Subject: [RFC][for -mm] memory controller enhancements for reclaiming take2 [0/8] introduction
Posted by KAMEZAWA Hiroyuki on Mon, 03 Dec 2007 09:33:55 GMT
View Forum Message <> Reply to Message

Hi, here is my memory cgroup patchset for 2.6.24-rc3-mm2.
(for review again. sorry. added -mm to the CC:)

Patch contents are
- clean up
- (possible) race fix.
- throttling LRU scan.
- background reclaim and high/low watermark

If it's better to divide aboves into a few sets, I'll do so.

Tested on ia64/8CPU/NUMA and x86_64/2CPU/SMP.

[Patches]
[1/8] ... remove unused variable (clean up)
[2/8] ... change 'if' sentense to BUG_ON() (clean up)
[3/8] ... add free_mem_cgroup_per_zone_info (clean up)
[4/8] ... possible race fix in resource controler
[5/8] ... throttling simultaneous direct reclaim under cgroup
[6/8] ... high/low watermark for resource controler
[7/8] ... use high/low watermark in memory controler
[8/8] ... wake up reclaim waiters at unchage.

TODO(1): Should I care resource controler users who doesn't use watermarks ?
and how should I do ?

==

here is a brief kernbench result on ia64/8CPU/2-node NUMA.
(average of 3 runs)

Used 800M limitation and High/Low watermark is 790M/760M (for patched kernel.)
make -j 4 and make -j 32.

*** Before patch ***********

Average Half Load -j 4 Run:	Average Optimal -j 32 Load Run:
Elapsed Time 266.104	Elapsed Time 353.957
User Time 1015.82	User Time 1070.68
System Time 70.0701	System Time 154.63
Percent CPU 407.667	Percent CPU 351.667
Context Switches 136916	Context Switches 223173
Sleeps 135404	Sleeps 199366

*** After patch ***********
Average Half Load -j 4 Run:
Elapsed Time 266.96
User Time 1016.55
System Time 70.24
Percent CPU 406.666
Context Switches 133219
Sleeps 137232

Average Optimal -j 32 Load Run:
Elapsed Time 232.32 ---(*1)
User Time 1120.9
System Time 116.843 ---(*2)
Percent CPU 537.667 ---(*3)
Context Switches 246752
Sleeps 195215

make -j 4 result has no difference between "before" and "after".
(800M was enough)
make -j 32 result has a difference.

(*1) Elapsed Time is decreased.
(*2) System Time is decreased.
(*3) CPU Utilization is increased.

This is because %iowait is decreased and "recaliming too much" is avoided.

here is vmstat -n 60 result amount make -j 32.

*** Before Patch ***

```
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in    cs    us sy id wa st
37  6 416176 5758384  47824 288656   25 15823   311 18408 37259 2935  31  5 39 25  0
10 28 590336 5774528  50112 294560  323 27815   950 28225 52751 3405  45  8  4 43  0
18 28 648384 5614352  53744 298464  507 27710  1148 28226 54039 3521  60  9  2 29  0
 2 38 921312 5269648  50256 285392  185 29883   995 30471 55296 3000  39  7  8 46  0
 4  9 974944 5710672  52288 309104  206 31675   950 31885 54836 3074  48  6  6 40  0
 4 33 919568 5553984  49520 285664  119 3775   891  6850 17000 2810   21  4 63 12  0
15 24 1063856 5290144  51904 294128   56 27886   952 28406 56501 3317 54  9  1 36  0
 0 44 1399696 4933296  54592 292416  260 28046   847 28361 58778 3104 43  7  5 46  0
 0 47 1351200 4992208  56400 302896  275 22439   816 22694 44746 2945 32  5 14 50  0
 2 40 1433568 4878784  58032 290768  189 19617   971 19907 33764 2883 20  4 17 59  0
 3 41 1600608 4716688  59168 304496  155 19675   598 20032 39205 3007 39  4 14 43  0
 0  0 1143728 5587440  81456 301792  373 4040  1095  7052 11288 3006  30  3 50 17  0
 0 56 1487536 5281472  52208 276624   52 21253   403 21568 42882 2793 31  6 18 45  0
.....
```

*** After Patch ***

```
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in    cs     us sy id wa st
25 22 271872 5869728  51792 290656   14 12589   593 13659 68145 3692   68  8 20  3  0
30 25 546592 5621552  53216 288128  106 25837  1058 26541 98549 3377   70  9  8 12  0
 1 16 878384 5103024  54064 290032  345 25477  1251 26176 94491 3325   67  8  8 18  0
 1  0 767200 5918896  82032 302704   25 8035  1023 10953 32435 2859    30  4 56 10  0
28 10 948640 5156912  53920 290416   21 13784   814 14706 77824 3740   84  9  5  3  0
```

 0 15 1320224 4683840 55680 296912 158 28465   893 29008 98158 3303 65 8 8 19 0
40 15 1619600 4372128 55424 297424 136 31633   848 32190 102808 3188 60 7 6 27 0
 0 0 1529152 5203728 83584 299968   47 11217   950 14145 39845 2799 27 4 52 17 0
30 10 1821776 4272416 56384 292528   16 13483   831 14271 77710 3749 77 9 11 3 0
33 16 2016464 3998480 56448 296416   96 19846   739 20454 78992 3342 58 6 11 26 0
 0 17 2148832 3820112 56544 289920   68 19842   528 20350 75083 2966 45 5 23 26 0
30 17 2118320 3891520 56320 293776   44 17544   533 18000 61345 2613 36 4 23 36 0

%iowait is decreased to some extent.
%user is increased (as a result)

No meaningful difference when memory is enough.

Thanks,
-Kame

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers