Applies to **openSUSE Leap 15.1**

10 Automatic Non-Uniform Memory Access (NUMA) Balancing

Abstract

There are physical limitations to hardware that are encountered when many CPUs and lots of memory are required. In this chapter, the important limitation is that there is limited communication bandwidth between the CPUs and the memory. One architecture modification that was introduced to address this is Non-Uniform Memory Access (NUMA).

In this configuration, there are multiple nodes. Each of the nodes contains a subset of all CPUs and memory. The access speed to main memory is determined by the location of the memory relative to the CPU. The performance of a workload depends on the application threads accessing data that is local to the CPU the thread is executing on. Automatic NUMA Balancing is a new feature of SLE 12. Automatic NUMA Balancing migrates data on demand to memory nodes that are local to the CPU accessing that data. Depending on the workload, this can dramatically boost performance when using NUMA hardware.

10.1 Implementation

Automatic NUMA balancing happens in three basic steps:

1. A task scanner periodically scans a portion of a task's address space and marks the memory to force a page fault when the data is next accessed.
2. The next access to the data will result in a NUMA Hinting Fault. Based on this fault, the data can be migrated to a memory node associated with the task accessing the memory.
3. To keep a task, the CPU it is using and the memory it is accessing together, the scheduler groups tasks that share data.

The unmapping of data and page fault handling incurs overhead. However, commonly the overhead will be offset by threads accessing data associated with the CPU.

10.2 Configuration

Static configuration has been the recommended way of tuning workloads on NUMA hardware for some time. To do this, memory policies can be set with **`numactl`**, **`taskset`** or **`cpusets`**. NUMA-aware applications can use special APIs. In cases where the static policies have already been created, automatic NUMA balancing should be disabled as the data access should already be local.

`numactl` `--hardware` will show the memory configuration of the machine and whether it supports NUMA or not. This is example output from a 4-node machine.

```
tux > numactl --hardware
available: 4 nodes (0-3)
node 0 cpus: 0 4 8 12 16 20 24 28 32 36 40 44
node 0 size: 16068 MB
node 0 free: 15909 MB
node 1 cpus: 1 5 9 13 17 21 25 29 33 37 41 45
node 1 size: 16157 MB
node 1 free: 15948 MB
node 2 cpus: 2 6 10 14 18 22 26 30 34 38 42 46
node 2 size: 16157 MB
node 2 free: 15981 MB
node 3 cpus: 3 7 11 15 19 23 27 31 35 39 43 47
node 3 size: 16157 MB
node 3 free: 16028 MB
node distances:
node   0   1   2   3
```

```
  0:  10  20  20  20
  1:  20  10  20  20
  2:  20  20  10  20
  3:  20  20  20  10
```

Automatic NUMA balancing can be enabled or disabled for the current session by writing `1` or `0` to `/proc/sys/kernel/numa_balancing` which will enable or disable the feature respectively. To permanently enable or disable it, use the kernel command line option `numa_balancing=[enable|disable]`.

If Automatic NUMA Balancing is enabled, the task scanner behavior can be configured. The task scanner balances the overhead of Automatic NUMA Balancing with the amount of time it takes to identify the best placement of data.

`numa_balancing_scan_delay_ms`
: The amount of CPU time a thread must consume before its data is scanned. This prevents creating overhead because of short-lived processes.

`numa_balancing_scan_period_min_ms` **and** `numa_balancing_scan_period_max_ms`
: Controls how frequently a task's data is scanned. Depending on the locality of the faults the scan rate will increase or decrease. These settings control the min and max scan rates.

`numa_balancing_scan_size_mb`
: Controls how much address space is scanned when the task scanner is active.

10.3 Monitoring

The most important task is to assign metrics to your workload and measure the performance with Automatic NUMA Balancing enabled and disabled to measure the impact. Profiling tools can be used to monitor local and remote memory accesses if the CPU supports such monitoring. Automatic NUMA Balancing activity can be monitored via the following parameters in `/proc/vmstat`:

`numa_pte_updates`
: The amount of base pages that were marked for NUMA hinting faults.

`numa_huge_pte_updates`

The amount of transparent huge pages that were marked for NUMA hinting faults. In combination with `numa_pte_updates` the total address space that was marked can be calculated.

`numa_hint_faults`
Records how many NUMA hinting faults were trapped.

`numa_hint_faults_local`
Shows how many of the hinting faults were to local nodes. In combination with `numa_hint_faults`, the percentage of local versus remote faults can be calculated. A high percentage of local hinting faults indicates that the workload is closer to being converged.

`numa_pages_migrated`
Records how many pages were migrated because they were misplaced. As migration is a copying operation, it contributes the largest part of the overhead created by NUMA balancing.

