Subject: Container Test Campaign
Posted by Clement Calmels on Wed, 07 Jun 2006 14:20:12 GMT
View Forum Message <> Reply to Message

Hello !

I'm part of a team of IBMers working on lightweight containers and we are going to start a new test campaign. Candidates are vserver, vserver context, namespaces (being pushed upstream), openvz, mcr (our simple container dedicated to migration) and eventually xen.

We will focus on the performance overhead but we are also interested in checkpoint/restart and live migration. A last topic would be how well the
resource managment criteria are met, but that's extra for the moment.

We plan on measuring performance overhead by comparing the results on a vanilla kernel with a partial and with a complete virtual environment. By partial, we mean the patched kernel and a 'namespace' virtualisation.

Test tools
----
o For network performance :

 * netpipe (http://www.scl.ameslab.gov/netpipe/)
 * netperf (http://www.netperf.org/netperf/NetperfPage.html)
 * tbench (http://samba.org/ftp/tridge/dbench/README)

o Filesystem :

  * dbench (http://samba.org/ftp/tridge/dbench/README)
  * iozone (http://www.iozone.org/)

o General

  * kernbench (http://ck.kolivas.org/kernbench/) stress cpu and
    filesystem through kernel compilation
  * More 'real world' application could be used, feel free to submit
    candidates...

We have experience on C/R and migration so we'll start with our own scenario, migrating oracle under load. The load is generated by DOTS (http://ltp.sourceforge.net/dotshowto.php).

If you could provided us some material on what has already been done : URL, bench tools, scenarios. We'll try to compile them in. configuration hints and tuning are most welcome if they are reasonable.

Results, tools, scenarios will be published on lxc.sf.net . We will set up the testing environment so as to be able to accept new versions, patches, test tools and rerun the all on demand. Results, tools, scenarios will be published on lxc.sf.net.

thanks !

Clement,

Subject: RE: Container Test Campaign
Posted by mef on Wed, 21 Jun 2006 19:25:04 GMT

View Forum Message <> Reply to Message

Hello Clement,

Sorry for the late response, as I have been on vacation.

We are interested in this test campaign. Our work so far has focused on performance, scalability, and isolation properties of vserver compared with xen. My guess is that you cc'd me due to the posting of our paper comparing vserver with xen (attached for those of you who have not seen it yet). In what way can be participate/contribute (i.e., where do we start)? We could share our test setup (except SpecWeb 99) that we used for our paper with everyone. Also, we'd appreciate if the folks participating in this test campaign could skim our paper and give us some feedback wrt the evaluation section and the appendix where we describe in reasonable the kernel vars, lvm partition setup, etc., we've used to eliminate differences between systems.

Best regards,
Marc

> -----Original Message-----
> From: Clement Calmels [mailto:clement.calmels@fr.ibm.com]
> Sent: Wednesday, June 07, 2006 10:20 AM
> To: devel@openvz.org; vserver@list.linux-vserver.org
> Cc: kir@openvz.org; dev@openvz.org; sam.vilain@catalyst.net.nz;
> mef@CS.Princeton.EDU; clg@fr.ibm.com; serue@us.ibm.com;
> haveblue@us.ibm.com; dlezcano@fr.ibm.com
> Subject: Container Test Campaign
>
>
> Hello !
>
> I'm part of a team of IBMers working on lightweight containers and we

> are going to start a new test campaign. Candidates are vserver,
> vserver context, namespaces (being pushed upstream), openvz, mcr (our
> simple container dedicated to migration) and eventually xen.
>
> We will focus on the performance overhead but we are also interested in
> checkpoint/restart and live migration. A last topic would be how well
> the
> resource managment criteria are met, but that's extra for the moment.
>
> We plan on measuring performance overhead by comparing the results on
> a vanilla kernel with a partial and with a complete virtual
> environment. By partial, we mean the patched kernel and a 'namespace'
> virtualisation.
>
> Test tools
> ----------
> o For network performance :
>
> * netpipe (http://www.scl.ameslab.gov/netpipe/)
> * netperf (http://www.netperf.org/netperf/NetperfPage.html)
> * tbench (http://samba.org/ftp/tridge/dbench/README)
>
> o Filesystem :
>
>   * dbench (http://samba.org/ftp/tridge/dbench/README)
>   * iozone (http://www.iozone.org/)
>
> o General
>
>   * kernbench (http://ck.kolivas.org/kernbench/) stress cpu and
>   filesystem through kernel compilation
>   * More 'real world' application could be used, feel free to submit
>   candidates...
>
> We have experience on C/R and migration so we'll start with our own
> scenario, migrating oracle under load. The load is generated by DOTS
> (http://ltp.sourceforge.net/dotshowto.php).
>
> If you could provided us some material on what has already been done :
> URL, bench tools, scenarios. We'll try to compile them in. configuration
> hints and tuning are most welcome if they are reasonable.
>
> Results, tools, scenarios will be published on lxc.sf.net . We will
> set up the testing environment so as to be able to accept new
> versions, patches, test tools and rerun the all on demand. Results,
> tools, scenarios will be published on lxc.sf.net.
>
> thanks !

>
> Clement,

File Attachments

1) paper.pdf, downloaded 836 times

Subject: RE: Container Test Campaign
Posted by mef on Wed, 21 Jun 2006 19:25:27 GMT
View Forum Message <> Reply to Message

Hi Clement,

You mention that testing isolation properties is more of an extra than an immediate criteria. Based on our experience, this actually is a fairly important criteria. Without decent isolation (both from a namespace and resource perspective) it is rather difficult to support lots of concurrent users. As our paper states, we run anywhere from 30-90 vservers per machine (each machine usually with a 2GHz processor and 1GB of RAM).

We are interested in checkpoint/restart too, but have nothing to test / contribute. I've forwarded your message to Jason Nieh @ Columbia. He has a relatively long history of working in that area. I saw a demo of their checkpoint/restart/migration support last December (live video migrated between servers within a single IBM blade system). Their latest paper published at USENIX LISA also states that they can migrate from one linux kernel version to another. This enables "live" system upgrade, which IMHO is just as important as load balancing.

Another area we are quite interested in is "network virtualization" (private route tables, ip tables, etc). We are aware that other container based systems (e.g., openvz) have support for this, but we (i.e., PlanetLab) are pretty much a vserver shop at the moment. We added our own support to safely share a single, public IPv4 address between multiple containers, while simultaneously support raw sockets etc. This is an absolute requirement for PlanetLab, and I'd argue (but not here) that it also is important for desktop usage scenarios that involve containers and want to avoid the use of NAT.

Best regards,
Marc

Subject: Re: Container Test Campaign
Posted by serue on Thu, 22 Jun 2006 11:31:35 GMT
View Forum Message <> Reply to Message

Quoting Marc E. Fiuczynski (mef@CS.Princeton.EDU):
> Hi Clement,
>
> You mention that testing isolation properties is more of an extra than an
> immediate criteria. Based on our experience, this actually is a fairly
> important criteria. Without decent isolation (both from a namespace and

As we develop our own patches for upstream inclusion, we will also be writing testcases to verify isolation, but obviously sitting down and writing such testcases for every c/r implementation is not something we can commit to :)

OTOH, perhaps we can collaborate on a test wrapper. This would require details to be filled in by each implementation's owner, but would save us from each having to come up with the boundary conditions. For instance, my testcase for the utsname patches which are in -mm is attached. While the testcase is specific to that implementation, by abstracting the "start a new container" command into a variable which can be filled in for vserver, openvz, etc, we might be able to come up with a generic utsname resource testing shell which can be easily filled in to work for each implementation.

Just a thought.

-serge

File Attachments

1) utstest.c, downloaded 320 times

Subject: Re: Container Test Campaign
Posted by serue on Thu, 22 Jun 2006 11:33:43 GMT
View Forum Message <> Reply to Message

Quoting Serge E. Hallyn (serue@us.ibm.com):
> OTOH, perhaps we can collaborate on a test wrapper. This would require
> details to be filled in by each implementation's owner, but would save
> us from each having to come up with the boundary conditions. For
> instance, my testcase for the utsname patches which are in -mm is
> attached. While the testcase is specific to that implementation, by
> abstracting the "start a new container" command into a variable which
> can be filled in for vserver, openvz, etc, we might be able to come up
> with a generic utsname resource testing shell which can be easily filled
> in to work for each implementation.
>
> Just a thought.

Oh yeah, and I meant to point out that once we have isolation testcases

for each implementation, then I think Clement would be able to easily
run these testcases along with the performance tests. (Clement, correct
me if I'm wrong :)

-serge

Subject: RE: Container Test Campaign
Posted by Clement Calmels on Thu, 22 Jun 2006 16:33:19 GMT
View Forum Message <> Reply to Message

Hi,

We have currently set up tests. Our goal is to provide a huge variety of test cases and measurements (to reduce imprecise plot). We started with same kind of microbenchmark: ltp, dbench, tbench and more complex benchmarks: kernel compilation. Tests are launched outside and inside a container. Besides we want to make tests with different number of container within a real node to get some clues on the different solutions's scalabilities.
After taking a glance at your paper, it seems we got same kind of results.
It would be a good idea to share our test setup. It seems there are easy ways to disadvantage one container solution against another one (Xen using loop device instead of a dedicated LVM partition for example). Or for example, the way we "share" a node between different containers could have some consequences on test results. I would prefer "fair" benchmarks.
In my opinion, a "fight" between the different container solutions would be as useless as a "Google Fight". But finding real world cases where a solution seems better than the others may result in more accurate conclusions.

Concerning the checkpoint/restart/migration topic, we (IBM) owned a solution called Metacluster. The main goal of Metacluster was the migration issue... but as a result it brought isolation in some areas (pid for example). We will use this solution and make some performance measures during the migration of well known application (Oracle under different workloads...). Openvz and Xen should be included in such benchs.

Best regards,
Clement.

Le mercredi 21 juin 2006 à 15:25 -0400, Marc E. Fiuczynski a écrit :
> Hello Clement,
>
> Sorry for the late response, as I have been on vacation.

>
> We are interested in this test campaign. Our work so far has focused on
> performance, scalability, and isolation properties of vserver compared with
> xen. My guess is that you cc'd me due to the posting of our paper comparing
> vserver with xen (attached for those of you who have not seen it yet). In
> what way can be participate/contribute (i.e., where do we start)? We could
> share our test setup (except SpecWeb 99) that we used for our paper with
> everyone. Also, we'd appreciate if the folks participating in this test
> campaign could skim our paper and give us some feedback wrt the evaluation
> section and the appendix where we describe in reasonable the kernel vars,
> lvm partition setup, etc., we've used to eliminate differences between
> systems.
>
> Best regards,
> Marc
>
>
> > -----Original Message-----
> > From: Clement Calmels [mailto:clement.calmels@fr.ibm.com]
> > Sent: Wednesday, June 07, 2006 10:20 AM
> > To: devel@openvz.org; vserver@list.linux-vserver.org
> > Cc: kir@openvz.org; dev@openvz.org; sam.vilain@catalyst.net.nz;
> > mef@CS.Princeton.EDU; clg@fr.ibm.com; serue@us.ibm.com;
> > haveblue@us.ibm.com; dlezcano@fr.ibm.com
> > Subject: Container Test Campaign
> >
> >
> > Hello !
> >
> > I'm part of a team of IBMers working on lightweight containers and we
> > are going to start a new test campaign. Candidates are vserver,
> > vserver context, namespaces (being pushed upstream), openvz, mcr (our
> > simple container dedicated to migration) and eventually xen.
> >
> > We will focus on the performance overhead but we are also interested in
> > checkpoint/restart and live migration. A last topic would be how well
> > the
> > resource managment criteria are met, but that's extra for the moment.
> >
> > We plan on measuring performance overhead by comparing the results on
> > a vanilla kernel with a partial and with a complete virtual
> > environment. By partial, we mean the patched kernel and a 'namespace'
> > virtualisation.
> >
> > Test tools
> > ----------
> > o For network performance :
> >

> > * netpipe (http://www.scl.ameslab.gov/netpipe/)
> > * netperf (http://www.netperf.org/netperf/NetperfPage.html)
> > * tbench (http://samba.org/ftp/tridge/dbench/README)
> >
> > o Filesystem :
> >
> >   * dbench (http://samba.org/ftp/tridge/dbench/README)
> >   * iozone (http://www.iozone.org/)
> >
> > o General
> >
> >   * kernbench (http://ck.kolivas.org/kernbench/) stress cpu and
> >   filesystem through kernel compilation
> >   * More 'real world' application could be used, feel free to submit
> >   candidates...
> >
> > We have experience on C/R and migration so we'll start with our own
> > scenario, migrating oracle under load. The load is generated by DOTS
> > (http://ltp.sourceforge.net/dotshowto.php).
> >
> > If you could provided us some material on what has already been done :
> > URL, bench tools, scenarios. We'll try to compile them in. configuration
> > hints and tuning are most welcome if they are reasonable.
> >
> > Results, tools, scenarios will be published on lxc.sf.net . We will
> > set up the testing environment so as to be able to accept new
> > versions, patches, test tools and rerun the all on demand. Results,
> > tools, scenarios will be published on lxc.sf.net.
> >
> > thanks !
> >
> > Clement,

Subject: Re: Container Test Campaign
Posted by Cedric Le Goater on Thu, 22 Jun 2006 21:39:16 GMT

View Forum Message <> Reply to Message

Marc E. Fiuczynski wrote:

> Sorry for the late response, as I have been on vacation.

a good thing indeed :)

> We are interested in this test campaign. Our work so far has focused on
> performance, scalability, and isolation properties of vserver compared with
> xen. My guess is that you cc'd me due to the posting of our paper comparing
> vserver with xen (attached for those of you who have not seen it yet).

yes.

> In what way can be participate/contribute (i.e., where do we start)? We could
> share our test setup (except SpecWeb 99) that we used for our paper with
> everyone. Also, we'd appreciate if the folks participating in this test
> campaign could skim our paper and give us some feedback wrt the evaluation
> section and the appendix where we describe in reasonable the kernel vars,
> lvm partition setup, etc., we've used to eliminate differences between
> systems.

OK, first half is great. Not finished yet but some of your results confirm
ours : vserver is great, openvz will be great, xen overhead is important
and does not scale as well. What we expect from you is also feedback on the
test and their environment to make sure they are relevant.

Xen is not our major focus but if we have time we'll torture it a bit.

thanks,

C.

Subject: Re: Container Test Campaign
Posted by Cedric Le Goater on Thu, 22 Jun 2006 21:51:10 GMT
View Forum Message <> Reply to Message

Hi marc !

Marc E. Fiuczynski wrote:

> You mention that testing isolation properties is more of an extra than an
> immediate criteria. Based on our experience, this actually is a fairly
> important criteria. Without decent isolation (both from a namespace and
> resource perspective) it is rather difficult to support lots of concurrent
> users. As our paper states, we run anywhere from 30-90 vservers per machine
> (each machine usually with a 2GHz processor and 1GB of RAM).

is that a common setup for planet lab or a maximum ? how many vservers/
vcontext do you think we should try to reach ?

> We are interested in checkpoint/restart too, but have nothing to test /
> contribute. I've forwarded your message to Jason Nieh @ Columbia. He has a
> relatively long history of working in that area. I saw a demo of their
> checkpoint/restart/migration support last December (live video migrated
> between servers within a single IBM blade system).

we've worked a few years with a zap guy. I only wished they were bit more open (source) about what they've been doing since crak.

> Their latest paper
> published at USENIX LISA also states that they can migrate from one linux
> kernel version to another. This enables "live" system upgrade, which IMHO
> is just as important as load balancing.

this feature is one the *major* features of mobile containers but it will
require specific kernel APIs to make it maintainable on the long term.

> Another area we are quite interested in is "network virtualization" (private
> route tables, ip tables, etc). We are aware that other container based
> systems (e.g., openvz) have support for this, but we (i.e., PlanetLab) are
> pretty much a vserver shop at the moment. We added our own support to
> safely share a single, public IPv4 address between multiple containers,
> while simultaneously support raw sockets etc. This is an absolute
> requirement for PlanetLab, and I'd argue (but not here) that it also is
> important for desktop usage scenarios that involve containers and want to
> avoid the use of NAT.

Did you contribute that feature to vserver ?

So you have different containers exposing the same IP address ? How do you
assign incoming packets to a container ?

thanks,

C.

Subject: Re: Container Test Campaign
Posted by Sam Vilain on Thu, 22 Jun 2006 23:39:24 GMT
View Forum Message <> Reply to Message

Serge E. Hallyn wrote:
> Quoting Marc E. Fiuczynski (mef@CS.Princeton.EDU):
>> Hi Clement,
>>
>> You mention that testing isolation properties is more of an extra than an
>> immediate criteria. Based on our experience, this actually is a fairly
>> important criteria. Without decent isolation (both from a namespace and
>
> As we develop our own patches for upstream inclusion, we will also be
> writing testcases to verify isolation, but obviously sitting down and
> writing such testcases for every c/r implementation is not something we
> can commit to :)
>
> OTOH, perhaps we can collaborate on a test wrapper. This would require

> details to be filled in by each implementation's owner, but would save
> us from each having to come up with the boundary conditions. For
> instance, my testcase for the utsname patches which are in -mm is
> attached. While the testcase is specific to that implementation, by
> abstracting the "start a new container" command into a variable which
> can be filled in for vserver, openvz, etc, we might be able to come up
> with a generic utsname resource testing shell which can be easily filled
> in to work for each implementation.
>
> Just a thought.
>
> -serge

You might like to consider making the output of these tests use the
Perl Test::TAP output; eg:

```
 1..5 # declare how many tests you expect to run - 0..0 means unknown
 ok 1 # pass test one
 not ok 2 # this one failed - perhaps say why in comments
 ok 3 # SKIP somereason - this test was skipped
 ok 4
 ok 5
```

This makes it easier to run the tests inside Harnesses.

Sam.

```
>
>  ------------------------------------------------------------ ----------
>
> /*
>  * Copyright (C) 2005 IBM
>  * Author: Serge Hallyn <serue@us.ibm.com>
>  * Compile using "gcc -o utstest utstest.c"
>  * Run using "for i in `seq 1 5`; do  ./utstest $i; done"
>  *
>  * test1:
>     P1: A=gethostname
>     P2: B=gethostname
>     Ensure(A==B)
>
>  * test2:
>     P1: sethostname(newname); A=gethostname
>     P2: (wait); B=gethostname
>     Ensure (A==B)
>
>  * test3:
>     P1: A=gethostname; unshare(utsname); sethostname(newname);
```

```
C=gethostname
>     P2: B=gethostname; (wait); (wait); D=gethostname
>     Ensure (A==B && A==D && C!=D)
>
>  * test4:
>     P1: A=gethostname; unshare(utsname); (wait); C=gethostname
>     P2: B=gethostname; (wait); sethostname(newname); D=gethostname
>     Ensure (A==B && A==C && C!=D)
>
>  * test5:
>     P1: A=gethostname; unshare(utsname) without suff. perms; (wait);
C=gethostname
>     P2: B=gethostname; (wait); sethostname(newname); D=gethostname
>     Ensure (A==B==C==D) and state is ok.
>  *
>  */
>
> #include <sys/wait.h>
> #include <assert.h>
> #include <stdio.h>
> #include <stdlib.h>
> #include <unistd.h>
> #include <string.h>
> #include <errno.h>
>
> int drop_root()
> {
>     int ret;
>     ret = setresuid(1000, 1000, 1000);
>     if (ret) {
>         perror("setresuid");
>         exit(4);
>     }
>     return 1;
> }
>
> #include <linux/unistd.h>
>
> static inline _syscall1 (int,  unshare, int, flags)
>
> #ifndef CLONE_NEWUTS
> #define CLONE_NEWUTS            0x04000000      /* New utsname group? */
> #endif
>
> int p1fd[2], p2fd[2];
> pid_t cpid;
> int testnum;
>
```

```
> #define HLEN 100
> #define NAME1 "serge1"
> #define NAME2 "serge2"
>
> void picknewhostname(char *orig, char *new)
> {
>     memset(new, 0, HLEN);
>     if (strcmp(orig, NAME1) == 0)
>         strcpy(new, NAME2);
>     else
>         strcpy(new, NAME1);
> }
>
> void P1(void)
> {
>     char hostname[HLEN], newhostname[HLEN], rhostname[HLEN];
>     int err;
>     int len;
>
>     close(p1fd[1]);
>     close(p2fd[0]);
>
>     switch(testnum) {
>     case 1:
>         gethostname(hostname, HLEN);
>         len = read(p1fd[0], rhostname, HLEN);
>         if (len == strlen(hostname) &&
>             strncmp(hostname, rhostname, len) == 0) {
>             printf("test 1: success\n");
>             exit(0);
>         }
>         printf("test 1: fail\n");
>         printf("Proc 1: hostname %s\n", hostname);
>         printf("test 2: hostname %s\n", rhostname);
>         exit(1);
>     case 2:
>         gethostname(hostname, HLEN);
>         picknewhostname(hostname, newhostname);
>         err = sethostname(newhostname, strlen(newhostname));
>         write(p2fd[1], "1", 1);
>         if (err == -1) { perror("sethostname"); exit(1); }
>         len = read(p1fd[0], rhostname, HLEN);
>         if (len == strlen(newhostname) &&
>                 strncmp(newhostname, rhostname, len) == 0) {
>             printf("test 2: success\n");
>             exit(0);
>         }
>         printf("test 2: fail\n");
>       printf("Proc 1: hostname %s\n", newhostname);
>       printf("test 2: hostname %s\n", rhostname);
>       exit(1);
>   case 3:
>       gethostname(hostname, HLEN);
>       picknewhostname(hostname, newhostname);
>       err = unshare(CLONE_NEWUTS);
>       printf("unshare returned %d (should be 0)\n", err);
>       err = sethostname(newhostname, strlen(newhostname));
>       write(p2fd[1], "1", 1);
>       if (err == -1) { perror("sethostname"); exit(1); }
>
>       len = read(p1fd[0], rhostname, HLEN);
>       if (len == strlen(newhostname) &&
>               strncmp(newhostname, rhostname, len) == 0) {
>           printf("test 3: fail\n");
>           printf("Proc 1: hostname %s\n", newhostname);
>           printf("test 2: hostname %s\n", rhostname);
>           printf("These should have been different\n");
>           exit(1);
>       }
>       if (len == strlen(hostname) &&
>               strncmp(hostname, rhostname, len) == 0) {
>           printf("test 3: success\n");
>           exit(0);
>       }
>       printf("test 3: fail\n");
>       printf("Proc 1: original hostname %s\n", hostname);
>       printf("Proc 2: hostname %s\n", rhostname);
>       printf("These should have been the same\n");
>       exit(1);
>
>   case 4:
>       gethostname(hostname, HLEN);
>       err = unshare(CLONE_NEWUTS);
>       printf("unshare returned %d (should be 0)\n", err);
>       write(p2fd[1], "1", 1); /* tell p2 to go ahead and sethostname */
>       len = read(p1fd[0], rhostname, HLEN);
>       gethostname(newhostname, HLEN);
>       if (strcmp(hostname, newhostname) != 0) {
>           printf("test 4: fail\n");
>           printf("Proc 1: hostname %s\n", hostname);
>           printf("Proc 1: new hostname %s\n", newhostname);
>           printf("These should have been the same\n");
>           exit(1);
>       }
>       if (strncmp(hostname, rhostname, len)==0) {
>           printf("test 4: fail\n");
>           printf("Proc 1: hostname %s\n", hostname);
>           printf("Proc 2: new hostname %s\n", rhostname);
>           printf("These should have been different\n");
>           exit(1);
>       }
>       printf("test 4: success\n");
>       exit(0);
>   case 5:
>       /* drop CAP_SYS_ADMIN, then do same as case 4 but check
>        * that hostname != newhostname && rhostname == newhostname */
>       if (!drop_root()) {
>           printf("failed to drop root.\n");
>           exit(3);
>       }
>       gethostname(hostname, HLEN);
>       err = unshare(CLONE_NEWUTS);
>       printf("unshare returned %d (should be -1)\n", err);
>       write(p2fd[1], "1", 1); /* tell p2 to go ahead and sethostname */
>       len = read(p1fd[0], rhostname, HLEN);
>       gethostname(newhostname, HLEN);
>       if (strncmp(newhostname, rhostname, len)!=0) {
>           printf("test 5: fail\n");
>           printf("Proc 1: newhostname %s\n", newhostname);
>           printf("Proc 2: new hostname %s\n", rhostname);
>           printf("These should have been the same\n");
>           exit(1);
>       }
>       if (strcmp(hostname, newhostname) == 0) {
>           printf("test 5: fail\n");
>           printf("Proc 1: hostname %s\n", hostname);
>           printf("Proc 1: new hostname %s\n", newhostname);
>           printf("These should have been different\n");
>           exit(1);
>       }
>       printf("test 5: success\n");
>       exit(0);
>   default:
>       break;
>   }
>   return;
> }
>
> void P2(void)
> {
>   char hostname[HLEN], newhostname[HLEN];
>   int len;
>   int err;
>
```

```
>     close(p1fd[0]);
>     close(p2fd[1]);
>
>     switch(testnum) {
>     case 1:
>         gethostname(hostname, HLEN);
>         write(p1fd[1], hostname, strlen(hostname));
>         break;
>     case 2:
>     case 3:
>         len = 0;
>         while (!len) {
>             len = read(p2fd[0], hostname, 1);
>         }
>         gethostname(hostname, HLEN);
>         write(p1fd[1], hostname, strlen(hostname));
>         break;
>     case 4:
>     case 5:
>         len = 0;
>         while (!len) {
>             len = read(p2fd[0], hostname, 1);
>         }
>         gethostname(hostname, HLEN);
>         picknewhostname(hostname, newhostname);
>         sethostname(newhostname, strlen(newhostname));
>         write(p1fd[1], newhostname, strlen(newhostname));
>         break;
>     default:
>         printf("undefined test: %d\n", testnum);
>         break;
>     }
>     return;
> }
>
> int main(int argc, char *argv[])
> {
>     if (argc != 2) {
>         printf("Usage: %s <testnum>\n", argv[0]);
>         printf(" where testnum is between 1 and 5 inclusive\n");
>         exit(2);
>     }
>     if (pipe(p1fd) == -1) { perror("pipe"); exit(EXIT_FAILURE); }
>     if (pipe(p2fd) == -1) { perror("pipe"); exit(EXIT_FAILURE); }
>
>     testnum = atoi(argv[1]);
>     if (testnum < 1 || testnum > 5) {
>         printf("testnum should be between 1 and 5 inclusive.\n");
>         exit(2);
>     }
>
>     cpid = fork();
>
>     if (cpid == -1) { perror("fork"); exit(EXIT_FAILURE); }
>
>     if (cpid == 0)
>         P1();
>     else
>         P2();
>
>     return 0;
> }
```

