

AutoNUMA25 vs AutoNUMA25 MoF

Red Hat, Inc.

Andrea Arcangeli
aarcange at redhat.com

21 Sep 2012



Objective

- Run the autonuma-benchmark (tag 0.1 d20e5a0) to compare:
 - AutoNUMA25 cdcfc6b
 - Based on 3.6-rc6+ 4651afb
 - AutoNUMA25 MoF
 - Based on AutoNUMA25
- Run on 2 nodes with HT enabled
- Pending run on 8 nodes
- Pending more benchmarks
 - Autonuma-benchmark not best to show the benefits of async migration

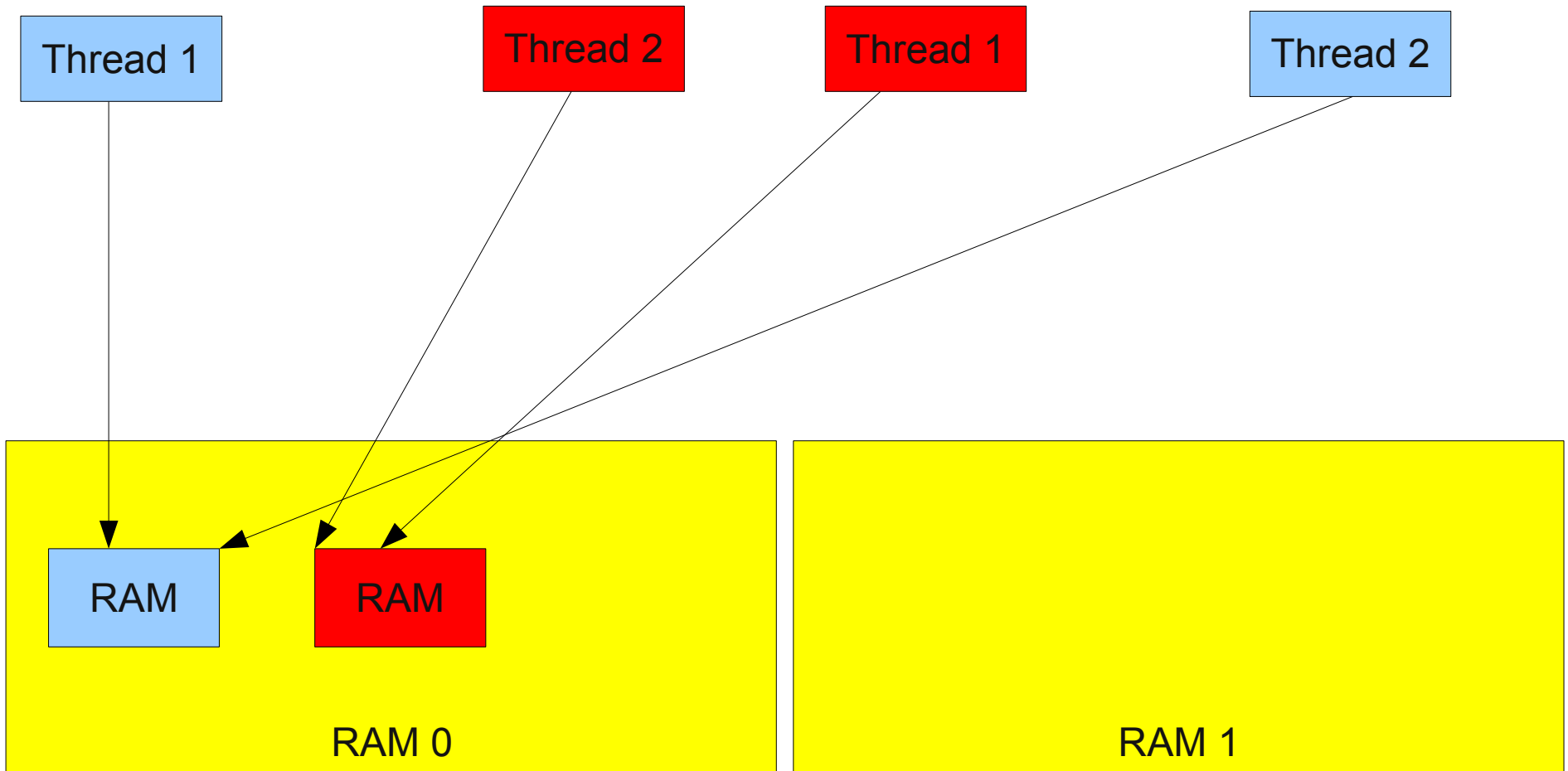


AutoNUMA-benchmark tests

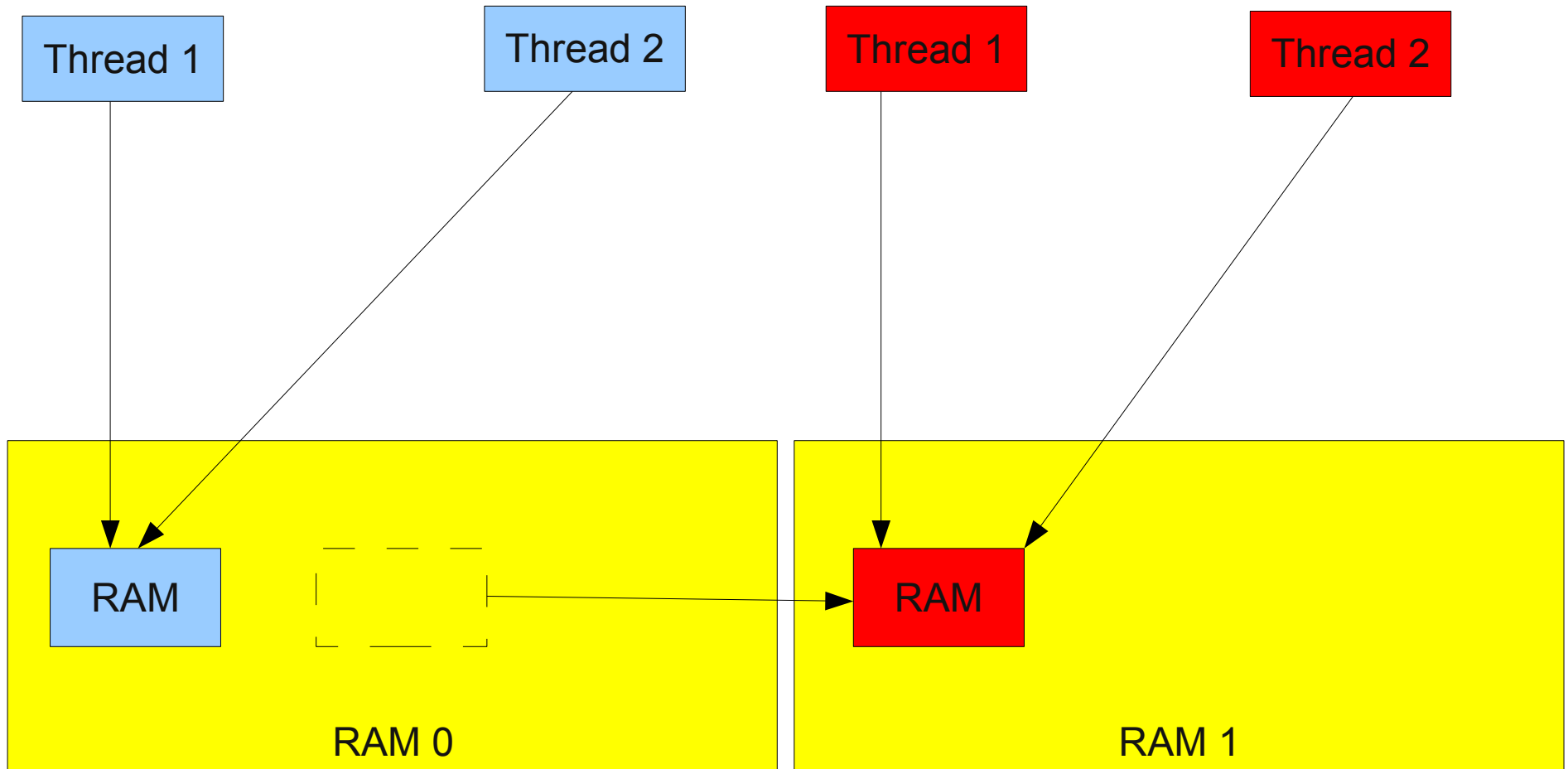
- › P =# of processes, T =# of threads per process, M =memory per process
- › numa01
 - › $M=3\text{GiB}$ $P=2$, $T=\text{nr_cpus}/2$, all threads share all process memory
- › numa01_THREAD_ALLOC
 - › $M=3\text{GiB}$, $P=2$, $T=\text{nr_cpus}/2$, all threads use per-thread local memory
- › numa02
 - › $M=1\text{GiB}$, $P=1$, $T=\text{nr_cpus}$, all threads use per-thread local memory
- › numa02_SMT
 - › $M=1\text{GiB}$, $P=1$, $T=\text{nr_cpus}/2$, all threads use per-thread local memory
 - › The kernel to get this right must not use more than one HT thread per core, and in turn it must decide to split the load over two NUMA nodes even if the load would fit in a single NUMA node
 - › Testing with $T=\text{nr_cpus}/4$ and smaller T values, would also be interesting, but if the kernel behaves well with $T=\text{nr_cpus}/2$ there's a good chance it'll behave sanely with more than half of the CPUs idle too
- › More will be added...



