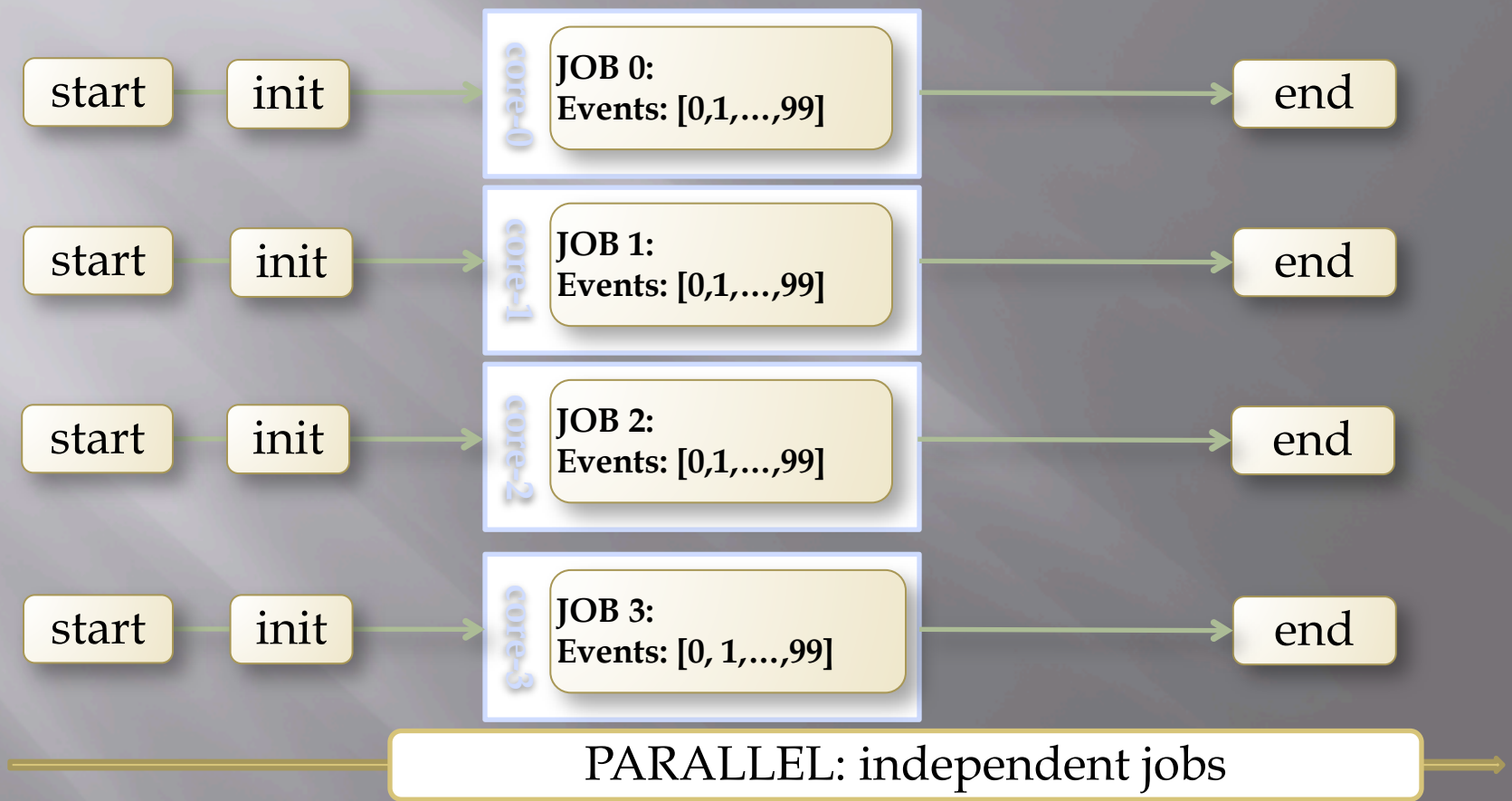
Keith R. Jackson

Lawrence Berkeley National Lab

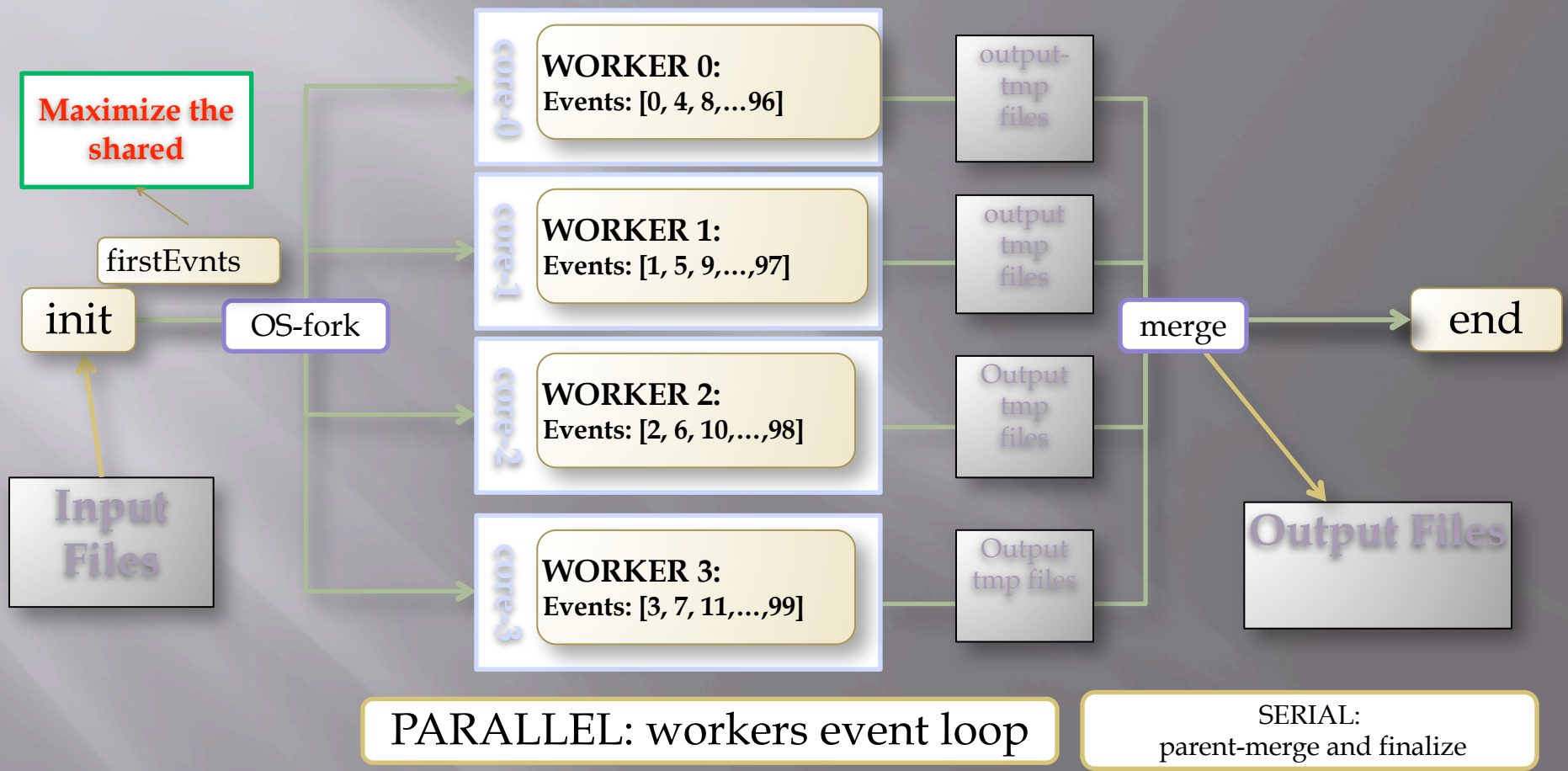
Goals

- Benchmark athenaMP on multi-core
  - Investigate application event processing event processing scalability
- Identify bottlenecks to scalability and propose strategies to mitigate them
  - Test at higher core counts then in production