--
Sam Vilain, Catalyst IT (NZ) Ltd.
phone: +64 4 499 2267   cell: +64 21 55 40 50
DDI: +64 4 803 2342   PGP ID: 0x66B25843

Subject: Re: Container Test Campaign

Posted by Sam Vilain on Fri, 23 Jun 2006 03:40:49 GMT

View Forum Message <> Reply to Message

Marc E. Fiuczynski wrote:
> Hello Clement,
>
> Sorry for the late response, as I have been on vacation.
>
> We are interested in this test campaign. Our work so far has
> focused on performance, scalability, and isolation properties of
> vserver compared with xen. My guess is that you cc'd me due to the
> posting of our paper comparing vserver with xen (attached for those
> of you who have not seen it yet). In what way can be
> participate/contribute (i.e., where do we start)? We could share
> our test setup (except SpecWeb 99) that we used for our paper with
> everyone. Also, we'd appreciate if the folks participating in this
> test campaign could skim our paper and give us some feedback wrt
> the evaluation section and the appendix where we describe in
> reasonable the kernel vars, lvm partition setup, etc., we've used
> to eliminate differences between systems.

One area it would be interesting to see benchmarks for is the
performance impact of filesystem unification and a lot of vservers -
for instance, a system with 10 vservers, each running apache and
actively serving pages, I'd expect to see more cache hits at the L2

and/or L3 CPU cache layers on account of the fact that, eg, C
libraries are not being paged out to load in other (identical) C
libraries.

My guess is that you just can't leverage that kind of benefit from a
hypervisor approach, but I don't really know enough about how they
work under the hood to be able to say.

Sam.

>> -----Original Message----- From: Clement Calmels
>> [mailto:clement.calmels@fr.ibm.com] Sent: Wednesday, June 07,
>> 2006 10:20 AM To: devel@openvz.org;
>> vserver@list.linux-vserver.org Cc: kir@openvz.org;
>> dev@openvz.org; sam.vilain@catalyst.net.nz; mef@CS.Princeton.EDU;
>> clg@fr.ibm.com; serue@us.ibm.com; haveblue@us.ibm.com;
>> dlezcano@fr.ibm.com Subject: Container Test Campaign
>>
>>
>> Hello !
>>
>> I'm part of a team of IBMers working on lightweight containers
>> and we are going to start a new test campaign. Candidates are
>> vserver, vserver context, namespaces (being pushed upstream),
>> openvz, mcr (our simple container dedicated to migration) and
>> eventually xen.
>>
>> We will focus on the performance overhead but we are also
>> interested in checkpoint/restart and live migration. A last topic
>> would be how well the resource managment criteria are met, but
>> that's extra for the moment.
>>
>> We plan on measuring performance overhead by comparing the
>> results on a vanilla kernel with a partial and with a complete
>> virtual environment. By partial, we mean the patched kernel and a
>> 'namespace' virtualisation.
>>
>> Test tools ---------- o For network performance :
>>
>> * netpipe (http://www.scl.ameslab.gov/netpipe/) * netperf
>> (http://www.netperf.org/netperf/NetperfPage.html) * tbench
>> (http://samba.org/ftp/tridge/dbench/README)
>>
>> o Filesystem :
>>
>> * dbench (http://samba.org/ftp/tridge/dbench/README) * iozone
>> (http://www.iozone.org/)
>>

>> o General
>>
>> * kernbench (http://ck.kolivas.org/kernbench/) stress cpu and
>> filesystem through kernel compilation * More 'real world'
>> application could be used, feel free to submit candidates...
>>
>> We have experience on C/R and migration so we'll start with our
>> own scenario, migrating oracle under load. The load is generated
>> by DOTS (http://ltp.sourceforge.net/dotshowto.php).
>>
>> If you could provided us some material on what has already been
>> done : URL, bench tools, scenarios. We'll try to compile them in.
>> configuration hints and tuning are most welcome if they are
>> reasonable.
>>
>> Results, tools, scenarios will be published on lxc.sf.net . We
>> will set up the testing environment so as to be able to accept
>> new versions, patches, test tools and rerun the all on demand.
>> Results, tools, scenarios will be published on lxc.sf.net.
>>
>> thanks !
>>
>> Clement,

Subject: RE: Container Test Campaign
Posted by mef on Fri, 23 Jun 2006 07:40:34 GMT

View Forum Message <> Reply to Message

> > (each machine usually with a 2GHz processor and 1GB of RAM).
>
> is that a common setup for planet lab or a maximum ?

We are working to upgrade to dual-core 3Ghz sysems w/ 4GB of RAM.

> how many vservers/ vcontext do you think we should try to reach ?

As mentioned in our paper, we run anywhere from 30-90 on a single box.

> Did you contribute that feature to vserver ?

Yes (Mark Huang @ Princeton) added this feature as a kernel module that
interfaces with vserver.

> So you have different containers exposing the same IP address ? How do you
> assign incoming packets to a container ?

I'll let Mark Huang provide further details.

Marc

Subject: RE: Container Test Campaign
Posted by mef on Fri, 23 Jun 2006 07:40:45 GMT
View Forum Message <> Reply to Message

Hi Sam,

> One area it would be interesting to see benchmarks for is the
> performance impact of filesystem unification and a lot of vservers -
> for instance, a system with 10 vservers, each running apache and
> actively serving pages, I'd expect to see more cache hits at the L2
> and/or L3 CPU cache layers on account of the fact that, eg, C
> libraries are not being paged out to load in other (identical) C
> libraries.

We can try this with our SPECWEB'99 benchmark. Our current setup does not use FS unification.

> My guess is that you just can't leverage that kind of benefit from a
> hypervisor approach, but I don't really know enough about how they
> work under the hood to be able to say.

Right, our current benchmark shows that even w/o the benefit of FS unification that a container-based approach (in our case vserver) still significantly outperforms Xen on both SPECWEB'99 and OSDB benchmarks. Please take a peek at our evaluation sections in the paper that I forwarded before.

Marc

Subject: Re: Container Test Campaign
Posted by Mark Huang on Fri, 23 Jun 2006 17:31:15 GMT
View Forum Message <> Reply to Message

Cedric Le Goater wrote:
> Did you contribute that feature to vserver ?

The feature is fairly specific to our needs and would not be very useful to the most common vserver use case (shared hosting).

> So you have different containers exposing the same IP address ? How do you
> assign incoming packets to a container ?

We wrote a kernel module that leverages netfilter hooks and ip_conntrack. You're only allowed to send IP, but you can send IP packets through any type of socket (TCP, UDP, raw IP, or even raw packet). As Marc mentioned, this flexibility was an absolute requirement.

The kernel module sits in the input and output path of the stack, and associates every incoming and outgoing packet with an ip_conntrack struct (to which we added container IDs). Once a container sends out an outgoing packet, it is entitled to receive incoming packets associated with that connection. A container may also receive incoming packets associated with ports that it has reserved by calling bind() (the kernel module keeps track of bind() calls). You can think of the kernel module as a local stateful firewall for sockets.

To users, it looks like they can run pretty much anything root would be able to, including programs that use raw IP sockets (ping and traceroute), programs that use raw packet sockets (tcpdump), and regular server apps (Apache, MySQL, etc.). When they run tcpdump, they of course only see packets that they "own" (i.e., packets that are associated with their active connections).

There's technical documentation for the kernel module on our website:

http://www.planet-lab.org/doc/vnet.php

The kernel module does a lot more than this as well, which is another reason that it hasn't been merged into mainline vserver. Recent features include virtualized TUN/TAP and IP aliasing support.

Lastly, you're of course free to browse the code:

http://cvs.planet-lab.org/cvs/vnet/

Subject: Re: Container Test Campaign
Posted by Sam Vilain on Sun, 25 Jun 2006 22:00:07 GMT

View Forum Message <> Reply to Message

Marc E. Fiuczynski wrote:
>> My guess is that you just can't leverage that kind of benefit
>> from a hypervisor approach, but I don't really know enough about
>> how they work under the hood to be able to say.
>
> Right, our current benchmark shows that even w/o the benefit of FS
> unification that a container-based approach (in our case vserver)
> still significantly outperforms Xen on both SPECWEB'99 and OSDB
> benchmarks. Please take a peek at our evaluation sections in the
> paper that I forwarded before.

Yes, and very nice paper it is indeed, though I'm still curious as to

why all the dbench scores go down for SMP, even on the vanilla Linux
kernel.

I guess that demonstrating the effects of the unification would have
to come under the scalability benchmarks section, and probably with a
non-IO bound test. If Specweb doesn't support testing a bunch of
machines at once, perhaps you could fudge it by using IP Virtual
Server on another machine to distribute the load across the VM instances.

Sam.

Subject: RE: Container Test Campaign
Posted by mef on Mon, 26 Jun 2006 08:57:19 GMT

View Forum Message <> Reply to Message

> Mark Huang wrote:
>> Cedric Le Goater wrote:
>> Did you contribute that feature to vserver ?
>
> The feature is fairly specific to our needs and would not be very
> useful to the most common vserver use case (shared hosting).

Indeed this is true. However, it would enable containers to be used on linux desktop/laptop/cellphone users, as they provide a nice clean way to keep third party software separate/protected from the base installation. In such environments (assuming IPv6 cannot be used) it would simplify matters if the containers could share a global IP address vs. needing to do NAT traversal and requiring supernodes for things like skype etc.. While this might not be the targeted usage scenario for the moment, I would not ignore this segment.

Marc

Subject: Re: Container Test Campaign
Posted by Clement Calmels on Fri, 30 Jun 2006 17:28:06 GMT

View Forum Message <> Reply to Message

Hi,

A first round about virtualisation benchmarks can be found here:
http://lxc.sourceforge.net/bench/
These benchmarks run with vanilla kernels and the patched versions of
well know virtualisation solutions: VServer and OpenVZ. Some benchs also
run inside the virtual 'guest' but we ran into trouble trying to run
some of them... probably virtual 'guest' configuration issues... we will
trying to fix them...

The metacluster migration solution (formely a Meiosys company produt) was added as it seems that the checkpoint/restart topic is close to the virtualisation's one (OpenVZ now provides a checkpoint/restart capability).
For the moment, benchmarks only ran on xeon platform but we expect more architecture soon. Besides the 'classic' benchs used, more network oriented benchs will be added. Netpipe between two virtual 'guests' for example. We hope we will be able to provide results concerning the virtual 'guest' scalability, running several 'guest' at the same time.

Best regards,

Le mercredi 07 juin 2006 à 16:20 +0200, Clement Calmels a écrit :
> Hello !
>
> I'm part of a team of IBMers working on lightweight containers and we
> are going to start a new test campaign. Candidates are vserver,
> vserver context, namespaces (being pushed upstream), openvz, mcr (our
> simple container dedicated to migration) and eventually xen.
>
> We will focus on the performance overhead but we are also interested in
> checkpoint/restart and live migration. A last topic would be how well
> the
> resource managment criteria are met, but that's extra for the moment.
>
> We plan on measuring performance overhead by comparing the results on
> a vanilla kernel with a partial and with a complete virtual
> environment. By partial, we mean the patched kernel and a 'namespace'
> virtualisation.
>
> Test tools
> ----------
> o For network performance :
>
> * netpipe (http://www.scl.ameslab.gov/netpipe/)
> * netperf (http://www.netperf.org/netperf/NetperfPage.html)
> * tbench (http://samba.org/ftp/tridge/dbench/README)
>
> o Filesystem :
>
>   * dbench (http://samba.org/ftp/tridge/dbench/README)
>   * iozone (http://www.iozone.org/)
>
> o General
>
>   * kernbench (http://ck.kolivas.org/kernbench/) stress cpu and
>   filesystem through kernel compilation

>   * More 'real world' application could be used, feel free to submit
>   candidates...
>
> We have experience on C/R and migration so we'll start with our own
> scenario, migrating oracle under load. The load is generated by DOTS
> (http://ltp.sourceforge.net/dotshowWe ran into trouble trying to run sto.php).
>
> If you could provided us some material on what has already been done :
> URL, bench tools, scenarios. We'll try to compile them in. configuration
> hints and tuning are most welcome if they are reasonable.
>
> Results, tools, scenarios will be published on lxc.sf.net . We will
> set up the testing environment so as to be able to accept new
> versions, patches, test tools and rerun the all on demand. Results,
> tools, scenarios will be published on lxc.sf.net.
>
> thanks !
>
> Clement,
>

--
Clément Calmels <clement.calmels@fr.ibm.com>

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by Herbert Poetzl on Sun, 02 Jul 2006 15:32:29 GMT
View Forum Message <> Reply to Message

> Hi,
>
> A first round about virtualisation benchmarks can be found here:
> http://lxc.sourceforge.net/bench/

very interesting results, tx ...

> These benchmarks run with vanilla kernels and the patched versions of
> well know virtualisation solutions: VServer and OpenVZ. Some benchs
> also run inside the virtual 'guest' but we ran into trouble trying to
> run some of them... probably virtual 'guest' configuration issues...
> we will trying to fix them... The metacluster migration solution
> (formely a Meiosys company produt) was added as it seems that the
> checkpoint/restart topic is close to the virtualisation's one (OpenVZ
> now provides a checkpoint/restart capability).

from the tests:
 "For benchs inside real 'guest' nodes (OpenVZ/VServer) you should
  take into account that the FS tested is not the 'host' node one's."

at least for Linux-VServer it should not be hard to avoid the
chroot/filesystem namespace part and have it run on the host fs.
a bind mount into the guest might do the trick too, if you need
help to accomplish that, just let me know ...

> For the moment, benchmarks only ran on xeon platform but we expect
> more architecture soon. Besides the 'classic' benchs used, more
> network oriented benchs will be added. Netpipe between two virtual
> 'guests' for example. We hope we will be able to provide results
> concerning the virtual 'guest' scalability, running several 'guest'
> at the same time.

best,
Herbert

> Best regards,
>

> > Hello !
> >
> > I'm part of a team of IBMers working on lightweight containers and we
> > are going to start a new test campaign. Candidates are vserver,
> > vserver context, namespaces (being pushed upstream), openvz, mcr (our
> > simple container dedicated to migration) and eventually xen.
> >
> > We will focus on the performance overhead but we are also interested in
> > checkpoint/restart and live migration. A last topic would be how well
> > the
> > resource managment criteria are met, but that's extra for the moment.
> >
> > We plan on measuring performance overhead by comparing the results on
> > a vanilla kernel with a partial and with a complete virtual
> > environment. By partial, we mean the patched kernel and a 'namespace'
> > virtualisation.
> >
> > Test tools
> > ----------
> > o For network performance :
> >
> > * netpipe (http://www.scl.ameslab.gov/netpipe/)
> > * netperf (http://www.netperf.org/netperf/NetperfPage.html)
> > * tbench (http://samba.org/ftp/tridge/dbench/README)
> >
> > o Filesystem :
> >
> >   * dbench (http://samba.org/ftp/tridge/dbench/README)
> >   * iozone (http://www.iozone.org/)

> >
> > o General
> >
> >   * kernbench (http://ck.kolivas.org/kernbench/) stress cpu and
> >   filesystem through kernel compilation
> >   * More 'real world' application could be used, feel free to submit
> >   candidates...
> >
> > We have experience on C/R and migration so we'll start with our own
> > scenario, migrating oracle under load. The load is generated by DOTS
> > (http://ltp.sourceforge.net/dotshowWe ran into trouble trying to run sto.php).
> >
> > If you could provided us some material on what has already been done :
> > URL, bench tools, scenarios. We'll try to compile them in. configuration
> > hints and tuning are most welcome if they are reasonable.
> >
> > Results, tools, scenarios will be published on lxc.sf.net . We will
> > set up the testing environment so as to be able to accept new
> > versions, patches, test tools and rerun the all on demand. Results,
> > tools, scenarios will be published on lxc.sf.net.
> >
> > thanks !
> >
> > Clement,
> >
> --

>
> ____
> Vserver mailing list
> Vserver@list.linux-vserver.org
> http://list.linux-vserver.org/mailman/listinfo/vserver

Subject: Re: Container Test Campaign
Posted by Kirill Korotaev on Mon, 03 Jul 2006 07:49:45 GMT

View Forum Message <> Reply to Message

>From what I see just just after 1 minute check of your results:

DBench:
- different disk I/O schedulers are compared. This makes comparison useless
  (not virtualization technologies are compared itself).
- the fact that there is too much difference between measurements
  (e.g. vserver makes linux faster :lol:) makes me believe that you use large disk partiion,
  where data blocks allocation on the disk influence your results.
  To make these measurements correct the same partition with the size closest to the required
  max disk space should be used in all DBench tests.

TBench:
- when running inside VE, please, make sure, that /proc/user_beancounters doesn't show you
  resource allocation fails (failcnt column).
  Resource limits set by default can be just too small to finish your test case.
  And this doesn't mean your conclusion 'Concerning the results, obviously more isolation brings more overhead.'.
  I'm really suprised to see such statements.

I also noticed that you do the measurements with different HZ settings.
This influences the results as well...

BTW, do you plan to do functional testing in addition to performance?

Thanks,
Kirill

> Hi,
>
> A first round about virtualisation benchmarks can be found here:
> http://lxc.sourceforge.net/bench/
> These benchmarks run with vanilla kernels and the patched versions of
> well know virtualisation solutions: VServer and OpenVZ. Some benchs also
> run inside the virtual 'guest' but we ran into trouble trying to run
> some of them... probably virtual 'guest' configuration issues... we will
> trying to fix them...
> The metacluster migration solution (formely a Meiosys company produt)
> was added as it seems that the checkpoint/restart topic is close to the
> virtualisation's one (OpenVZ now provides a checkpoint/restart
> capability).
> For the moment, benchmarks only ran on xeon platform but we expect more
> architecture soon. Besides the 'classic' benchs used, more network
> oriented benchs will be added. Netpipe between two virtual 'guests' for
> example. We hope we will be able to provide results concerning the
> virtual 'guest' scalability, running several 'guest' at the same time.
>
> Best regards,
>
>
> Le mercredi 07 juin 2006 à 16:20 +0200, Clement Calmels a écrit :
>
>>Hello !
>>
>>I'm part of a team of IBMers working on lightweight containers and we
>>are going to start a new test campaign. Candidates are vserver,
>>vserver context, namespaces (being pushed upstream), openvz, mcr (our
>>simple container dedicated to migration) and eventually xen.

```
>>
>>We will focus on the performance overhead but we are also interested in
>>checkpoint/restart and live migration. A last topic would be how well
>>the
>>resource managment criteria are met, but that's extra for the moment.
>>
>>We plan on measuring performance overhead by comparing the results on
>>a vanilla kernel with a partial and with a complete virtual
>>environment. By partial, we mean the patched kernel and a 'namespace'
>>virtualisation.
>>
>>Test tools
>>----------
>>o For network performance :
>>
>> * netpipe (http://www.scl.ameslab.gov/netpipe/)
>> * netperf (http://www.netperf.org/netperf/NetperfPage.html)
>> * tbench (http://samba.org/ftp/tridge/dbench/README)
>>
>>o Filesystem :
>>
>>  * dbench (http://samba.org/ftp/tridge/dbench/README)
>>  * iozone (http://www.iozone.org/)
>>
>>o General
>>
>>  * kernbench (http://ck.kolivas.org/kernbench/) stress cpu and
>>    filesystem through kernel compilation
>>  * More 'real world' application could be used, feel free to submit
>>    candidates...
>>
>>We have experience on C/R and migration so we'll start with our own
>>scenario, migrating oracle under load. The load is generated by DOTS
>>(http://ltp.sourceforge.net/dotshowWe ran into trouble trying to run sto.php).
>>
>>If you could provided us some material on what has already been done :
>>URL, bench tools, scenarios. We'll try to compile them in. configuration
>>hints and tuning are most welcome if they are reasonable.
>>
>>Results, tools, scenarios will be published on lxc.sf.net . We will
>>set up the testing environment so as to be able to accept new
>>versions, patches, test tools and rerun the all on demand. Results,
>>tools, scenarios will be published on lxc.sf.net.
>>
>>thanks !
>>
>>Clement,
>>
```

Subject: Re: Container Test Campaign
Posted by kir on Mon, 03 Jul 2006 08:14:55 GMT
View Forum Message <> Reply to Message

Clement,

Thanks for sharing the results! A few comments...