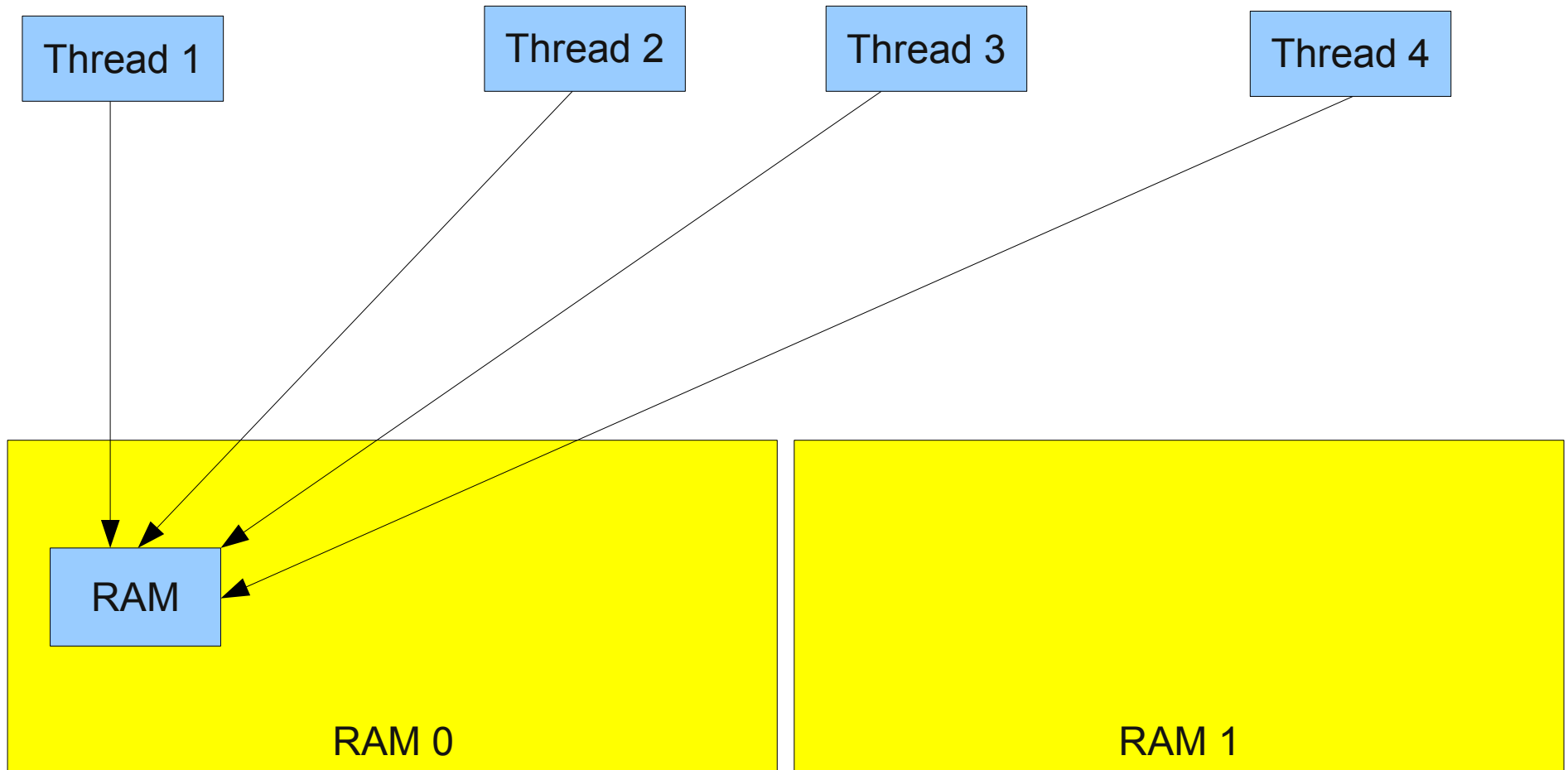
numa01 (startup)



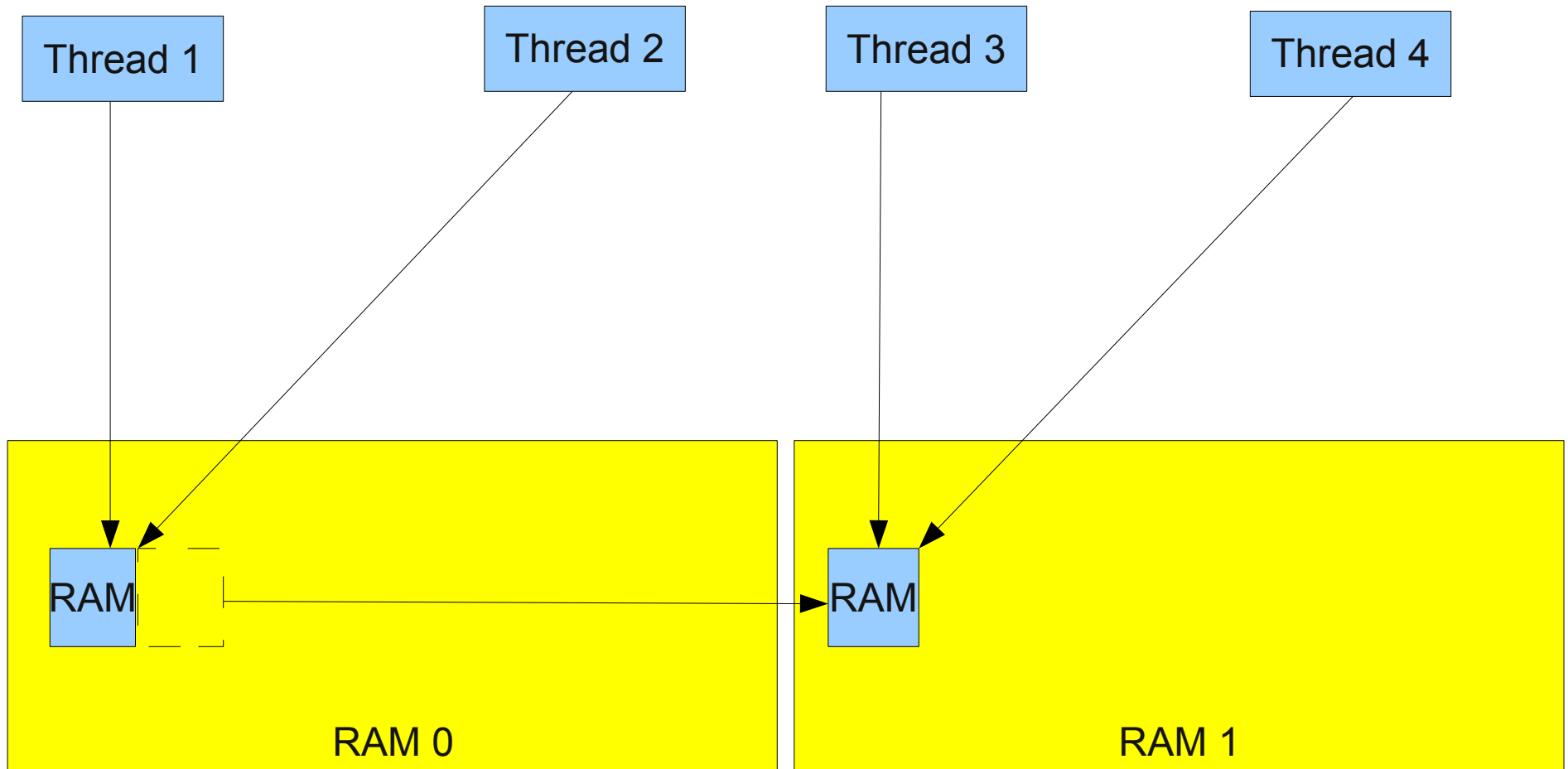
numa01 (converged)



numa02 (startup)



numa02 (converged)



autonuma-benchmark

```
$ git clone git://gitorious.org/autonuma-benchmark/autonuma-benchmark.git
```

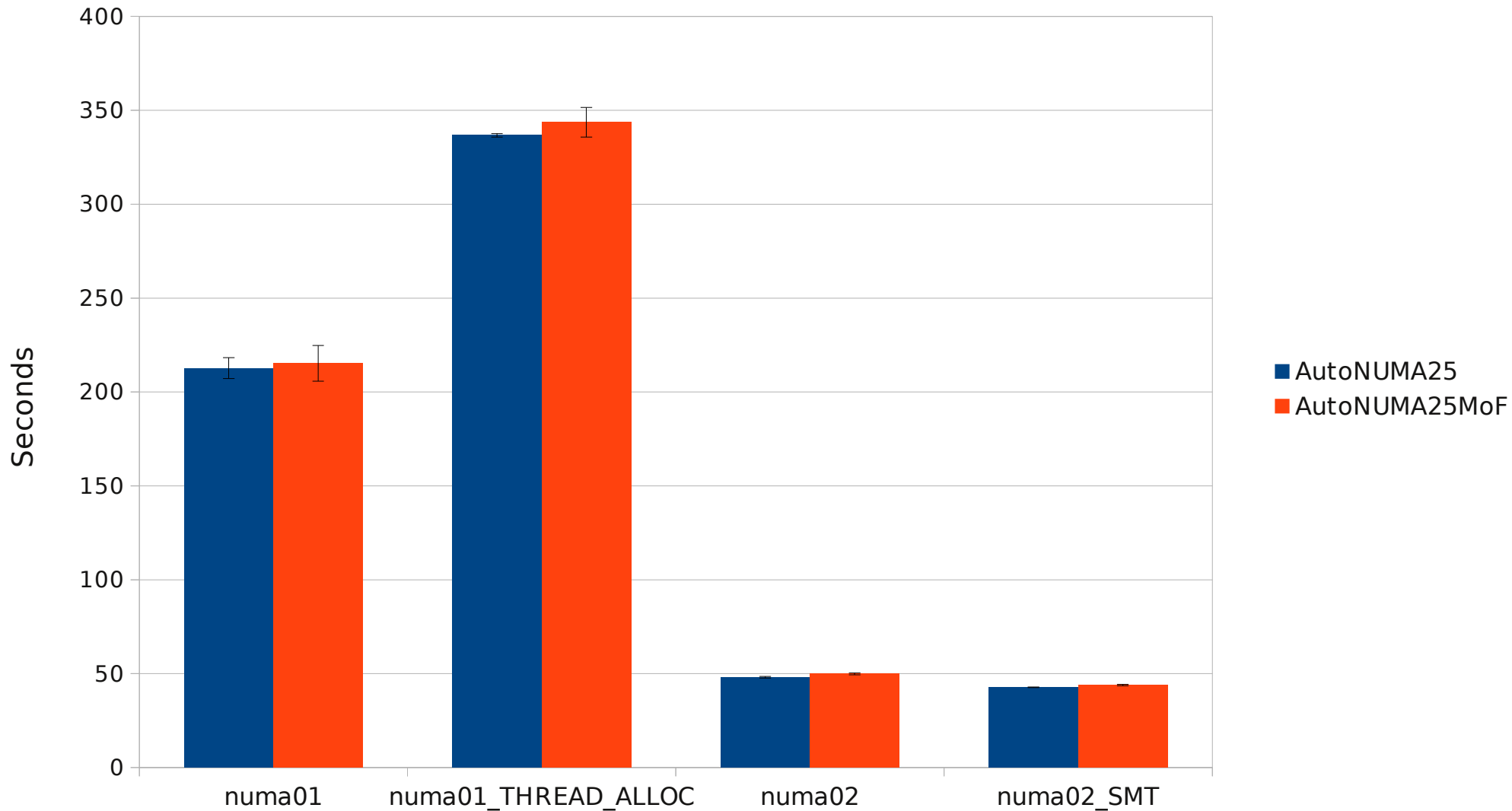
```
$ cd autonuma-benchmark
```

```
$ sudo ./start_bench.sh -s -t
```



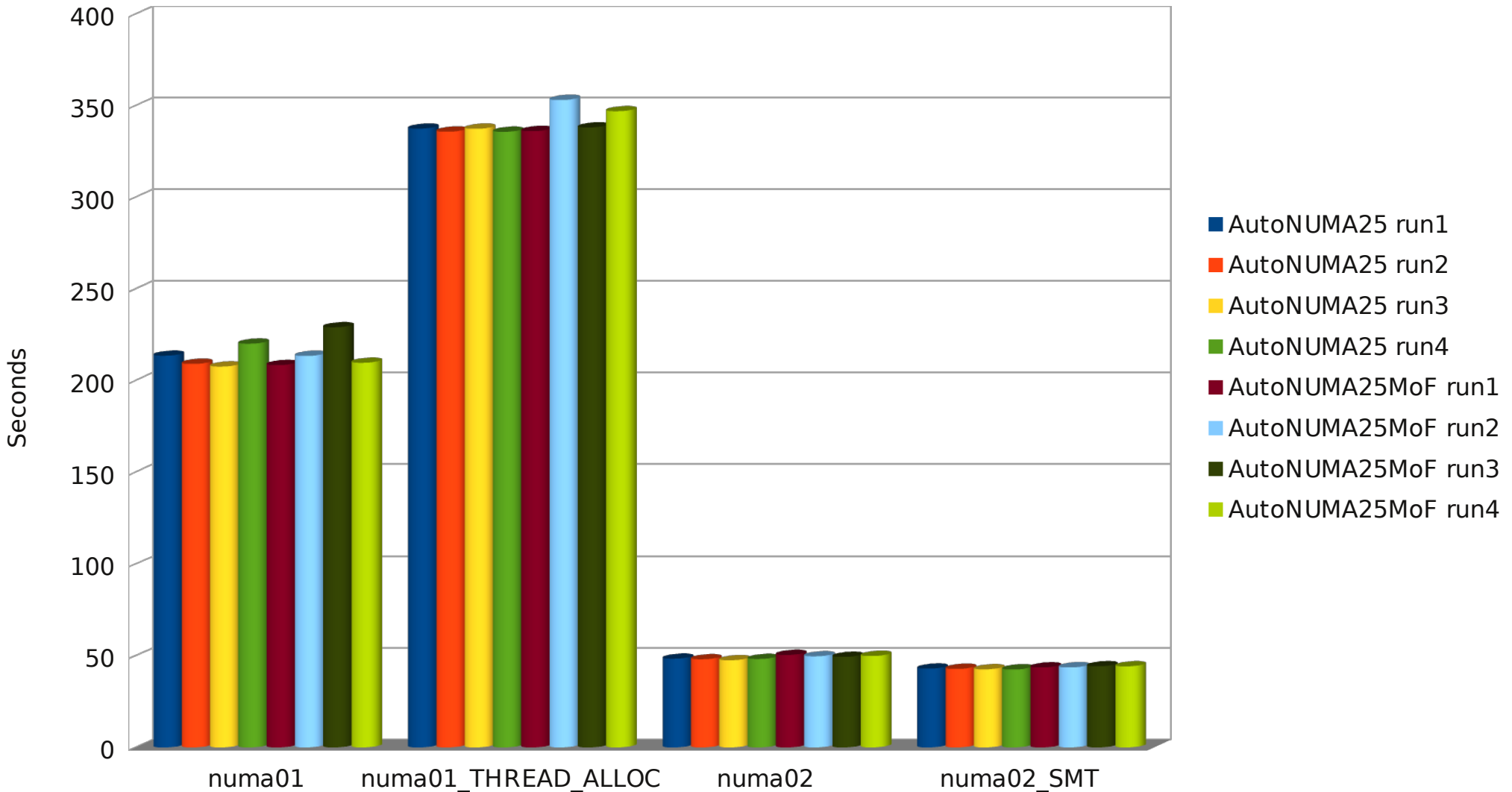
4 runs, average and stdev

Lower is better



4 runs of each test, all results

Lower is better



Raw results

Lower is better Average seconds	numa01	numa01_THREAD_ALLOC	numa02	numa02_SMT
AutoNUMA25	212.72	336.6725	48.07	42.755
AutoNUMA25MoF	215.2875	343.65	49.87	43.9425

