Subject: RSS controller v2 Test results (lmbench)
Posted by Balbir Singh on Thu, 17 May 2007 17:50:12 GMT
View Forum Message <> Reply to Message

Hi, Pavel/Andrew,

I've run lmbench on RSS controller v2 with the following patches
applied

rss-fix-free-of-active-pages.patch
rss-fix-nodescan.patch
rss-implement-per-container-page-referenced.patch
rss-fix-lru-race

(NOTE: all of these were posted on lkml)

I've used three configurations for testing

1. Container mounted with the RSS controller and the tests started within
   a container whose RSS is limited to 256 MB
2. Counter mounted, but no limit set
3. Counter not mounted

(1) is represented by cont256, (2) by contmnt and (3) by contnomnt respectively
in the results.

L M B E N C H 2 . 0 S U M M A R Y

Basic system parameters

Host OS Description Mhz

--- --- --- ----
cont256 Linux 2.6.20- x86_64-linux-gnu 1993
contmnt Linux 2.6.20- x86_64-linux-gnu 1993
contnomnt Linux 2.6.20- x86_64-linux-gnu 1993

Processor, Processes - times in microseconds - smaller is better
---
Host OS Mhz null null open selct sig sig fork exec sh
call I/O stat clos TCP inst hndl proc proc proc
--- --- --- --- --- --- --- --- --- --- --- --- ---
cont256 Linux 2.6.20- 1993 0.08 0.33 4.31 5.93 9.910 0.23 1.59 152. 559. 5833
contmnt Linux 2.6.20- 1993 0.08 0.35 3.25 5.80 6.422 0.23 1.53 161. 562. 5937
contnomnt Linux 2.6.20- 1993 0.08 0.29 3.18 5.14 11.3 0.23 1.37 159. 570. 5973

Context switching - times in microseconds - smaller is better

Host	OS	2p/0K ctxsw	2p/16K ctxsw	2p/64K ctxsw	8p/16K ctxsw	8p/64K ctxsw	16p/16K ctxsw	16p/64K ctxsw
cont256	Linux 2.6.20-	1.760	1.9800	6.6600	3.0100	6.5500	3.12000	6.84000
contmnt	Linux 2.6.20-	1.950	1.9900	6.2900	3.6400	6.6800	3.59000	14.8
contnomnt	Linux 2.6.20-	1.420	2.5100	6.6400	3.7600	6.5300	3.34000	21.5

Local Communication latencies in microseconds - smaller is better

Host	OS	2p/0K ctxsw	Pipe	AF UNIX	UDP	RPC/ UDP	TCP	RPC/ TCP conn
cont256	Linux 2.6.20-	1.760	18.9	46.5	19.2	22.9	23.0	28.0 40.0
contmnt	Linux 2.6.20-	1.950	20.0	44.6	19.2	20.1	37.9	25.2 42.6
contnomnt	Linux 2.6.20-	1.420	23.2	38.5	19.2	23.2	24.4	28.9 54.3

File & VM system latencies in microseconds - smaller is better

Host	OS	0K File Create	0K File Delete	10K File Create	10K File Delete	Mmap Latency	Prot Fault	Page Fault
cont256	Linux 2.6.20-	17.6	15.4	62.8	29.4	1010.0	0.401	3.00000
contmnt	Linux 2.6.20-	20.7	16.4	68.1	31.9	3886.0	0.495	3.00000
contnomnt	Linux 2.6.20-	21.1	16.8	69.3	31.6	4383.0	0.443	2.00000

Local Communication bandwidths in MB/s - bigger is better

Host	OS	Pipe UNIX	AF	TCP reread	File reread	Mmap (libc)	Bcopy (hand)	Bcopy read	Mem write	Mem
cont256	Linux 2.6.20-	382.	802.	869.	1259.5	1757.8	1184.8	898.4	1875	1497.
contmnt	Linux 2.6.20-	307.	850.	810.	1236.2	1758.8	1173.2	890.9	2636	1469.
contnomnt	Linux 2.6.20-	403.	980.	875.	1236.8	2531.7	912.0	1141.7	2636	1229.

Memory latencies in nanoseconds - smaller is better
(WARNING - may not be correct, check graphs)

Host	OS	Mhz	L1 $	L2 $	Main mem	Guesses
cont256	Linux 2.6.20-	1993	1.506	6.0260	63.8	
contmnt	Linux 2.6.20-	1993	1.506	6.0380	64.0	
contnomnt	Linux 2.6.20-	1993	1.506	6.9410	97.4	

Quick interpretation of results

1. contmnt and cont256 are comparable in performance
2. contnomnt showed degraded performance compared to contmnt

A meaningful container size does not hamper performance. I am in the process of getting more results (with varying container sizes). Please let me know what you think of the results? Would you like to see different benchmarks/ tests/configuration results?

Any feedback, suggestions to move this work forward towards identifying and correcting bottlenecks or to help improve it is highly appreciated.

--
Warm Regards,
Balbir Singh
Linux Technology Center
IBM, ISTL