(1) General

1.1 It would be nice to run vmstat (say, vmstat 10) for the duration of the tests, and put the vmstat output logs to the site.

1.2 Can you tell how you run the tests. I am particularly interested in
- how many iterations do you do?
- what result do you choose from those iterations?
- how reproducible are the results?
- are you rebooting the box between the iterations?
- are you reformatting the partition used for filesystem testing?
- what settings are you using (such as kernel vm params)?
- did you stop cron daemons before running the test?
- are you using the same test binaries across all the participants?
- etc. etc...

Basically, the detailed description of a process would be nice to have, in order to catch possible problems. There are a lot of tiny things which are influencing the results. For example, in linux kernels 2.4 binding the NIC IRQ to a single CPU on an SMP system boosts network performance by about 15%! Sure this is not relevant here, it's just an example.

1.3 Would be nice to have diffs between different kernel configs.

(2) OpenVZ specifics

2.1 Concerning the tests running inside an OpenVZ VE, the problem is there is a (default) set of resource limits applied to each VE. Basically one should tailor those limits to suit the applications running, OR, for the purpose of testing, just set those limits to some very high values so they will never be reached.

For example, the tbench test is probably failed to finish because it hits the limits for privvmpages, tcpsndbuf and tcprcvbuf. I have increased the limits for those parameters and the test was finished successfully. Also, dbench test could hit the disk quota limit for a VE.

Some more info is available at http://wiki.openvz.org/Resource_management

2.2 For OpenVZ specifically, it would be nice to collect

/proc/user_beancounters output before and after the test.

Clément Calmels wrote:
> Hi,
>
> A first round about virtualisation benchmarks can be found here:
> http://lxc.sourceforge.net/bench/
> These benchmarks run with vanilla kernels and the patched versions of
> well know virtualisation solutions: VServer and OpenVZ. Some benchs also
> run inside the virtual 'guest' but we ran into trouble trying to run
> some of them... probably virtual 'guest' configuration issues... we will
> trying to fix them...
> The metacluster migration solution (formely a Meiosys company produt)
> was added as it seems that the checkpoint/restart topic is close to the
> virtualisation's one (OpenVZ now provides a checkpoint/restart
> capability).
> For the moment, benchmarks only ran on xeon platform but we expect more
> architecture soon. Besides the 'classic' benchs used, more network
> oriented benchs will be added. Netpipe between two virtual 'guests' for
> example. We hope we will be able to provide results concerning the
> virtual 'guest' scalability, running several 'guest' at the same time.
>
> Best regards,
>
>
> Le mercredi 07 juin 2006 à 16:20 +0200, Clement Calmels a écrit :
>
>> Hello !
>>
>> I'm part of a team of IBMers working on lightweight containers and we
>> are going to start a new test campaign. Candidates are vserver,
>> vserver context, namespaces (being pushed upstream), openvz, mcr (our
>> simple container dedicated to migration) and eventually xen.
>>
>> We will focus on the performance overhead but we are also interested in
>> checkpoint/restart and live migration. A last topic would be how well
>> the
>> resource managment criteria are met, but that's extra for the moment.
>>
>> We plan on measuring performance overhead by comparing the results on
>> a vanilla kernel with a partial and with a complete virtual
>> environment. By partial, we mean the patched kernel and a 'namespace'
>> virtualisation.
>>
>> Test tools
>> ----------
>> o For network performance :
>>

>> * netpipe (http://www.scl.ameslab.gov/netpipe/)
>> * netperf (http://www.netperf.org/netperf/NetperfPage.html)
>> * tbench (http://samba.org/ftp/tridge/dbench/README)
>>
>> o Filesystem :
>>
>>   * dbench (http://samba.org/ftp/tridge/dbench/README)
>>   * iozone (http://www.iozone.org/)
>>
>> o General
>>
>>   * kernbench (http://ck.kolivas.org/kernbench/) stress cpu and
>>   filesystem through kernel compilation
>>   * More 'real world' application could be used, feel free to submit
>>   candidates...
>>
>> We have experience on C/R and migration so we'll start with our own
>> scenario, migrating oracle under load. The load is generated by DOTS
>> (http://ltp.sourceforge.net/dotshowWe ran into trouble trying to run sto.php).
>>
>> If you could provided us some material on what has already been done :
>> URL, bench tools, scenarios. We'll try to compile them in. configuration
>> hints and tuning are most welcome if they are reasonable.
>>
>> Results, tools, scenarios will be published on lxc.sf.net . We will
>> set up the testing environment so as to be able to accept new
>> versions, patches, test tools and rerun the all on demand. Results,
>> tools, scenarios will be published on lxc.sf.net.
>>
>> thanks !
>>
>> Clement,
>>

Subject: RE: Container Test Campaign
Posted by mef on Mon, 03 Jul 2006 18:23:04 GMT

View Forum Message <> Reply to Message

Hi Kirill,

Thanks for the feedback. Not sure whether you are referring to Clement's work or our paper. I'll assume you are referring to our paper.

> >From what I see just just after 1 minute check of your results:
>
> DBench:
> - different disk I/O schedulers are compared. This makes

> comparison useless (not virtualization technologies are
> compared itself).

Correct. As you can image, it is not easy to pick benchmarks. The intention was simply to repeat measurements presented by the Xen paper published in 2003 Symposium of Operating Sytems Principles, and show how Vserver compares on those. The point of the paper is to show how "container based virtualization" compares to hypervisor based systems. We do not have the time nor expertise to show this with OpenVZ. It would be great if someon from the openvz community could step in and show how OpenVZ compares to Xen.

> - the fact that there is too much difference between measurements
> (e.g. vserver makes linux faster :lol:)

This is an interesting, odd, and likely laughable result. We just reported what we observed. It is quite possible that we made a mistake somewhere. However, I believe that the problem lies more with the dbench benchmark than with our setup. We did try to eliminate as many variables as possible. Please take a peek on the last page of the paper to see our discussion wrt normalizing the configurations. We are open to further suggestions to eliminate further variables.

> I also noticed that you do the measurements with different HZ settings.
> This influences the results as well...

Of course. My assumption is that it would negatively affect Vserver. Are you suggesting that it can positively affect the benchmark results to run at 1000HZ vs. 100HZ, as the Xen Domains are configured to do?

> BTW, do you plan to do functional testing in addition to performance?

Please clarify what you mean here? From what I gather, the main thing that Vserver lacks is the degree of network virtualization that OpenVZ supports. Is there anything else? From my perspective, the comparison will have to be with Xen/UML rather than, say, a contest between container based systems. I say this because it appears that the majority of the LKML community is believes that container-based systems don't add much above and beyond what Xen/UML/QEMU/VMware already offer today.

Best regards,
Marc

> -----Original Message-----
> From: Kirill Korotaev [mailto:dev@openvz.org]
> Sent: Monday, July 03, 2006 3:50 AM
> To: ? Calmels
> Cc: devel@openvz.org; vserver@list.linux-vserver.org;
> sam.vilain@catalyst.net.nz; serue@us.ibm.com; DLEZCANO@fr.ibm.com;
> mef@CS.Princeton.EDU
> Subject: Re: [Devel] Container Test Campaign
>
>

> >From what I see just just after 1 minute check of your results:
>
> DBench:
> - different disk I/O schedulers are compared. This makes
> comparison useless
>   (not virtualization technologies are compared itself).
> - the fact that there is too much difference between measurements
>   (e.g. vserver makes linux faster :lol:) makes me believe that
> you use large disk partiion,
>   where data blocks allocation on the disk influence your results.
>   To make these measurements correct the same partition with the
> size closest to the required
>   max disk space should be used in all DBench tests.
>
> TBench:
> - when running inside VE, please, make sure, that
> /proc/user_beancounters doesn't show you
>   resource allocation fails (failcnt column).
>   Resource limits set by default can be just too small to finish
> your test case.
>   And this doesn't mean your conclusion 'Concerning the results,
> obviously more isolation brings more overhead.'.
>   I'm really suprised to see such statements.
>
> I also noticed that you do the measurements with different HZ settings.
> This influences the results as well...
>
> BTW, do you plan to do functional testing in addition to performance?
>
> Thanks,
> Kirill
>
>
> > Hi,
> >
> > A first round about virtualisation benchmarks can be found here:
> > http://lxc.sourceforge.net/bench/
> > These benchmarks run with vanilla kernels and the patched versions of
> > well know virtualisation solutions: VServer and OpenVZ. Some benchs also
> > run inside the virtual 'guest' but we ran into trouble trying to run
> > some of them... probably virtual 'guest' configuration issues... we will
> > trying to fix them...
> > The metacluster migration solution (formely a Meiosys company produt)
> > was added as it seems that the checkpoint/restart topic is close to the
> > virtualisation's one (OpenVZ now provides a checkpoint/restart
> > capability).
> > For the moment, benchmarks only ran on xeon platform but we expect more
> > architecture soon. Besides the 'classic' benchs used, more network

> > oriented benchs will be added. Netpipe between two virtual 'guests' for
> > example. We hope we will be able to provide results concerning the
> > virtual 'guest' scalability, running several 'guest' at the same time.
> >
> > Best regards,
> >
> >
> > Le mercredi 07 juin 2006 à 16:20 +0200, Clement Calmels a écrit :
> >
> >>Hello !
> >>
> >>I'm part of a team of IBMers working on lightweight containers and we
> >>are going to start a new test campaign. Candidates are vserver,
> >>vserver context, namespaces (being pushed upstream), openvz, mcr (our
> >>simple container dedicated to migration) and eventually xen.
> >>
> >>We will focus on the performance overhead but we are also interested in
> >>checkpoint/restart and live migration. A last topic would be how well
> >>the
> >>resource managment criteria are met, but that's extra for the moment.
> >>
> >>We plan on measuring performance overhead by comparing the results on
> >>a vanilla kernel with a partial and with a complete virtual
> >>environment. By partial, we mean the patched kernel and a 'namespace'
> >>virtualisation.
> >>
> >>Test tools
> >>----------
> >>o For network performance :
> >>
> >> * netpipe (http://www.scl.ameslab.gov/netpipe/)
> >> * netperf (http://www.netperf.org/netperf/NetperfPage.html)
> >> * tbench (http://samba.org/ftp/tridge/dbench/README)
> >>
> >>o Filesystem :
> >>
> >> * dbench (http://samba.org/ftp/tridge/dbench/README)
> >> * iozone (http://www.iozone.org/)
> >>
> >>o General
> >>
> >> * kernbench (http://ck.kolivas.org/kernbench/) stress cpu and
> >>   filesystem through kernel compilation
> >> * More 'real world' application could be used, feel free to submit
> >>   candidates...
> >>
> >>We have experience on C/R and migration so we'll start with our own
> >>scenario, migrating oracle under load. The load is generated by DOTS

> >>(http://ltp.sourceforge.net/dotshowWe ran into trouble trying
> to run sto.php).
> >>
> >>If you could provided us some material on what has already been done :
> >>URL, bench tools, scenarios. We'll try to compile them in. configuration
> >>hints and tuning are most welcome if they are reasonable.
> >>
> >>Results, tools, scenarios will be published on lxc.sf.net . We will
> >>set up the testing environment so as to be able to accept new
> >>versions, patches, test tools and rerun the all on demand. Results,
> >>tools, scenarios will be published on lxc.sf.net.
> >>
> >>thanks !
> >>
> >>Clement,
> >>

Subject: Re: Container Test Campaign
Posted by Clement Calmels on Tue, 04 Jul 2006 09:42:50 GMT
View Forum Message <> Reply to Message

Hi,

> 1.1 It would be nice to run vmstat (say, vmstat 10) for the duration of
> the tests, and put the vmstat output logs to the site.

Our benchmark framework allows us to use oprofile during test...
couldn't it be better than vmstat?

> Basically, the detailed description of a process would be nice to have,
> in order to catch possible problems. There are a lot of tiny things
> which are influencing the results. For example, in linux kernels 2.4
> binding the NIC IRQ to a single CPU on an SMP system boosts network
> performance by about 15%! Sure this is not relevant here, it's just an
> example.

I agree. Actually, I always try to use 'default' configuration or
installation but I will try to describe the tests in details.

> 1.3 Would be nice to have diffs between different kernel configs.

The different configs used are available in the lxc site. You will
notice that I used a minimal config file for most of the test, but for
Openvz I had to use the one I found in the OpenVZ site because I faced
kernel build error (some CONFIG_NET... issues). I think that the
differences are more dealing with network stuff.

> For example, the tbench test is probably failed to finish because it
> hits the limits for privvmpages, tcpsndbuf and tcprcvbuf. I have
> increased the limits for those parameters and the test was finished
> successfully. Also, dbench test could hit the disk quota limit for a VE.
> Some more info is available at http://wiki.openvz.org/Resource_management

I already used this page. I had to increase 'diskinodes' and 'diskspace'
resources in order to run some test properly (the disk errors were more
selfexplicit).
I'm wondering why a default 'guest' creation implies some resources
restrictions? Couldn't the resources be unlimited? I understand the need
for resource management, but the default values look a little bit
tiny...

> 2.2 For OpenVZ specifically, it would be nice to collect
> /proc/user_beancounters output before and after the test.

For sure... I will take a look at how integrating it in our automatic
test environment.

Best regards,

--
Clément Calmels <clement.calmels@fr.ibm.com>

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by Clement Calmels on Tue, 04 Jul 2006 11:10:11 GMT

View Forum Message <> Reply to Message

Hi,

> from the tests:
> "For benchs inside real 'guest' nodes (OpenVZ/VServer) you should
>   take into account that the FS tested is not the 'host' node one's."
>
> at least for Linux-VServer it should not be hard to avoid the
> chroot/filesystem namespace part and have it run on the host fs.
> a bind mount into the guest might do the trick too, if you need
> help to accomplish that, just let me know ...

For the moment I just use the chcontext command to get rid of filesystem
part. But even if the tested filesystem is not the host filesystem, I
just keep in mind that all applications running inside a 'guest' will
use _this_ filesystem and not the host one.
>From what I understand about VServer, it looks flexible enougth to let
us test different 'virtualisation' parts. A 'guest' looks like a stack
of different 'virtualisation' layers (chcontext + ipv4root + chroot).

But it's not the case for all solutions.

Best regards,

--
Clément Calmels <clement.calmels@fr.ibm.com>

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by kir on Tue, 04 Jul 2006 12:19:02 GMT

View Forum Message <> Reply to Message

Clément,

Thanks for addressing my concerns! See comments below.

Clément Calmels wrote:
> Hi,
>
>
>> 1.1 It would be nice to run vmstat (say, vmstat 10) for the duration of
>> the tests, and put the vmstat output logs to the site.
>>
>
> Our benchmark framework allows us to use oprofile during test...
> couldn't it be better than vmstat?
>
Good idea.
>> Basically, the detailed description of a process would be nice to have,
>> in order to catch possible problems. There are a lot of tiny things
>> which are influencing the results. For example, in linux kernels 2.4
>> binding the NIC IRQ to a single CPU on an SMP system boosts network
>> performance by about 15%! Sure this is not relevant here, it's just an
>> example.
>>
>
> I agree. Actually, I always try to use 'default' configuration or
> installation but I will try to describe the tests in details.
>
>> 1.3 Would be nice to have diffs between different kernel configs.
>>
> The different configs used are available in the lxc site. You will
> notice that I used a minimal config file for most of the test, but for
> Openvz I had to use the one I found in the OpenVZ site because I faced
> kernel build error (some CONFIG_NET... issues).
We are trying to eliminate those, so a bug report would be nice.
> I think that the
> differences are more dealing with network stuff.

>
>> For example, the tbench test is probably failed to finish because it
>> hits the limits for privvmpages, tcpsndbuf and tcprcvbuf. I have
>> increased the limits for those parameters and the test was finished
>> successfully. Also, dbench test could hit the disk quota limit for a VE.
>> Some more info is available at http://wiki.openvz.org/Resource_management
>>
>
> I already used this page. I had to increase 'diskinodes' and 'diskspace'
> resources in order to run some test properly (the disk errors were more
> selfexplicit).
> I'm wondering why a default 'guest' creation implies some resources
> restrictions? Couldn't the resources be unlimited? I understand the need
> for resource management, but the default values look a little bit
> tiny...
>

The reason is security. A guest is untrusted by default, though sane
limits are applied. Same as ulimit which has some sane defaults (check
output of ulimit -a). Same as those kernel settings from /proc/sys --
should /proc/sys/fs/file-max be 'unlimited' by default?

In fact, those limits are taken from a sample configuration file during
"vzctl create" stage. Sample file is specified in global OpenVZ config
file (/etc/vz/vz.conf, parameter name is CONFIGFILE, default is to take
configuration from /etc/vz/conf/ve-vps.basic.conf-sample).

There are several ways to change that default configuration:

1. (globally) Put another sample config and specify it in /etc/vz/vz.conf
2. (globally) Edit the existing sample config
(/etc/vz/conf/ve-vps.basic.conf-sample)
3. (per VE) Specify another config during vzctl create stage, like this:
vzctl create VEID [--config name]
4. (per VE) Tune the specific parameters using vzctl set [--param value
...] --save

>
>> 2.2 For OpenVZ specifically, it would be nice to collect
>> /proc/user_beancounters output before and after the test.
>>
>
> For sure... I will take a look at how integrating it in our automatic
> test environment.
>
> Best regards,
>
>

Subject: Re: Container Test Campaign
Posted by Clement Calmels on Tue, 04 Jul 2006 12:34:42 GMT

View Forum Message <> Reply to Message

Hi,

Sorry, just forgot one part of your email...

> 1.2 Can you tell how you run the tests. I am particularly interested in
> - how many iterations do you do?
> - what result do you choose from those iterations?
> - how reproducible are the results?
> - are you rebooting the box between the iterations?
> - are you reformatting the partition used for filesystem testing?
> - what settings are you using (such as kernel vm params)?
> - did you stop cron daemons before running the test?
> - are you using the same test binaries across all the participants?
> - etc. etc...

A basic test looks like:

--
Clément Calmels <clement.calmels@fr.ibm.com>

Subject: Re: Container Test Campaign
Posted by Clement Calmels on Tue, 04 Jul 2006 13:02:54 GMT

View Forum Message <> Reply to Message

Hi,

Sorry, I just forgot one part of your email... (and sorry for the mail
spamming, I probably got too big fingers or too tiny keyboard)

> 1.2 Can you tell how you run the tests. I am particularly interested in
> - how many iterations do you do?
> - what result do you choose from those iterations?
> - how reproducible are the results?
> - are you rebooting the box between the iterations?
> - are you reformatting the partition used for filesystem testing?
> - what settings are you using (such as kernel vm params)?
> - did you stop cron daemons before running the test?
> - are you using the same test binaries across all the participants?
> - etc. etc...

A basic 'patch' test looks like:
o build the appropriate kernel (2.6.16-026test014-x86_64-smp for
example)

o reboot
o run dbench on /tmp with 8 processes
o run tbench with 8 processes
o run lmbench
o run kernbench

For test inside a 'guest' I just do something like:
o build the appropriate kernel (2.6.16-026test014-x86_64-smp for example)
o reboot
o build the utilities (vztcl+vzquota for example)
o reboot
o launch a guest
o run in the guest dbench ...
o run in the guest tbench ...
....

-The results are the average value of several iterations of each set of these kind of tests. I will try to update the site with the numbers of iterations behind each values.
- For the filesystem testing, the partition is not reformatted. I can change this behaviour...
- For the settings of the guest I tried to use the default settings (I had to change some openvz guest settings) just following the HOWTO on vserver or openvz site.
For the kernel parameters, did you mean kernel config file tweaking?
- Cron are stopped during tests.
- All binaries are always build in the test node.

Feel free to provide me different scenario which you think are more relevant.

--
Clément Calmels <clement.calmels@fr.ibm.com>

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by Clement Calmels on Tue, 04 Jul 2006 13:54:09 GMT
View Forum Message <> Reply to Message

Hi,

> > I'm wondering why a default 'guest' creation implies some resources
> > restrictions? Couldn't the resources be unlimited? I understand the need
> > for resource management, but the default values look a little bit
> > tiny...
> >
> The reason is security. A guest is untrusted by default, though sane

> limits are applied. Same as ulimit which has some sane defaults (check
> output of ulimit -a). Same as those kernel settings from /proc/sys --
> should /proc/sys/fs/file-max be 'unlimited' by default?

Ok. So as our benchmarks have no security concern, you will see no
objection if I set all the parameters in the 'guest' to their value in
the host, won't you?

--
Clément Calmels <clement.calmels@fr.ibm.com>

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by dev on Tue, 04 Jul 2006 14:32:04 GMT

View Forum Message <> Reply to Message

>>from the tests:
>> "For benchs inside real 'guest' nodes (OpenVZ/VServer) you should
>> take into account that the FS tested is not the 'host' node one's."
>>
>>at least for Linux-VServer it should not be hard to avoid the
>>chroot/filesystem namespace part and have it run on the host fs.
>>a bind mount into the guest might do the trick too, if you need
>>help to accomplish that, just let me know ...
>
>
> For the moment I just use the chcontext command to get rid of filesystem
> part. But even if the tested filesystem is not the host filesystem, I
> just keep in mind that all applications running inside a 'guest' will
> use _this_ filesystem and not the host one.
>>From what I understand about VServer, it looks flexible enougth to let
> us test different 'virtualisation' parts. A 'guest' looks like a stack
> of different 'virtualisation' layers (chcontext + ipv4root + chroot).
> But it's not the case for all solutions.

For OpenVZ it is also possible to test different subsytems separately (virtualization/isolation,
resource management, disk quota, CPU scheduler).
I would notice also, that in OpenVZ all these features are ON by default.

So I probably miss something, but why we test other technologies in modes when not
all the features are ON? in this case we compare not the real overhead,
but the one minimized for this concrete benchmark. It's just like comparing with
Xen Domain0 which doesn't have any overhead, but not because it is a good technology, but
rather because it doesn't do anything valuable.

BTW, comparing with Xen would be interesting as well. Just to show the difference.

Thanks,

Kirill

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by kir on Tue, 04 Jul 2006 14:52:07 GMT

View Forum Message <> Reply to Message

Clément Calmels wrote:
> Hi,
>
>
>>> I'm wondering why a default 'guest' creation implies some resources
>>> restrictions? Couldn't the resources be unlimited? I understand the need
>>> for resource management, but the default values look a little bit
>>> tiny...
>>>
>>>
>> The reason is security. A guest is untrusted by default, though sane
>> limits are applied. Same as ulimit which has some sane defaults (check
>> output of ulimit -a). Same as those kernel settings from /proc/sys --
>> should /proc/sys/fs/file-max be 'unlimited' by default?
>>
>
> Ok. So as our benchmarks have no security concern, you will see no
> objection if I set all the parameters in the 'guest' to their value in
> the host, won't you?
Sure.

In case you are testing performance (but not, say, isolation), you can definitely set all the UBCs to unlimited values (i.e. both barrier and limit for each parameter should be set to MAX_LONG). The only issues is with vmguarpages parameter, because this is a guarantee but not limit -- but unless you are doing something weird it should be OK to set to to MAX_LONG as well.

Another approach is to generate sample config (for the given server) using vzsplit utility with the number of VEs set to 1, like this:
```
# vzsplit -f one-ve -n 1 [-s xxx]
```
and use it for new VE creation:
```
# vzctl create 123 --config one-ve
```

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by Cedric Le Goater on Tue, 04 Jul 2006 15:28:53 GMT

View Forum Message <> Reply to Message

Kirill Korotaev wrote:

> For OpenVZ it is also possible to test different subsytems separately
> (virtualization/isolation, resource management, disk quota, CPU scheduler).
> I would notice also, that in OpenVZ all these features are ON by default.

hmm, we didn't realize that. Good, it will make the results even more relevant. Any pointers on how these light virtualized/isolated environment, similar to vserver chontext, can be set up ?

> So I probably miss something, but why we test other technologies in
> modes when not all the features are ON ? in this case we compare not
> the real overhead, but the one minimized for this concrete benchmark.

you must know how much time it takes to set up such an environment. it's just a beginning. Clement is currently running tests on openvz stable, full vserver and adding new tests on dbench to take into account your remarks.

we want to add tests and also namespace patchsets to the matrix. If you have some time, please send us some of your material, that'll be nice.

> It's just like comparing with Xen Domain0 which doesn't have any overhead,
> but not because it is a good technology, but rather because it doesn't do
> anything valuable.

Xen Domain0 has overhead.

> BTW, comparing with Xen would be interesting as well. Just to show the
> difference.

We've done that in the OLS paper, soon to come, but we'll add the results to lxc.sf.net before ols. You can also check this paper :

http://list.linux-vserver.org/archive/vserver/msg13234.html

thanks,

C.

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by Cedric Le Goater on Tue, 04 Jul 2006 15:45:23 GMT
View Forum Message <> Reply to Message

Kir Kolyshkin wrote:

> In case you are testing performance (but not, say, isolation), you can
> definitely set all the UBCs to unlimited values (i.e. both barrier and
> limit for each parameter should be set to MAX_LONG). The only issues is

> with vmguarpages parameter, because this is a guarantee but not limit --
> but unless you are doing something weird it should be OK to set to to
> MAX_LONG as well.

that's something we're interested in also. How well the isolation criteria
are met ? This will be the next campaign.

do you happen to have some material in the field ? anything to contribute
to LTP ? that would be useful to test the patchsets we are all trying to
push in -mm and mainline.

thanks,

C.

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by kir on Tue, 04 Jul 2006 16:30:26 GMT

View Forum Message <> Reply to Message

See my comments below.

In general - please don't get the impression I try to be fastidious. I'm
just trying to help you create a system in which results can be
reproducible and trusted. There are a lot of factors that influence the
performance; some of those are far from being obvious.

Clément Calmels wrote:
> A basic 'patch' test looks like:
> o build the appropriate kernel (2.6.16-026test014-x86_64-smp for
> example)
> o reboot
> o run dbench on /tmp with 8 processes
>
IMO you should add a reboot here, in between _different_ tests, just
because previous tests should not influence the following ones.
Certainly you do not need a reboot before iterations of the same test.
> o run tbench with 8 processes
> o run lmbench
> o run kernbench
>
> For test inside a 'guest' I just do something like:
> o build the appropriate kernel (2.6.16-026test014-x86_64-smp for
> example)
> o reboot
>
Here you do not have to reboot. OpenVZ tools does not require OpenVZ
kernel to be built.

> o build the utilities (vztcl+vzquota for example)
> o reboot
> o launch a guest
>
Even this part is tricky! You haven't specified whether you create the guest before or after the reboot. Let me explain.

If you create a guest before the reboot, the performance (at least at the first iteration) could be a bit higher than if you create a guest after the reboot. The reason is in the second case the buffer cache will be filled with OS template data (which is several hundred megs).
> o run in the guest dbench ...
>
Again, a clean reboot is needed IMO.
> o run in the guest tbench ...
> ....
>
> -The results are the average value of several iterations of each set of
> these kind of tests.
Hope you do not recompile the kernels before the iterations (just to speed things up).
> I will try to update the site with the numbers of
> iterations behind each values.
>
Would be great to have that data (as well as the results of the individual iterations, and probably graphs for the individual iterations -- to see the "warming" progress, discrepancy between iterations, degradation over iterations (if that takes place) etc).

Based on that data, one can decide to further tailor the testing process. For example, if there are visible signs of "warming" for a first few iterations (i.e. the performance is worse) it makes sense to unconditionally exclude those from the results. If there is a sign of degradation, something is wrong. And so on...
> - For the filesystem testing, the partition is not reformatted. I can
> change this behaviour...
>
Disk layout is influencing the results of the test which do heavy I/O. Just a single example: if you try to test the performance of a web server, results will decrease over time. The reason of degradation is ... web server's access_log file! It grows over time, and write operation takes a bit longer (due to several different reasons).

The same will happen with most of the other tests involving I/O. Thus, test results will be non-accurate. To achieve more accuracy and exclude the impact of the disk and filesystem layout to the results, you should reformat the partition you use for testing each time before the test. Note that you don't have to reinstall everything from scratch -- just

use a separate partition (mounted to say /mnt/temptest) and make sure
most of the I/O during the test happens on that partition.
> - For the settings of the guest I tried to use the default settings (I
> had to change some openvz guest settings) just following the HOWTO on
> vserver or openvz site.
> For the kernel parameters, did you mean kernel config file tweaking?
>
No I mean those params from /proc/sys (== /etc/sysctl.conf). For
example, if you want networking for OpenVZ guests, you have to turn on
ip_forwarding. There are some params affecting network performance, such
as various gc_thresholds. For the big number of guests, you have to tune
some system-wide parameters as well.

So I am leading to the proposition that all such changes should be
documented in test results.
> - Cron are stopped during tests.
>
Hope you do that for the guest as well... :)
> - All binaries are always build in the test node.
>
I assuming you are doing your tests on the same system (i.e. same
compiler/libs/whatever else), and you do not change that system over
time (i.e. you do not upgrade gcc on it in between the tests).
> Feel free to provide me different scenario which you think are more
> relevant.

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by Clement Calmels on Wed, 05 Jul 2006 07:40:28 GMT
View Forum Message <> Reply to Message

Hi,

> In general - please don't get the impression I try to be fastidious. I'm
> just trying to help you create a system in which results can be
> reproducible and trusted. There are a lot of factors that influence the
> performance; some of those are far from being obvious.

Don't get me wrong I'm looking for such remarks :)

> IMO you should add a reboot here, in between _different_ tests, just
> because previous tests should not influence the following ones.
> Certainly you do not need a reboot before iterations of the same test.

I don't do this first because I didn't want to get test nodes wasting
their time rebooting instead of running test. What do you think of
something like this:
o reboot

o run dbench (or wathever) X times
o reboot

> > For test inside a 'guest' I just do something like:
> > o build the appropriate kernel (2.6.16-026test014-x86_64-smp for
> > example)
> > o reboot
> >
> Here you do not have to reboot. OpenVZ tools does not require OpenVZ
> kernel to be built.

You got me... I was still believing the VZKERNEL_HEADERS variable was
needed. Things have changed since vzctl 3.0.0-4...

> > o build the utilities (vztcl+vzquota for example)
> > o reboot
> > o launch a guest
> >
> Even this part is tricky! You haven't specified whether you create the
> guest before or after the reboot. Let me explain.
>
> If you create a guest before the reboot, the performance (at least at
> the first iteration) could be a bit higher than if you create a guest
> after the reboot. The reason is in the second case the buffer cache will
> be filled with OS template data (which is several hundred megs).
can
I can split the "launch a guest" part into 2 parts:
o guest creation
o reboot
o guest start-up
Do you feel comfortable with that?

> > -The results are the average value of several iterations of each set of
> > these kind of tests.
> Hope you do not recompile the kernels before the iterations (just to
> speed things up).
> > I will try to update the site with the numbers of
> > iterations behind each values.
> >
> Would be great to have that data (as well as the results of the
> individual iterations, and probably graphs for the individual iterations
> -- to see the "warming" progress, discrepancy between iterations,
> degradation over iterations (if that takes place) etc).

I will try to get/show those datas.

> The same will happen with most of the other tests involving I/O. Thus,
> test results will be non-accurate. To achieve more accuracy and exclude

> the impact of the disk and filesystem layout to the results, you should
> reformat the partition you use for testing each time before the test.
> Note that you don't have to reinstall everything from scratch -- just
> use a separate partition (mounted to say /mnt/temptest) and make sure
> most of the I/O during the test happens on that partition.

It would be possible for 'host' node... inside the 'guest' node, I don't
know if it makes sense. Just adding an 'external' partition to the
'guest' for I/O test purpose? For example in an OpenVZ guest, creating a
new and empty simfs partition in order to run test on it?

> > - For the settings of the guest I tried to use the default settings (I
> > had to change some openvz guest settings) just following the HOWTO on
> > vserver or openvz site.
> > For the kernel parameters, did you mean kernel config file tweaking?
> >
> No I mean those params from /proc/sys (== /etc/sysctl.conf). For
> example, if you want networking for canOpenVZ guests, you have to turn on
> ip_forwarding. There are some params affecting network performance, such
> as various gc_thresholds. For the big number of guests, you have to tune
> some system-wide parameters as well.

For the moment, I just follow the available documentation:
 http://wiki.openvz.org/Quick_installation#Configuring_sysctl _settings
Do you think these paramenters can hardly affect network performance?
>From what I understand lot of them are needed.

> > - All binaries are always build in the test node.
> >
> I assuming you are doing your tests on the same system (i.e. same
> compiler/libs/whatever else), and you do not change that system over
> time (i.e. you do not upgrade gcc on it in between the tests).