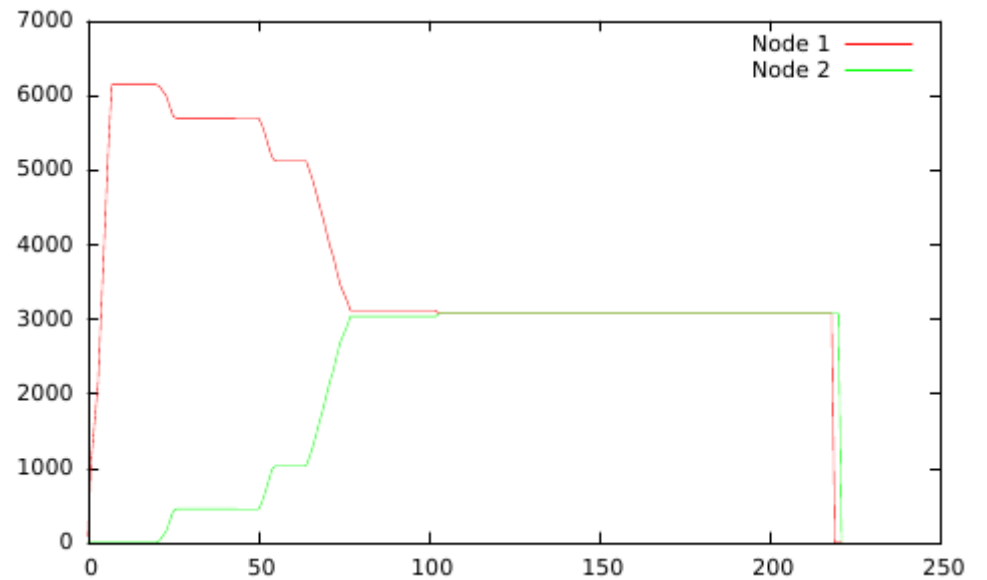
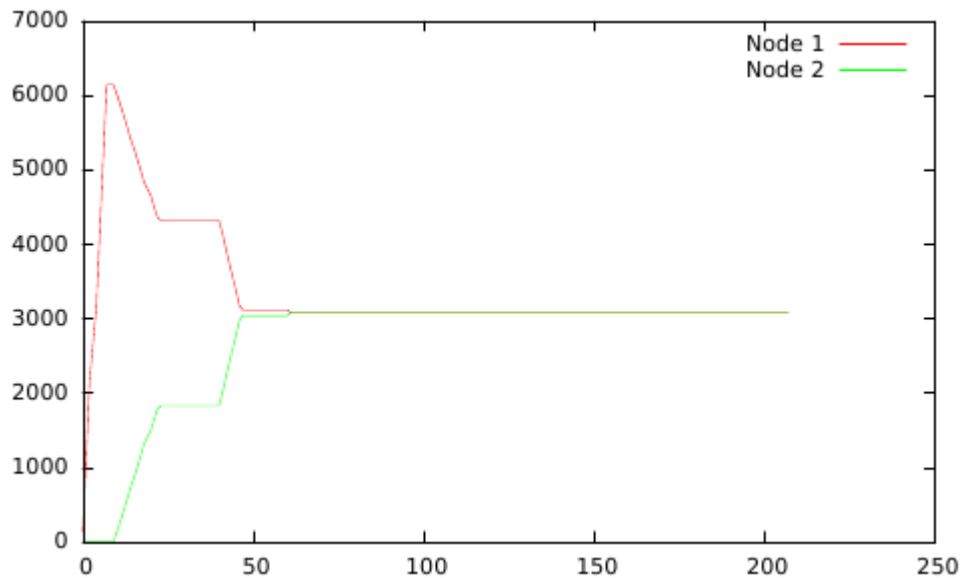
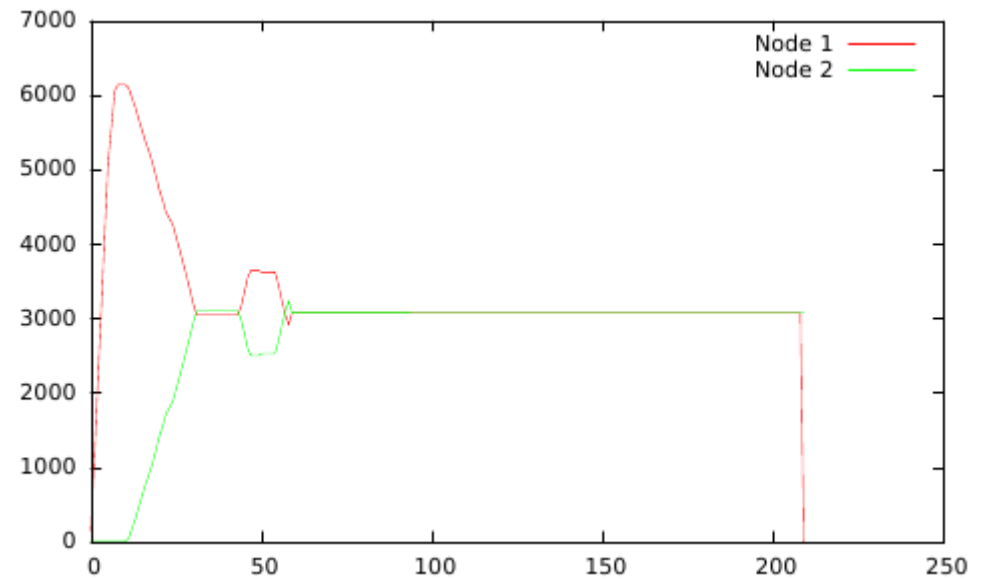
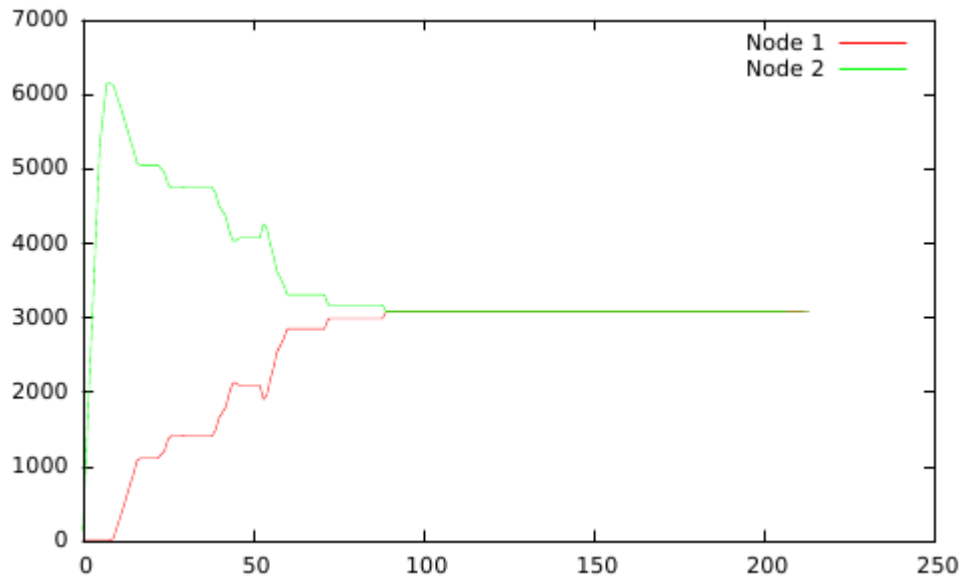
Lower is better stddev	numa01	numa01_THREA...	numa02	numa02_SMT
AutoNUMA25	5.609628627	0.962059423	0.3879003308	0.2093641166
AutoNUMA25MoF	9.5142047312	7.9072624846	0.45563143	0.3450966048

Convergence charts

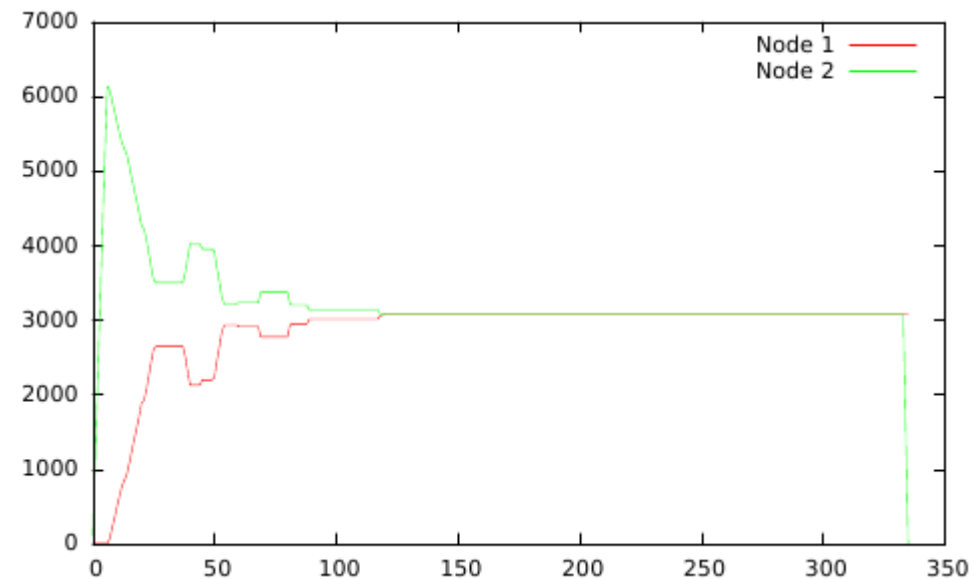
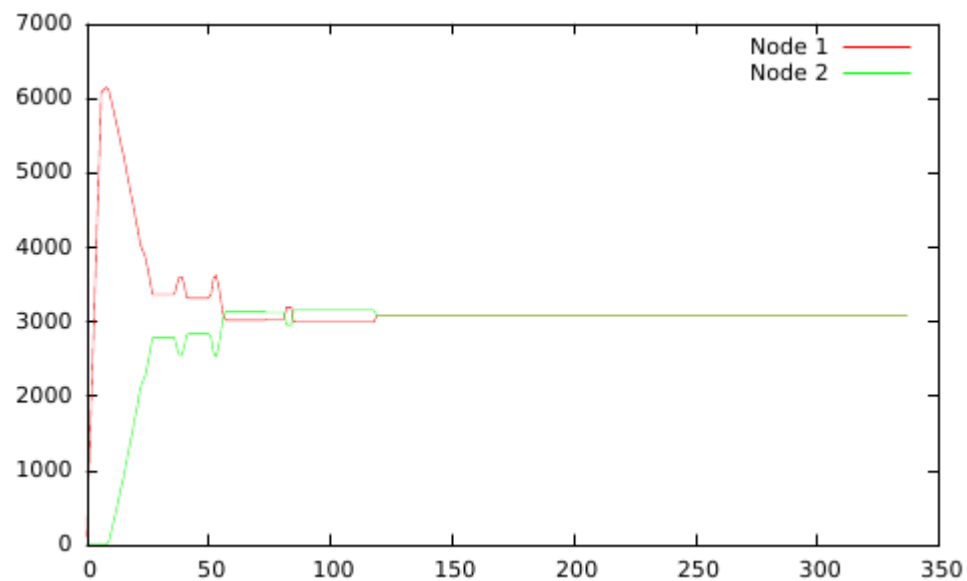
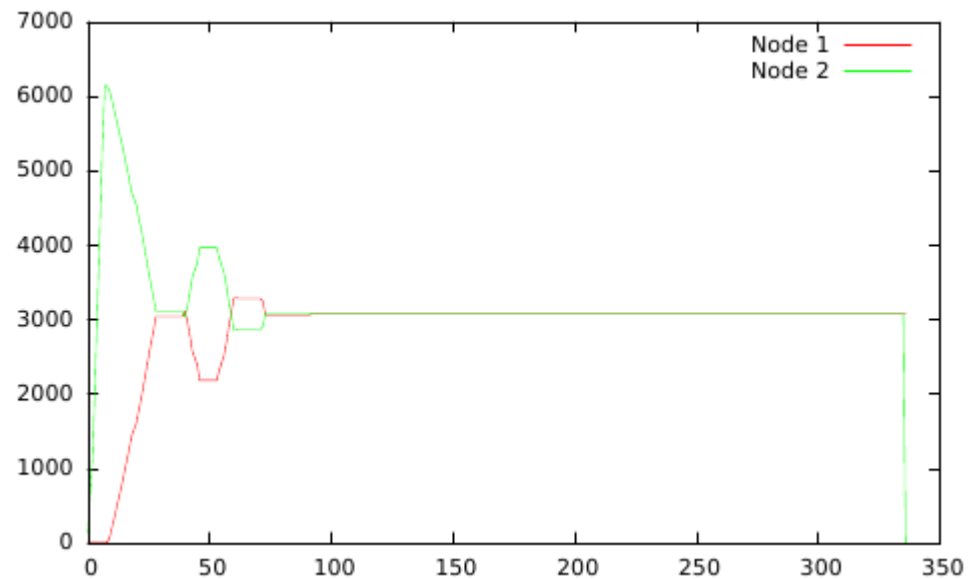
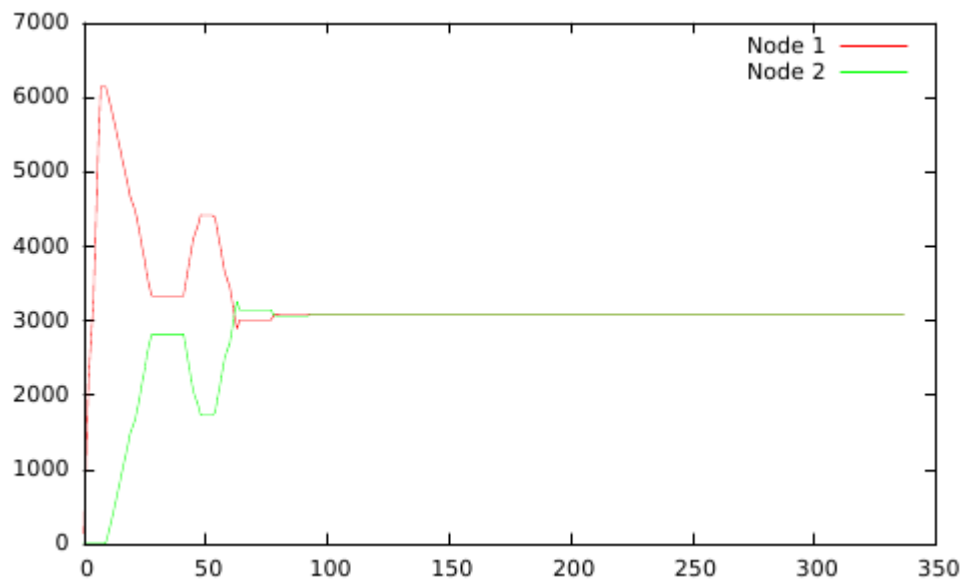
- The AutoNUMA-benchmark produces a chart for each test run (in pdf format).
 - In each chart there is one two dimensional line per NUMA node (node1, node2, etc.).
 - X=time (seconds)
 - Y=memory (MiB)
 - Each line represents how much of the test process's memory is in that NUMA node over time, for the duration of the test. Because all memory starts in one node, it illustrates how the memory migrates over time.
 - A workload converges when the memory levels are equal in all NUMA nodes
 - Note: with `nr_nodes > 2`, `numa01` may not fully converge because half of the cpus will thrash on the memory of half of the nodes, but it should get close enough
 - In the future we plan to add a new `numa01` "PER_NODE" test with $P=nr_nodes$ and $T=nr_cpus/nr_nodes$