10.4 Impact

The following illustrates a simple test case of a 4-node NUMA machine running the SpecJBB 2005 using a single instance of the JVM with no static tuning around memory policies. Note, however, that the impact for each workload will vary and that this example is based on a pre-release version of openSUSE Leap 12.

```
            Balancing disabled      Balancing enabled
TPut 1      26629.00 (  0.00%)     26507.00 ( -0.46%)
TPut 2      55841.00 (  0.00%)     53592.00 ( -4.03%)
TPut 3      86078.00 (  0.00%)     86443.00 (  0.42%)
TPut 4     116764.00 (  0.00%)    113272.00 ( -2.99%)
TPut 5     143916.00 (  0.00%)    141581.00 ( -1.62%)
TPut 6     166854.00 (  0.00%)    166706.00 ( -0.09%)
TPut 7     195992.00 (  0.00%)    192481.00 ( -1.79%)
TPut 8     222045.00 (  0.00%)    227143.00 (  2.30%)
TPut 9     248872.00 (  0.00%)    250123.00 (  0.50%)
TPut 10    270934.00 (  0.00%)    279314.00 (  3.09%)
TPut 11    297217.00 (  0.00%)    301878.00 (  1.57%)
TPut 12    311021.00 (  0.00%)    326048.00 (  4.83%)
TPut 13    324145.00 (  0.00%)    346855.00 (  7.01%)
TPut 14    345973.00 (  0.00%)    378741.00 (  9.47%)
TPut 15    354199.00 (  0.00%)    394268.00 ( 11.31%)
TPut 16    378016.00 (  0.00%)    426782.00 ( 12.90%)
TPut 17    392553.00 (  0.00%)    437772.00 ( 11.52%)
TPut 18    396630.00 (  0.00%)    456715.00 ( 15.15%)
TPut 19    399114.00 (  0.00%)    484020.00 ( 21.27%)
```

```
TPut 20    413907.00 (  0.00%)    493618.00 ( 19.26%)
TPut 21    413173.00 (  0.00%)    510386.00 ( 23.53%)
TPut 22    420256.00 (  0.00%)    521016.00 ( 23.98%)
TPut 23    425581.00 (  0.00%)    536214.00 ( 26.00%)
TPut 24    429052.00 (  0.00%)    532469.00 ( 24.10%)
TPut 25    426127.00 (  0.00%)    526548.00 ( 23.57%)
TPut 26    422428.00 (  0.00%)    531994.00 ( 25.94%)
TPut 27    424378.00 (  0.00%)    488340.00 ( 15.07%)
TPut 28    419338.00 (  0.00%)    543016.00 ( 29.49%)
TPut 29    403347.00 (  0.00%)    529178.00 ( 31.20%)
TPut 30    408681.00 (  0.00%)    510621.00 ( 24.94%)
TPut 31    406496.00 (  0.00%)    499781.00 ( 22.95%)
TPut 32    404931.00 (  0.00%)    502313.00 ( 24.05%)
TPut 33    397353.00 (  0.00%)    522418.00 ( 31.47%)
TPut 34    382271.00 (  0.00%)    491989.00 ( 28.70%)
TPut 35    388965.00 (  0.00%)    493012.00 ( 26.75%)
TPut 36    374702.00 (  0.00%)    502677.00 ( 34.15%)
TPut 37    367578.00 (  0.00%)    500588.00 ( 36.19%)
TPut 38    367121.00 (  0.00%)    496977.00 ( 35.37%)
TPut 39    355956.00 (  0.00%)    489430.00 ( 37.50%)
TPut 40    350855.00 (  0.00%)    487802.00 ( 39.03%)
TPut 41    345001.00 (  0.00%)    468021.00 ( 35.66%)
TPut 42    336177.00 (  0.00%)    462260.00 ( 37.50%)
TPut 43    329169.00 (  0.00%)    467906.00 ( 42.15%)
TPut 44    329475.00 (  0.00%)    470784.00 ( 42.89%)
TPut 45    323845.00 (  0.00%)    450739.00 ( 39.18%)
TPut 46    323878.00 (  0.00%)    435457.00 ( 34.45%)
TPut 47    310524.00 (  0.00%)    403914.00 ( 30.07%)
TPut 48    311843.00 (  0.00%)    459017.00 ( 47.19%)

                        Balancing Disabled        Balancing Enabled
 Expctd Warehouse          48.00 (  0.00%)          48.00 (  0.00%)
 Expctd Peak Bops      310524.00 (  0.00%)      403914.00 ( 30.07%)
 Actual Warehouse          25.00 (  0.00%)          29.00 ( 16.00%)
 Actual Peak Bops      429052.00 (  0.00%)      543016.00 ( 26.56%)
 SpecJBB Bops            6364.00 (  0.00%)        9368.00 ( 47.20%)
 SpecJBB Bops/JVM        6364.00 (  0.00%)        9368.00 ( 47.20%)
```

Automatic NUMA Balancing simplifies tuning workloads for high performance on NUMA machines. Where possible, it is still recommended to statically tune the workload to partition it within each node. However, in all other cases, automatic NUMA balancing should boost performance.