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	AF UNIX	TCP	File reread	Mmap reread	Bcopy (libc)	Bcopy (hand)	Mem read	Mem write
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

Subject: Re: RSS controller v2 Test results (lmbench)
Posted by Andrew Morton on Thu, 17 May 2007 18:23:57 GMT
View Forum Message <> Reply to Message

On Thu, 17 May 2007 23:20:12 +0530
Balbir Singh <balbir@linux.vnet.ibm.com> wrote:

> 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.

<wakes up>

Memory reclaim tends not to consume much CPU. Because in steady state it
tends to be the case that the memory reclaim rate (and hopefully the
scanning rate) is equal to the disk IO rate.

Often the most successful way to identify performance problems in there is
by careful code inspection followed by development of exploits.

Is this RSS controller built on Paul's stuff, or is it standalone?

Where do we stand on all of this now anyway? I was thinking of getting Paul's changes into -mm soon, see what sort of calamities that brings about.

Subject: Re: RSS controller v2 Test results (lmbench)
Posted by Rik van Riel on Fri, 18 May 2007 02:55:21 GMT
View Forum Message <> Reply to Message

Balbir Singh wrote:

> 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?

AIM7 results might be interesting, especially when run to crossover.

OTOH, AIM7 can make the current VM explode spectacularly :)

I saw it swap out 1.4GB of memory in one run, on my 2GB memory test system. That's right, it swapped out almost 75% of memory.

Presumably all the AIM7 processes got stuck in the pageout code simultaneously and all decided they needed to swap some pages out. However, the shell got stuck too so I could not get sysrq output on time.

I am trying out a little VM patch to fix that now, carefully watching vmstat output. Should be fun...

--
Politics is the struggle between those who want to make their country the best in the world, and those who believe it already is. Each group calls the other unpatriotic.

Subject: Re: RSS controller v2 Test results (lmbench)
Posted by Balbir Singh on Fri, 18 May 2007 03:46:46 GMT
View Forum Message <> Reply to Message

Andrew Morton wrote:
> On Thu, 17 May 2007 23:20:12 +0530
> Balbir Singh <balbir@linux.vnet.ibm.com> wrote:
>
>> 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.
>
> <wakes up>
>
> Memory reclaim tends not to consume much CPU. Because in steady state it
> tends to be the case that the memory reclaim rate (and hopefully the
> scanning rate) is equal to the disk IO rate.
>

With the memory controller, I suspect memory reclaim will become a function
of the memory the container tries to touch that lies outside its limit.
If a container requires 512 MB of memory and we configure the container
size as 256 MB, then we might see aggressive memory reclaim. We do provide
some statistics to help the user figure out if the reclaim is aggressive,
we'll try and add more statistics.

> Often the most successful way to identify performance problems in there is
> by careful code inspection followed by development of exploits.
>

> Is this RSS controller built on Paul's stuff, or is it standalone?
>

It's built on top of the containers infrastructure. Version 2 was posted
on top of containers v8.

> Where do we stand on all of this now anyway? I was thinking of getting Paul's
> changes into -mm soon, see what sort of calamities that brings about.
>

The RSS controller was posted by Pavel based on some initial patches by me,
so we are in agreement w.r.t approach to memory control. Vaidy is working
on a page cache controller, we are able to use the existing RSS infrastructure
for writing the page cache controller (unmapped). All the stake holders are
on cc, I would request them to speak out on the issues and help build a way
to take this forward.

I've been reviewing and testing Paul's containers v9 patches. As and when
I find more issues, I plan to send out fixes. It'll be good to have the
containers infrastructure in -mm, so that we can start posting controllers
against them for review and acceptance.

--
Warm Regards,
Balbir Singh

Subject: Re: RSS controller v2 Test results (lmbench)
Posted by Balbir Singh on Fri, 18 May 2007 04:07:22 GMT
View Forum Message <> Reply to Message

Rik van Riel wrote:
> Balbir Singh wrote:
>
>> 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?
>
> AIM7 results might be interesting, especially when run to crossover.
>

I'll try and get hold of AIM7, I have some AIM9 results (please
see the attachment, since the results overflow 80 columns, I've
attached them).