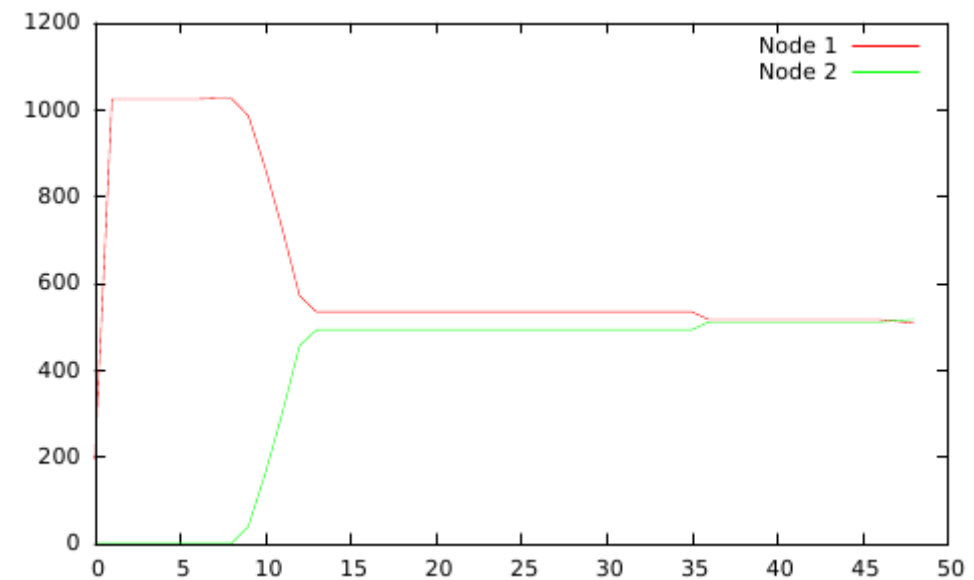
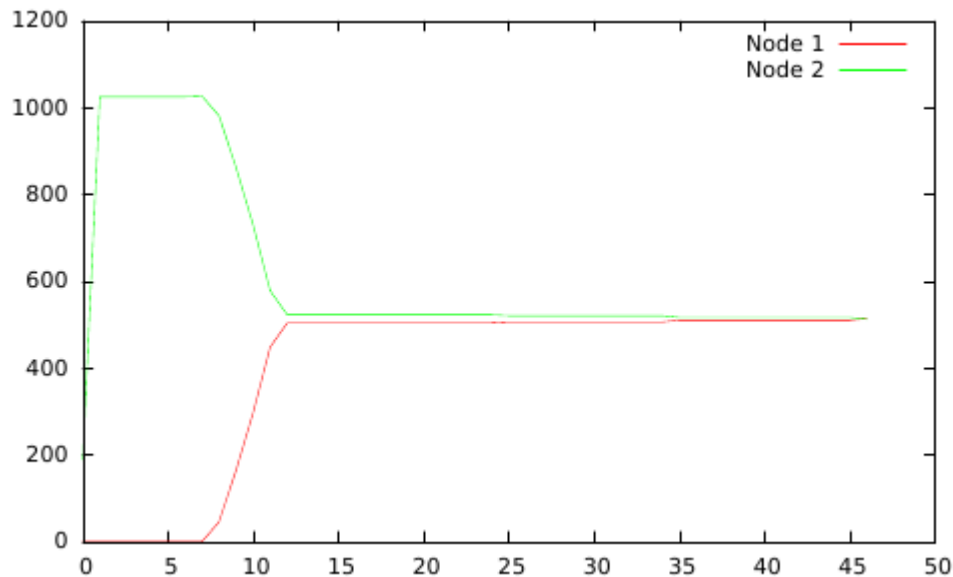
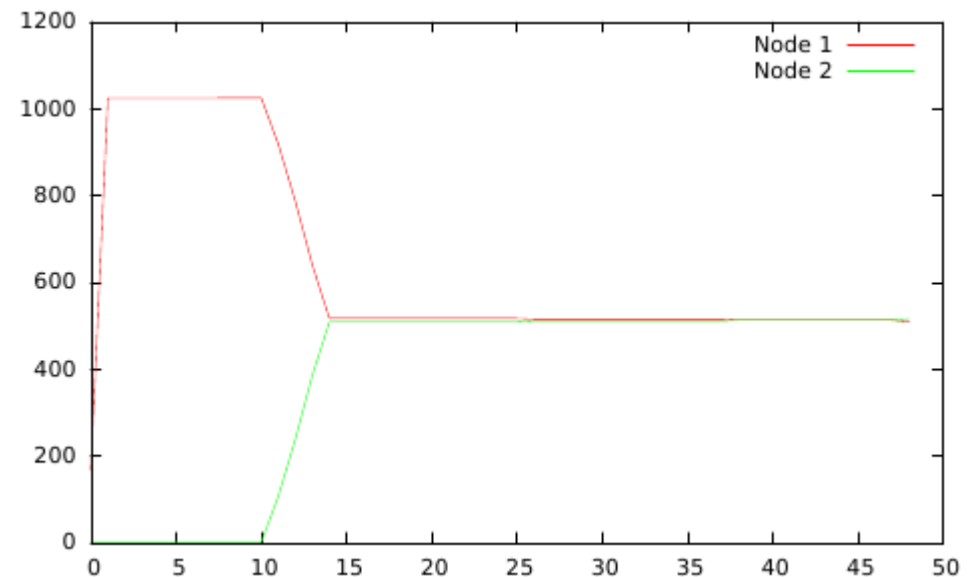
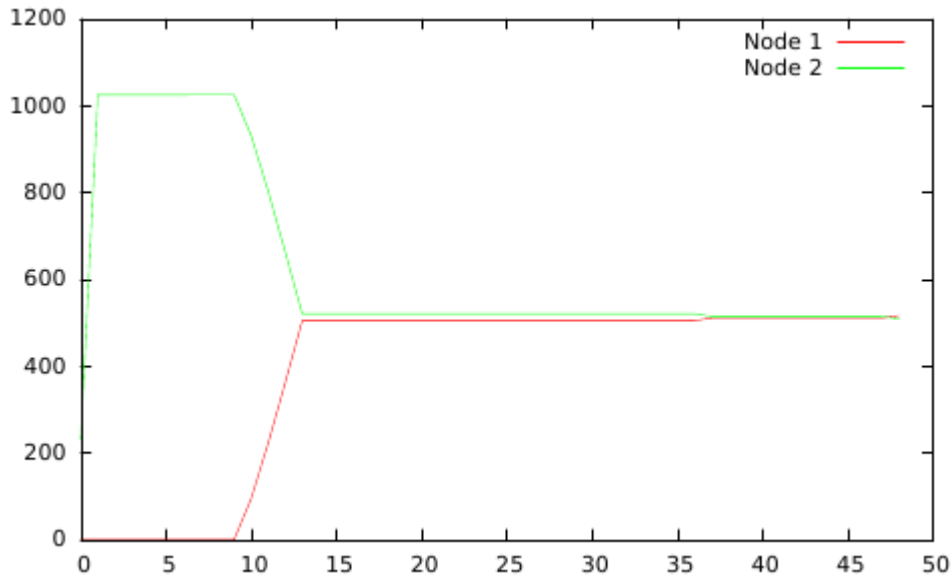
AutoNUMA25 numa01