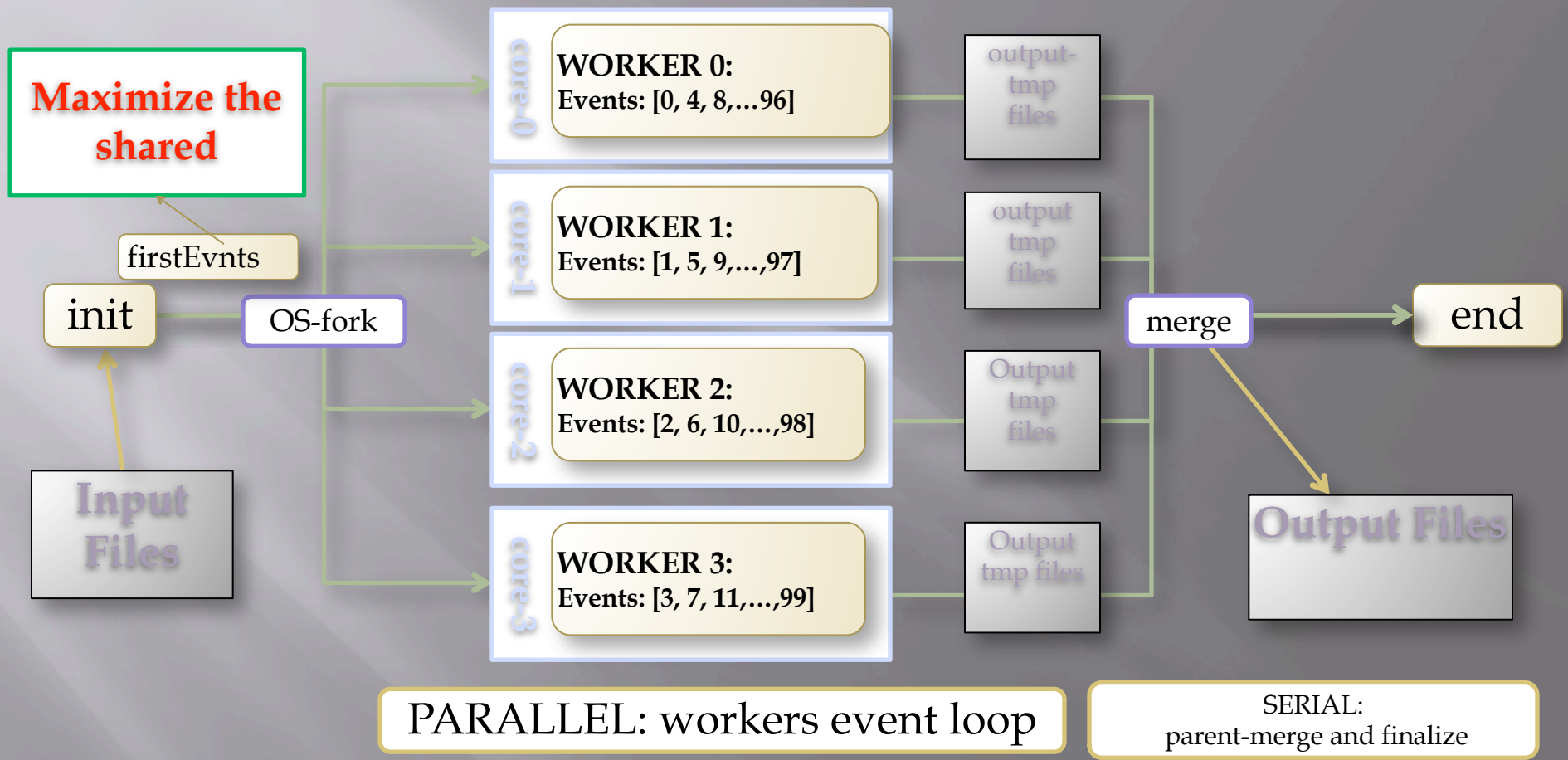
Athena MJ


start
init
core-0
JOB 0:
Events: [0,1,...,99]
end
start
init
core-1
JOB 1:
Events: [0,1,...,99]
end
start
init
core-2
JOB 2:
Events: [0,1,...,99]
end
start
init
core-3
JOB 3:
Events: [0, 1,...,99]
end
PARALLEL: independent jobs

Athena MP


Maximize the shared
firstEvnts
init
OS-fork
Input Files
core-0
WORKER 0:
Events: [0, 4, 8,...96]
core-1
WORKER 1:
Events: [1, 5, 9,...,97]
core-2
WORKER 2:
Events: [2, 6, 10,...,98]
core-3
WORKER 3:
Events: [3, 7, 11,...,99]
output-tmp files
output tmp files
Output tmp files
Output tmp files
merge
end
Output Files
PARALLEL: workers event loop
SERIAL:
parent-merge and finalize

Athena MP


Maximize the shared
firstEvnts
init
OS-fork
Input Files
core-0
WORKER 0:
Events: [0, 4, 8,...96]
core-1
WORKER 1:
Events: [1, 5, 9,...,97]
core-2
WORKER 2:
Events: [2, 6, 10,...,98]
core-3
WORKER 3:
Events: [3, 7, 11,...,99]
output-tmp files
output tmp files
Output tmp files
Output tmp files
merge
end
Output Files
PARALLEL: workers event loop
SERIAL:
parent-merge and finalize

Testing Platforms

Name	CPU	Cache	# CPU's	# cores	Mem
Turing	AMD Opteron 8384	L1: 128KB L2: 512KB L3: 6MB/cpu	8	32	256GB
Coors	Intel Xeon X5550	L1: 32KB (inst & data) L2: 256KB L3: 8MB/cpu	2	8	24GB
VoAtlas	Intel Xeon E5410	L1: 32KB (inst & data) L2: 32KB L3: 8MB/cpu	2	8	16GB

Tools Used

- sar: Measure number of reads and writes from physical disks, total I/O ouptut, system load
- vmstat: Measure memory performance including, total used, paging, free memory
- IPM (Integrated Performance Monitoring) to measure total amount of time spent in I/O vs. computation
- numastat/numactl controls and reports on NUMA settings

Intel Tools

- Performance Tuning Utility (PTU)
  - Uses a Linux kernel module to provide a sampling profiler
  - Captures information from the hardware counters available on Intel chips
    - Information varies between processor families
  - Large number of instruction cache misses, relatively few data cache misses
  - Not using SSE division instructions
- Intel compilers
  - Link level optimizations
  - Can use output from PTU to optimize hot spots

Why Test?

Why Test?

- Our initial assumptions were that athenaMP would be either:
  - I/O bandwidth limited
  - Memory bandwidth limited

Why Test?

- Our initial assumptions were that athenaMP would be either:
  - I/O bandwidth limited
  - Memory bandwidth limited
- Testing showed that we were:

Why Test?

- Our initial assumptions were that athenaMP would be either:
  - I/O bandwidth limited
  - Memory bandwidth limited
- Testing showed that we were:

Why Test?

- Our initial assumptions were that athenaMP would be either:
  - I/O bandwidth limited
  - Memory bandwidth limited
- Testing showed that we were:

WRONG!!