> OTOH, AIM7 can make the current VM explode spectacularly :)
>
> I saw it swap out 1.4GB of memory in one run, on my 2GB memory test
> system. That's right, it swapped out almost 75% of memory.
>

This would make a good test case for the RSS and the unmapped page
cache controller. Thanks for bringing it to my attention.

> Presumably all the AIM7 processes got stuck in the pageout code
> simultaneously and all decided they needed to swap some pages out.
> However, the shell got stuck too so I could not get sysrq output
> on time.
>

oops! I wonder if AIM7 creates too many processes and exhausts all
memory. I've seen a case where during an upgrade of my tetex on my
laptop, the setup process failed and continued to fork processes
filling up 4GB of swap.

> I am trying out a little VM patch to fix that now, carefully watching
> vmstat output. Should be fun...

>

VM debugging is always fun!

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

Test Number	Test Name	Elapsed Time (sec)	Iteration Count	Iteration Rate (loops/sec)	Operation Rate (ops/sec)	
1	creat-clo	60.00	8885	148.08333	148083.33	File Creations and Closes/second (256 MB container)
1	creat-clo	60.01	8547	142.42626	142426.26	File Creations and Closes/second (unlimited container)
1	creat-clo	60.01	8632	143.84269	143842.69	File Creations and Closes/second (container not mounted)
2	page_test	60.00	6068	101.13333	171926.67	System Allocations & Pages/second (256 MB container)
2	page_test	60.00	5275	87.91667	149458.33	System Allocations & Pages/second (unlimited container)
2	page_test	60.01	5411	90.16831	153286.12	System Allocations & Pages/second (container not mounted)
3	brk_test	60.01	9151	152.49125	2592351.27	System Memory Allocations/second (256 MB container)
3	brk_test	60.02	7404	123.35888	2097100.97	System Memory Allocations/second (unlimited container)
3	brk_test	60.01	8294	138.21030	2349575.07	System Memory Allocations/second (container not mounted)
4	jmp_test	60.01	983062	16381.63639	16381636.39	Non-local gotos/second (256 MB container)
4	jmp_test	60.00	983084	16384.73333	16384733.33	Non-local gotos/second (unlimited container)
4	jmp_test	60.00	982904	16381.73333	16381733.33	Non-local gotos/second (container not mounted)
5	signal_test	60.01	28013	466.80553	466805.53	Signal Traps/second (256 MB container)
5	signal_test	60.00	28360	472.66667	472666.67	Signal Traps/second (unlimited container)
5	signal_test	60.01	28593	476.47059	476470.59	Signal Traps/second (container not mounted)
6	exec_test	60.02	2596	43.25225	216.26	Program Loads/second (256 MB container)
6	exec_test	60.02	2539	42.30257	211.51	Program Loads/second (unlimited container)

6 exec_test 60.01 2536 42.25962 211.30 Program Loads/second (container not mounted)
7 fork_test 60.01 2118 35.29412 3529.41 Task Creations/second (256 MB container)
7 fork_test 60.03 2130 35.48226 3548.23 Task Creations/second (unlimited container)
7 fork_test 60.01 2130 35.49408 3549.41 Task Creations/second (container not mounted)
8 link_test 60.02 47760 795.73476 50131.29 Link/Unlink Pairs/second (256 MB container)
8 link_test 60.02 48156 802.33256 50546.95 Link/Unlink Pairs/second (unlimited container)
8 link_test 60.00 49778 829.63333 52266.90 Link/Unlink Pairs/second (container not mounted)
--- ---

Subject: Re: RSS controller v2 Test results (lmbench)
Posted by Lee Schermerhorn on Mon, 21 May 2007 13:53:34 GMT
View Forum Message <> Reply to Message

On Fri, 2007-05-18 at 09:37 +0530, Balbir Singh wrote:
> Rik van Riel wrote:
> > Balbir Singh wrote:
> >
> >> 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?
> >
> > AIM7 results might be interesting, especially when run to crossover.
> >
>
> I'll try and get hold of AIM7, I have some AIM9 results (please
> see the attachment, since the results overflow 80 columns, I've
> attached them).
>
> > OTOH, AIM7 can make the current VM explode spectacularly :)
> >
> > I saw it swap out 1.4GB of memory in one run, on my 2GB memory test
> > system. That's right, it swapped out almost 75% of memory.
> >
>
> This would make a good test case for the RSS and the unmapped page
> cache controller. Thanks for bringing it to my attention.