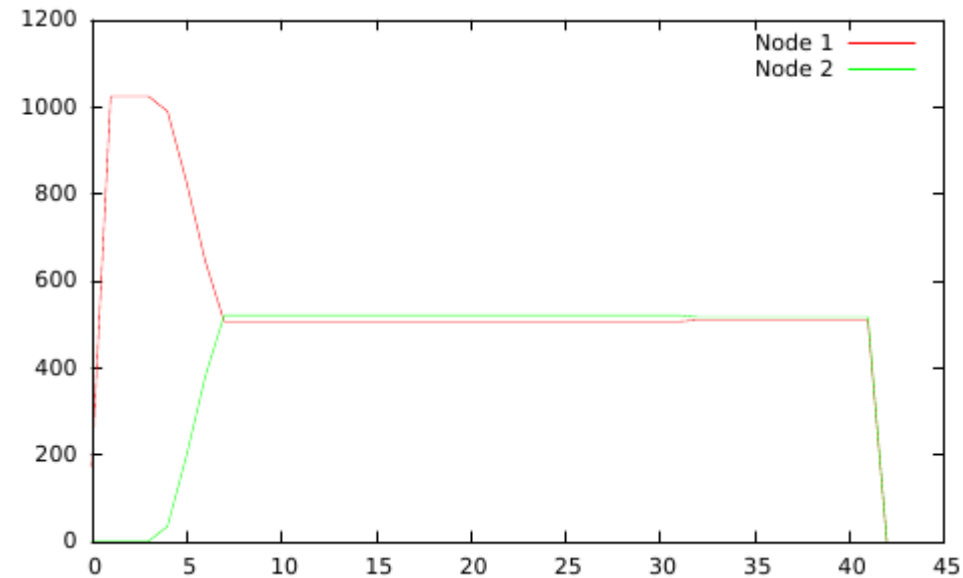
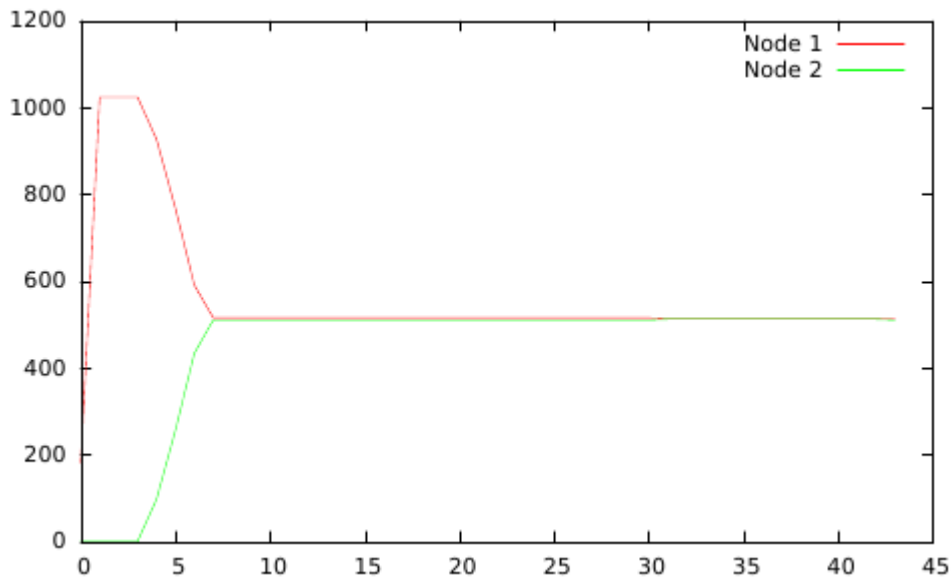
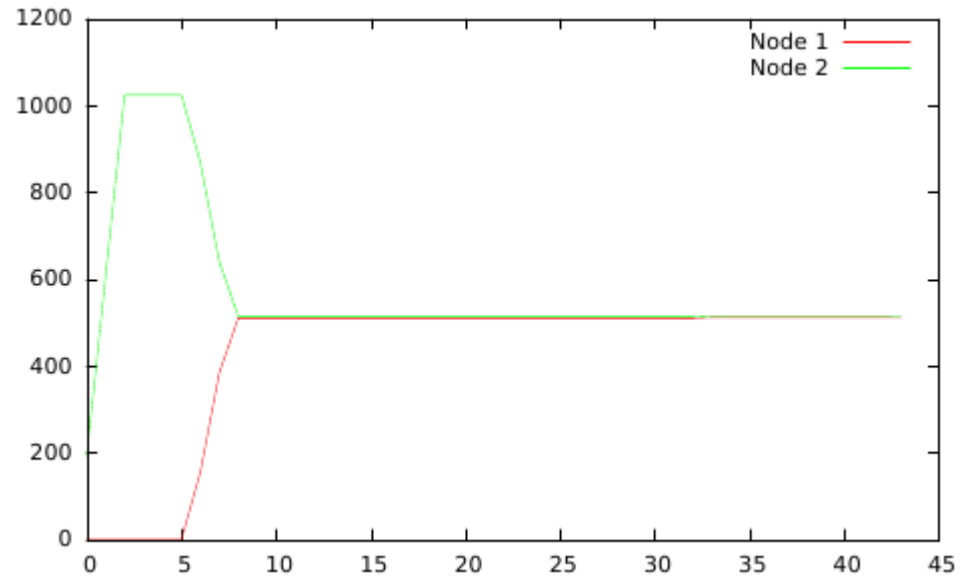
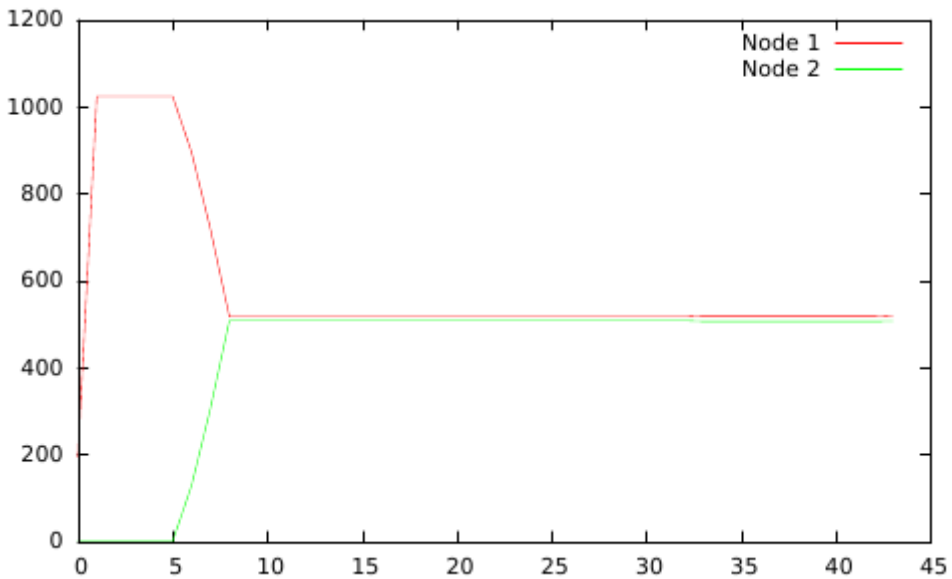
AutoNUMA25 numa01_THREAD..



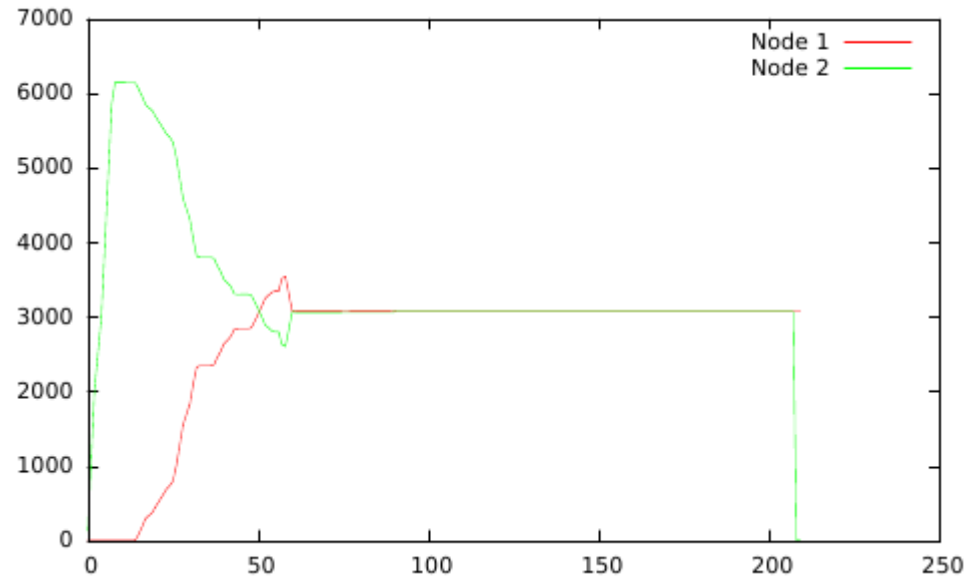
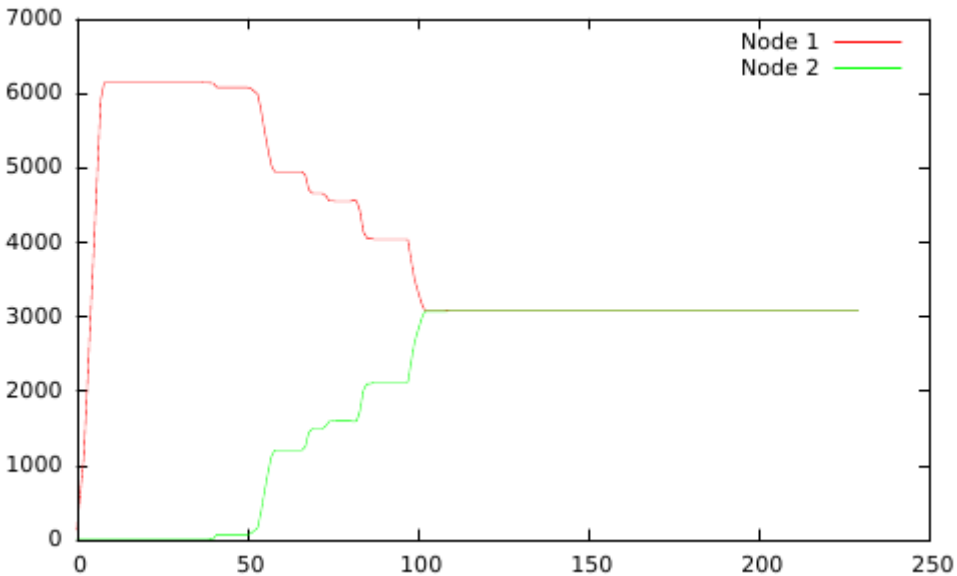
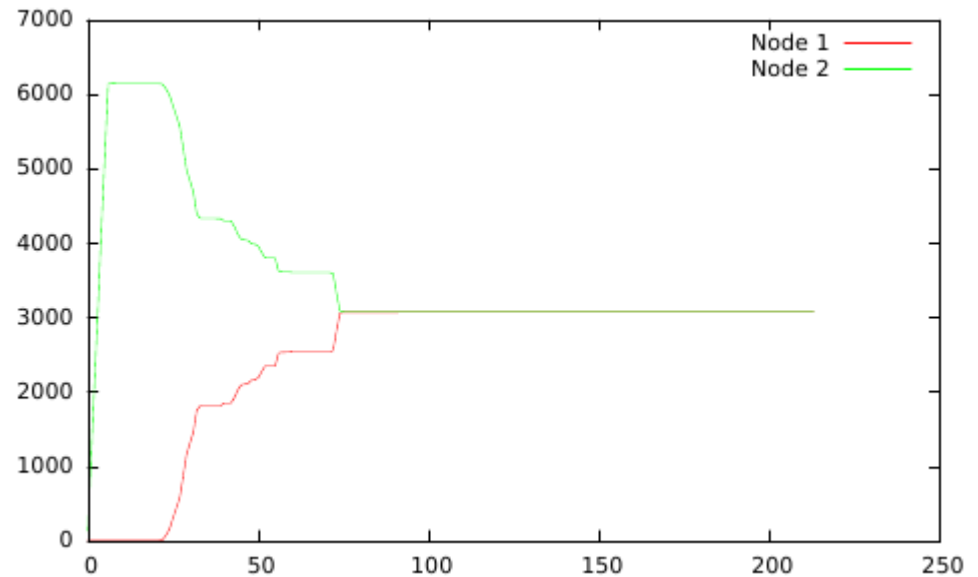
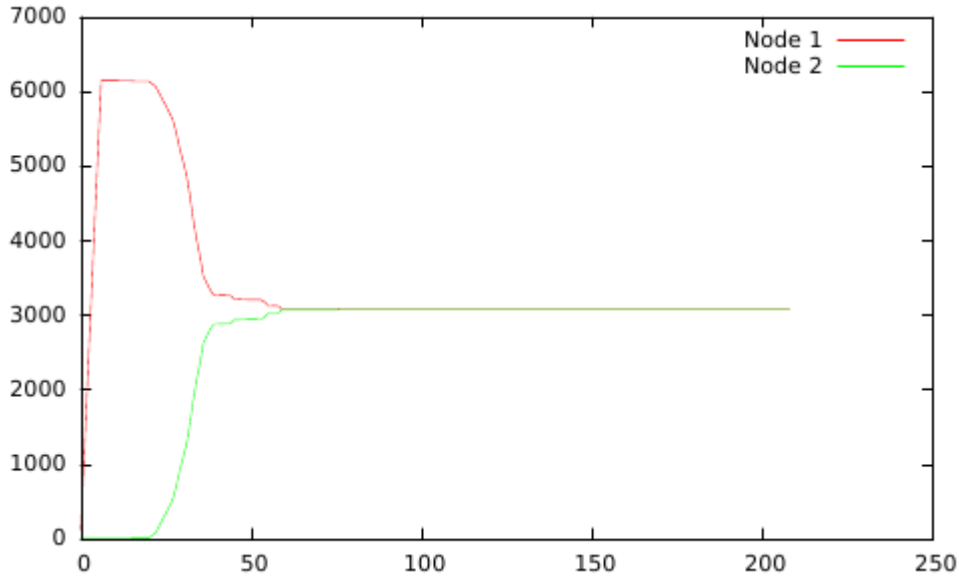
AutoNUMA25 numa02



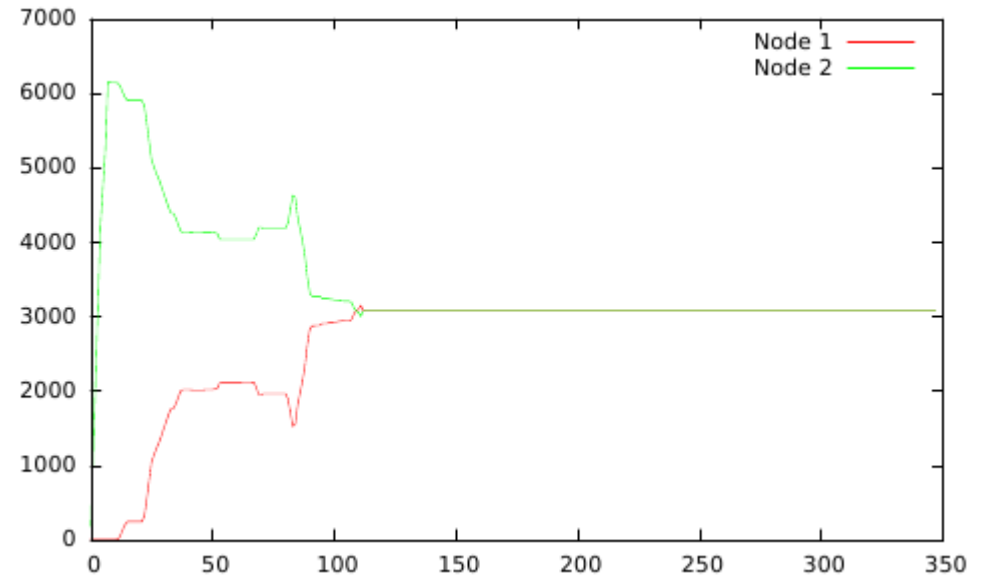
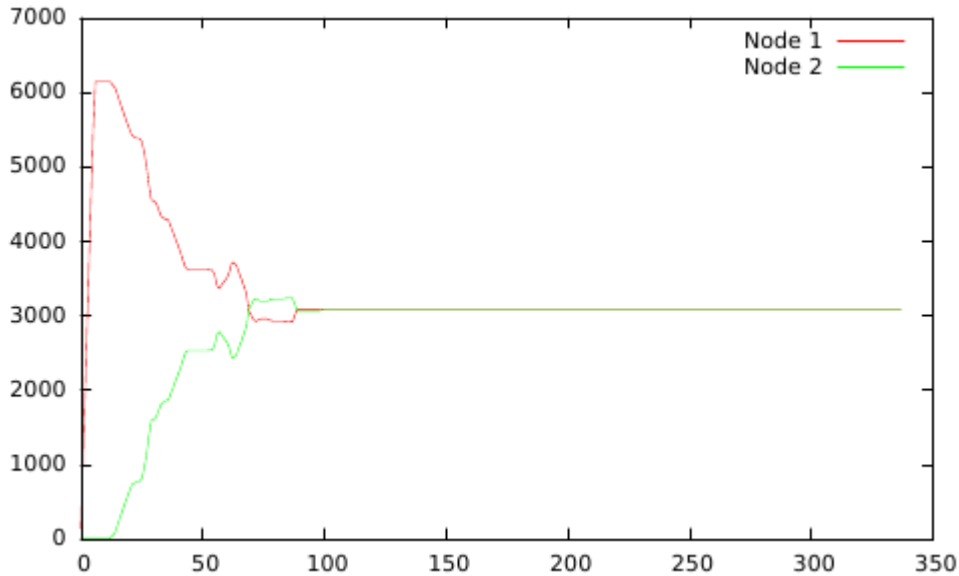
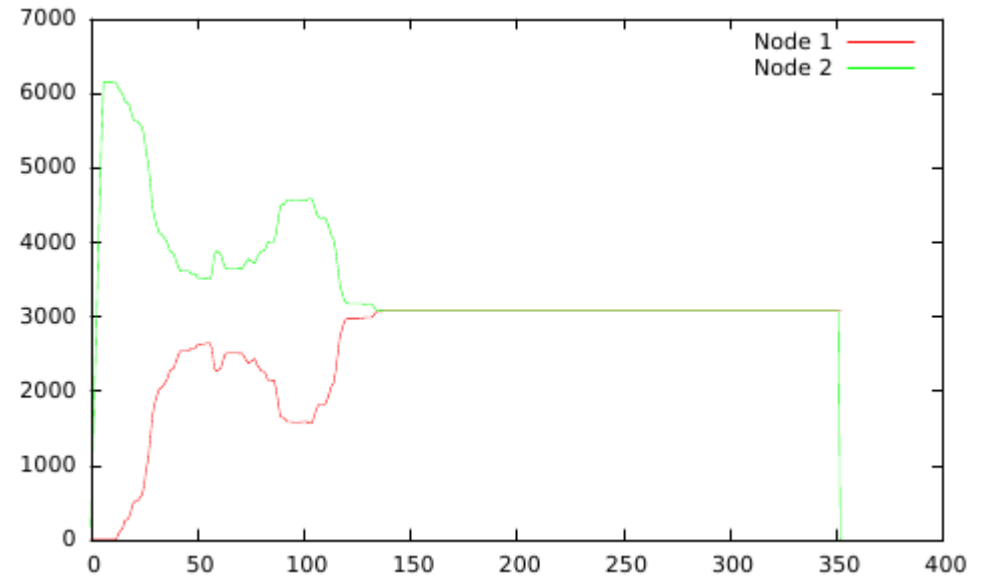
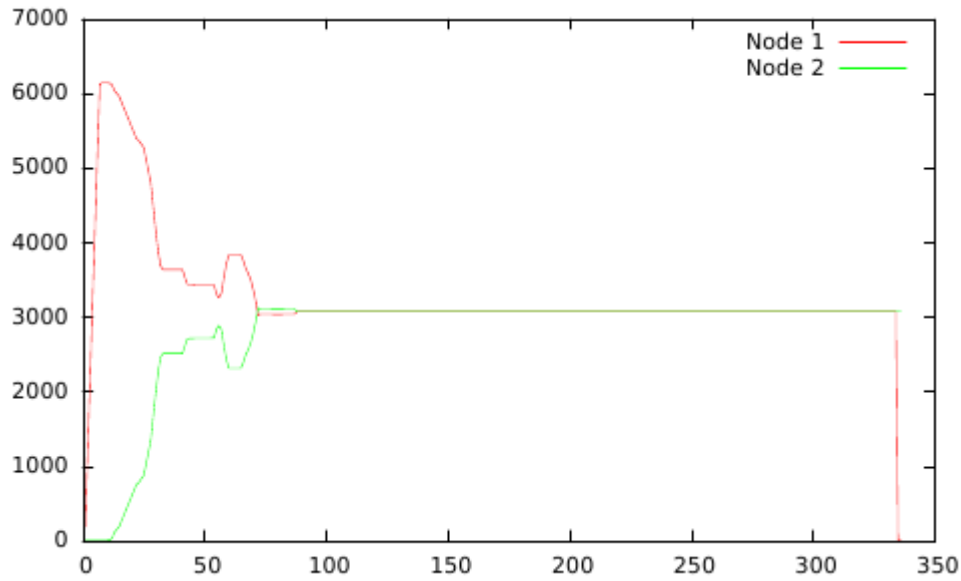
AutoNUMA25 numa02_SMT