>
> > Presumably all the AIM7 processes got stuck in the pageout code
> > simultaneously and all decided they needed to swap some pages out.
> > However, the shell got stuck too so I could not get sysrq output
> > on time.
> >
>
> oops! I wonder if AIM7 creates too many processes and exhausts all
> memory. I've seen a case where during an upgrade of my tetex on my
> laptop, the setup process failed and continued to fork processes
> filling up 4GB of swap.

Jumping in late, I just want to note that in our investigations, when AIM7 gets into this situation [non-responsive system], it's because all cpus are in reclaim, spinning on an anon_vma spin lock. AIM7 forks [10s of] thousands of children from a single parent, resultings in thousands of vmas on the anon_vma list. shrink_inactive_list() must walk this list twice [page_referenced() and try_to_unmap()] under spin_lock for each anon page.

[Aside: Just last week, I encountered a similar situation on the i_mmap_lock for page cache pages running a 1200 user Oracle/OLTP run on a largish ia64 system. Left the system spitting out "soft lockup" messages/stack dumps overnight. Still spitting the next day, so I decided to reboot.]

I have a patch that turns the anon_vma lock into a reader/writer lock that alleviates the problem somewhat, but with 10s of thousands of vmas on the lists, system still can't swap enough memory fast enough to recover.

We've run some AIM7 tests with Rik's "split lru list" patch, both with and without the anon_vma reader/writer lock patch. We'll be posting results later this week. Quick summary: with Rik's patch, AIM performance tanks earlier, as the system starts swapping earlier. However, system remains responsive to shell input. More into to follow.

>
> > I am trying out a little VM patch to fix that now, carefully watching
> > vmstat output. Should be fun...
> >
>
> VM debugging is always fun!

For some definition thereof...

Lee

Subject: Re: RSS controller v2 Test results (lmbench)
Posted by William Lee Irwin III on Mon, 21 May 2007 14:59:17 GMT
View Forum Message <> Reply to Message

On Fri, 2007-05-18 at 09:37 +0530, Balbir Singh wrote:
>> oops! I wonder if AIM7 creates too many processes and exhausts all
>> memory. I've seen a case where during an upgrade of my tetex on my
>> laptop, the setup process failed and continued to fork processes
>> filling up 4GB of swap.

On Mon, May 21, 2007 at 09:53:34AM -0400, Lee Schermerhorn wrote:
> Jumping in late, I just want to note that in our investigations, when
> AIM7 gets into this situation [non-responsive system], it's because all
> cpus are in reclaim, spinning on an anon_vma spin lock. AIM7 forks [10s
> of] thousands of children from a single parent, resultings in thousands
> of vmas on the anon_vma list. shrink_inactive_list() must walk this
> list twice [page_referenced() and try_to_unmap()] under spin_lock for
> each anon page.

I wonder how far out RCU'ing the anon_vma lock is.

On Mon, May 21, 2007 at 09:53:34AM -0400, Lee Schermerhorn wrote:
> [Aside: Just last week, I encountered a similar situation on the
> i_mmap_lock for page cache pages running a 1200 user Oracle/OLTP run on
> a largish ia64 system. Left the system spitting out "soft lockup"
> messages/stack dumps overnight. Still spitting the next day, so I
> decided to reboot.]
> I have a patch that turns the anon_vma lock into a reader/writer lock
> that alleviates the problem somewhat, but with 10s of thousands of vmas
> on the lists, system still can't swap enough memory fast enough to
> recover.

Oh dear. Some algorithmic voodoo like virtually clustered scanning may
be in order in addition to anon_vma lock RCU'ing/etc.

On Mon, May 21, 2007 at 09:53:34AM -0400, Lee Schermerhorn wrote:
> We've run some AIM7 tests with Rik's "split lru list" patch, both with
> and without the anon_vma reader/writer lock patch. We'll be posting
> results later this week. Quick summary: with Rik's patch, AIM
> performance tanks earlier, as the system starts swapping earlier.
> However, system remains responsive to shell input. More into to follow.

I'm not sure where policy comes into this.

-- wli

Subject: Re: RSS controller v2 Test results (lmbench)
Posted by dev on Mon, 21 May 2007 15:03:27 GMT
View Forum Message <> Reply to Message

Andrew Morton wrote:
> On Thu, 17 May 2007 23:20:12 +0530
> Balbir Singh <balbir@linux.vnet.ibm.com> wrote:
>
>
>>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.
>
>
> <wakes up>
>
> Memory reclaim tends not to consume much CPU. Because in steady state it
> tends to be the case that the memory reclaim rate (and hopefully the
> scanning rate) is equal to the disk IO rate.

> Often the most successful way to identify performance problems in there is
> by careful code inspection followed by development of exploits.
>
> Is this RSS controller built on Paul's stuff, or is it standalone?
it is based on Paul's patches.

> Where do we stand on all of this now anyway? I was thinking of getting Paul's
> changes into -mm soon, see what sort of calamities that brings about.
I think we can merge Paul's patches with *interfaces* and then switch to
developing/reviewing/commiting resource subsytems.
RSS control had good feedback so far from a number of people
and is a first candidate imho.

Thanks,
Kirill

Subject: Re: RSS controller v2 Test results (lmbench)
Posted by Balbir Singh on Thu, 24 May 2007 07:36:26 GMT
View Forum Message <> Reply to Message

Kirill Korotaev wrote:
>> Where do we stand on all of this now anyway? I was thinking of getting Paul's
>> changes into -mm soon, see what sort of calamities that brings about.

> I think we can merge Paul's patches with *interfaces* and then switch to
> developing/reviewing/commiting resource subsytems.
> RSS control had good feedback so far from a number of people
> and is a first candidate imho.
>

Yes, I completely agree!

> Thanks,
> Kirill
>

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

Subject: Re: RSS controller v2 Test results (lmbench)
Posted by Paul Menage on Thu, 24 May 2007 07:39:21 GMT
View Forum Message <> Reply to Message

On 5/24/07, Balbir Singh <balbir@linux.vnet.ibm.com> wrote:
> Kirill Korotaev wrote:
> >> Where do we stand on all of this now anyway? I was thinking of getting Paul's
> >> changes into -mm soon, see what sort of calamities that brings about.
> > I think we can merge Paul's patches with *interfaces* and then switch to
> > developing/reviewing/commiting resource subsytems.
> > RSS control had good feedback so far from a number of people
> > and is a first candidate imho.
> >
>
> Yes, I completely agree!
>

I'm just finishing up the latest version of my container patches -
hopefully sending them out tomorrow.

Paul

Subject: Re: RSS controller v2 Test results (lmbench)
Posted by Balbir Singh on Thu, 24 May 2007 08:00:13 GMT
View Forum Message <> Reply to Message

Paul Menage wrote:
> On 5/24/07, Balbir Singh <balbir@linux.vnet.ibm.com> wrote:
>> Kirill Korotaev wrote:
>> >> Where do we stand on all of this now anyway? I was thinking of
>> getting Paul's
>> >> changes into -mm soon, see what sort of calamities that brings about.
>> > I think we can merge Paul's patches with *interfaces* and then
>> switch to
>> > developing/reviewing/commiting resource subsytems.
>> > RSS control had good feedback so far from a number of people
>> > and is a first candidate imho.
>> >
>>
>> Yes, I completely agree!
>>
>
> I'm just finishing up the latest version of my container patches -
> hopefully sending them out tomorrow.
>
> Paul

Thats good news! As I understand Kirill wanted to get your patches
in -mm and then get the RSS controller as the first candidate in
that uses the containers interfaces and I completely agree with
that approach.

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