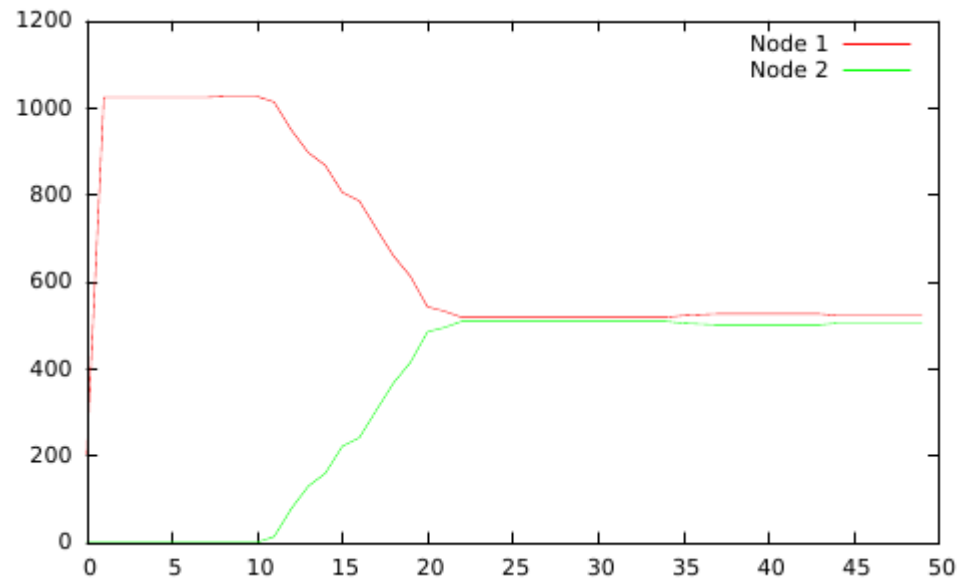
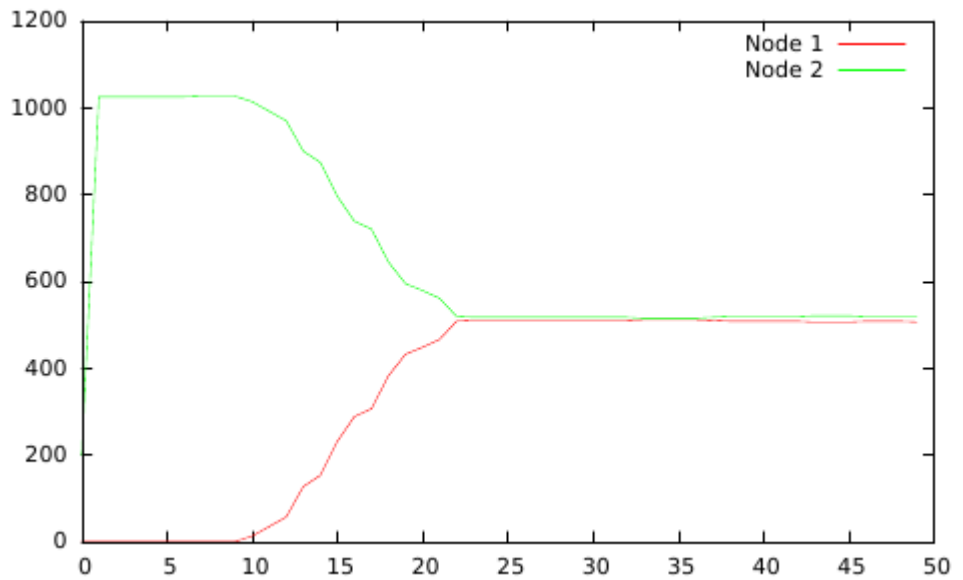
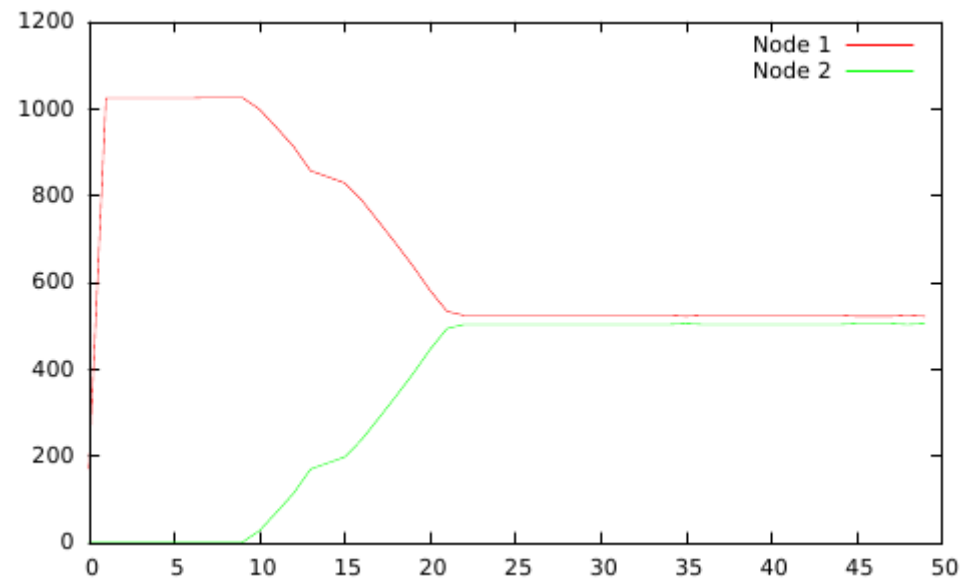
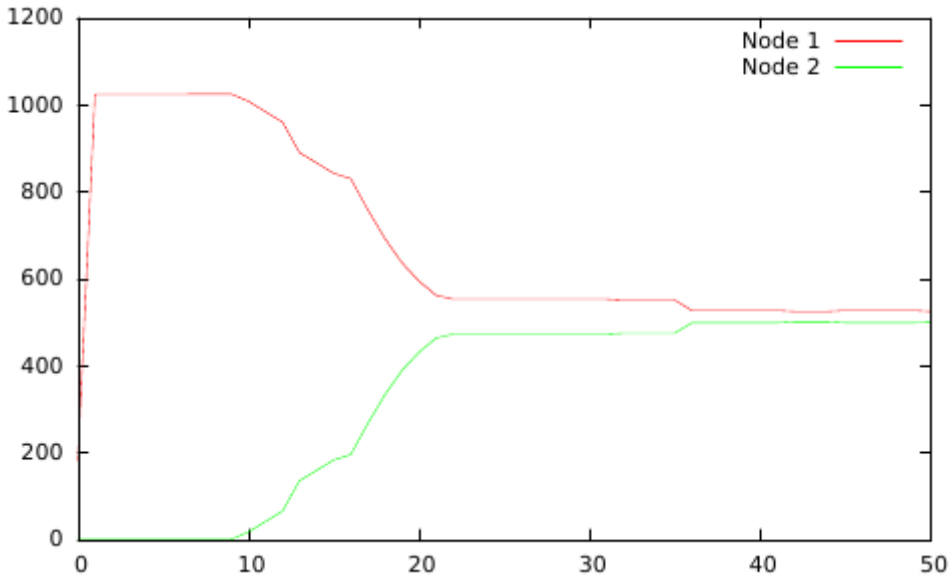
AutoNUMA25 MoF numa01



AutoNUMA25 MoF numa01_TH..



AutoNUMA25 MoF numa02



AutoNUMA25 MoF numa02_SMT