I hope! :)
--
Clément Calmels <clement.calmels@fr.ibm.com>

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by kir on Wed, 05 Jul 2006 08:34:24 GMT

View Forum Message <> Reply to Message

Clément Calmels wrote:
> What do you think of
> something like this:
> o reboot
> o run dbench (or wathever) X times
> o reboot

>
Perfectly fine with me.
>> Here you do not have to reboot. OpenVZ tools does not require OpenVZ
>> kernel to be built.
>>
>
> You got me... I was still believing the VZKERNEL_HEADERS variable was
> needed. Things have changed since vzctl 3.0.0-4..
Yes, we get rid off that dependency, to ease the external packages
maintenance.
> can I can split the "launch a guest" part into 2 parts:
> o guest creation
> o reboot
> o guest start-up
> Do you feel comfortable with that?
>
Perfectly fine. Same scenario applies to other cases: the rule of thumb
is if your test preparation involves a lot of I/O, you'd better reboot
in between preparation and the actual test.
>> The same will happen with most of the other tests involving I/O. Thus,
>> test results will be non-accurate. To achieve more accuracy and exclude
>> the impact of the disk and filesystem layout to the results, you should
>> reformat the partition you use for testing each time before the test.
>> Note that you don't have to reinstall everything from scratch -- just
>> use a separate partition (mounted to say /mnt/temptest) and make sure
>> most of the I/O during the test happens on that partition.
>>
> It would be possible for 'host' node... inside the 'guest' node, I don't
> know if it makes sense. Just adding an 'external' partition to the
> 'guest' for I/O test purpose? For example in an OpenVZ guest, creating a
> new and empty simfs partition in order to run test on it?
>
simfs is not a real filesystem, it is kinda 'pass-though' fake FS which
works on top of a real FS (like ext2 or ext3).
So, in order to have a new fresh filesystem for guests, you can create
some disk partition, mkfs and mount it to /vz. If you want to keep
templates, just
change the TEMPLATE variable in /etc/vz/vz.conf from /vz/template to
something outside of /vz. There are other ways possible, and I think the
same applies to VServer.
>>> - For the settings of the guest I tried to use the default settings (I
>>> had to change some openvz guest settings) just following the HOWTO on
>>> vserver or openvz site.
>>> For the kernel parameters, did you mean kernel config file tweaking?
>>>
>>>
>> No I mean those params from /proc/sys (== /etc/sysctl.conf). For
>> example, if you want networking for canOpenVZ guests, you have to turn on

>> ip_forwarding. There are some params affecting network performance, such
>> as various gc_thresholds. For the big number of guests, you have to tune
>> some system-wide parameters as well.
>>
>
> For the moment, I just follow the available documentation:
> http://wiki.openvz.org/Quick_installation#Configuring_sysctl _settings
> Do you think these paramenters can hardly affect network performance?
> From what I understand lot of them are needed.
>
OK. Still, such stuff should be documented on the test results pages.

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by dev on Wed, 05 Jul 2006 10:43:17 GMT
View Forum Message <> Reply to Message

>>>- All binaries are always build in the test node.
>>>
>>
>>I assuming you are doing your tests on the same system (i.e. same
>>compiler/libs/whatever else), and you do not change that system over
>>time (i.e. you do not upgrade gcc on it in between the tests).
>
>
> I hope! :)

All binaries should be built statically to work the same way inside host/guest or
you need to make sure that you have exactly the same versions of glibc and other
system libraries. At least glibc can affect perforamnce very much :/

Kirill

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by Herbert Poetzl on Wed, 05 Jul 2006 12:13:40 GMT
View Forum Message <> Reply to Message

On Tue, Jul 04, 2006 at 06:32:04PM +0400, Kirill Korotaev wrote:
>>> from the tests:
>>> "For benchs inside real 'guest' nodes (OpenVZ/VServer) you should
>>> take into account that the FS tested is not the 'host' node one's."
>>>
>>> at least for Linux-VServer it should not be hard to avoid the
>>> chroot/filesystem namespace part and have it run on the host fs.
>>> a bind mount into the guest might do the trick too, if you need
>>> help to accomplish that, just let me know ...

>> For the moment I just use the chcontext command to get rid of
>> filesystem part. But even if the tested filesystem is not the host
>> filesystem, I just keep in mind that all applications running inside
>> a 'guest' will use _this_ filesystem and not the host one.

>> From what I understand about VServer, it looks flexible enougth to let
>> us test different 'virtualisation' parts. A 'guest' looks like a stack
>> of different 'virtualisation' layers (chcontext + ipv4root + chroot).
>> But it's not the case for all solutions.

> For OpenVZ it is also possible to test different subsytems separately
> (virtualization/isolation, resource management, disk quota, CPU
> scheduler).

> I would notice also, that in OpenVZ all these features are ON by
> default.

which is the same as for a complete guest setup, what I
think (and you already mentioned too, IIRC) is that it is
very important to have identical test setups with and
without virtualization enabled, which means that the
following conditions are met:

 - filesystem and involved devices are identical
 - used resources and limits are identical/very close
 - number of processes and cache state as close as possible
 - kernel/system state as close as possible

> So I probably miss something, but why we test other technologies in
> modes when not all the features are ON?

doesn't make much sense, and is not what I actually
suggested in the first place, it might be interesting
to narrow down possible issues by putting together
a complete 'stack' slice by slice if that allows to
remove a single slice from that equation

> in this case we compare not the real overhead, but the one minimized
> for this concrete benchmark.

actually all cases are interesting as none of them
is supposed to add measurable overhead which is also
true for the whole thing which is at least the sum
of all of them

HTC,
Herbert

> It's just like comparing with Xen Domain0 which doesn't have any
> overhead, but not because it is a good technology, but rather because
> it doesn't do anything valuable.
>
> BTW, comparing with Xen would be interesting as well. Just to show the
> difference.
>
> Thanks,
> Kirill

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by Cedric Le Goater on Wed, 05 Jul 2006 12:31:59 GMT

View Forum Message <> Reply to Message

Clément Calmels wrote:

> I don't do this first because I didn't want to get test nodes wasting
> their time rebooting instead of running test. What do you think of
> something like this:
> o reboot
> o run dbench (or wathever) X times
> o reboot

[...]

> I can split the "launch a guest" part into 2 parts:
> o guest creation
> o reboot
> o guest start-up
> Do you feel comfortable with that?

we need to add a methodology page or similar on

 http://lxc.sourceforge.net/bench/

first page is a bit rough for the moment.

>>> -The results are the average value of several iterations of each set of
>>> these kind of tests.
>> Hope you do not recompile the kernels before the iterations (just to
>> speed things up).
>>> I will try to update the site with the numbers of
>>> iterations behind each values.
>>>
>> Would be great to have that data (as well as the results of the
>> individual iterations, and probably graphs for the individual iterations

>> -- to see the "warming" progress, discrepancy between iterations,
>> degradation over iterations (if that takes place) etc).
>
> I will try to get/show those datas.

this data is already rougly available :

http://lxc.sourceforge.net/bench/r3/dbenchraw
http://lxc.sourceforge.net/bench/r3/tbenchraw
etc.

is that what you are thinking about ?

>>> - All binaries are always build in the test node.
>>>
>> I assuming you are doing your tests on the same system (i.e. same
>> compiler/libs/whatever else), and you do not change that system over
>> time (i.e. you do not upgrade gcc on it in between the tests).
>
> I hope! :)

all host nodes are described here :

http://lxc.sourceforge.net/bench/r3/r3.html

may be add the list of installed packages ?

thanks,

C.

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by Cedric Le Goater on Wed, 05 Jul 2006 13:06:20 GMT

View Forum Message <> Reply to Message

Kirill Korotaev wrote:
>>>> - All binaries are always build in the test node.
>>>>
>>>
>>> I assuming you are doing your tests on the same system (i.e. same
>>> compiler/libs/whatever else), and you do not change that system over
>>> time (i.e. you do not upgrade gcc on it in between the tests).
>>
>>
>> I hope! :)
>
> All binaries should be built statically to work the same way inside

> host/guest or
> you need to make sure that you have exactly the same versions of glibc
> and other
> system libraries. At least glibc can affect perforamnce very much :/

yep. we could add a test line with the static versions.

C.

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by Herbert Poetzl on Wed, 05 Jul 2006 14:02:12 GMT
View Forum Message <> Reply to Message

On Tue, Jul 04, 2006 at 04:19:02PM +0400, Kir Kolyshkin wrote:

[lot of stuff zapped here]

> >The different configs used are available in the lxc site. You will
> >notice that I used a minimal config file for most of the test, but for
> >Openvz I had to use the one I found in the OpenVZ site because I faced
> >kernel build error (some CONFIG_NET... issues).
> We are trying to eliminate those, so a bug report would be nice.

this might be some help here ...

 http://plm.osdl.org/plm-cgi/plm?module=patch_info&patch_ id=5108
 http://plm.osdl.org/plm-cgi/plm?module=patch_info&patch_ id=5109

HTH,
Herbert

[zapped even more]

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by Gerrit Huizenga on Wed, 05 Jul 2006 22:37:45 GMT
View Forum Message <> Reply to Message

On Wed, 05 Jul 2006 14:43:17 +0400, Kirill Korotaev wrote:
> >>>- All binaries are always build in the test node.
> >>>
> >>
> >>I assuming you are doing your tests on the same system (i.e. same
> >>compiler/libs/whatever else), and you do not change that system over
> >>time (i.e. you do not upgrade gcc on it in between the tests).
> >

> >
> > I hope! :)
>
> All binaries should be built statically to work the same way inside host/guest or
> you need to make sure that you have exactly the same versions of glibc and other
> system libraries. At least glibc can affect perforamnce very much :/

Ick - no one builds binaries statically in the real world. And,
when you build binaries statically, you lose all ability to fix
security problems in base libraries by doing an update of that library.
Instead, all applications need to be rebuilt.

Performance tests should reflect real end user usage - not contrived
situations that make a particular solution look better or worse.
If glibc can affect performance, that should be demonstrated in the
real performance results - it is part of the impact of the solution and
may need an additional solution or discussion.

gerrit

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by dev on Thu, 06 Jul 2006 10:44:23 GMT

View Forum Message <> Reply to Message

Gerrit,

>>>>I assuming you are doing your tests on the same system (i.e. same
>>>>compiler/libs/whatever else), and you do not change that system over
>>>>time (i.e. you do not upgrade gcc on it in between the tests).
>>>
>>>
>>>I hope! :)
>>
>>All binaries should be built statically to work the same way inside host/guest or
>>you need to make sure that you have exactly the same versions of glibc and other
>>system libraries. At least glibc can affect perforamnce very much :/
>
>
> Ick - no one builds binaries statically in the real world. And,
> when you build binaries statically, you lose all ability to fix
> security problems in base libraries by doing an update of that library.
> Instead, all applications need to be rebuilt.
>
> Performance tests should reflect real end user usage - not contrived
> situations that make a particular solution look better or worse.
> If glibc can affect performance, that should be demonstrated in the
> real performance results - it is part of the impact of the solution and

> may need an additional solution or discussion.
What I tried to say is that performance results done in different
environments are not comparable so have no much meaning. I don't want us
to waste our time digging in why one environment is a bif faster or slower than another.
I hope you don't want too.

Now, to have the same environment there are at least 2 ways:
- make static binaries (not that good, but easiest way)
- have exactly the same packages in host/VPS for all test cases.

BTW, I also prefer 2nd way, but it is harder.

Thanks,
Kirill

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by Herbert Poetzl on Thu, 06 Jul 2006 10:51:31 GMT

View Forum Message <> Reply to Message

> Hi,
>
> Sorry, I just forgot one part of your email... (and sorry for the mail
> spamming, I probably got too big fingers or too tiny keyboard)
>
> > 1.2 Can you tell how you run the tests. I am particularly interested in
> > - how many iterations do you do?
> > - what result do you choose from those iterations?
> > - how reproducible are the results?
> > - are you rebooting the box between the iterations?
> > - are you reformatting the partition used for filesystem testing?
> > - what settings are you using (such as kernel vm params)?
> > - did you stop cron daemons before running the test?
> > - are you using the same test binaries across all the participants?
> > - etc. etc...
>
> A basic 'patch' test looks like:
> o build the appropriate kernel (2.6.16-026test014-x86_64-smp for
> example)
> o reboot
> o run dbench on /tmp with 8 processes

sidenote: on a 'typical' Linux-VServer guest, tmp
will be mounted as tmpfs, so be careful with that
OVZ might do similar as might your host distro :)

HTH,

Herbert

> o run tbench with 8 processes
> o run lmbench
> o run kernbench
>
> For test inside a 'guest' I just do something like:
> o build the appropriate kernel (2.6.16-026test014-x86_64-smp for
> example)
> o reboot
> o build the utilities (vztcl+vzquota for example)
> o reboot
> o launch a guest
> o run in the guest dbench ...
> o run in the guest tbench ...
> ....
>
> -The results are the average value of several iterations of each set of
> these kind of tests. I will try to update the site with the numbers of
> iterations behind each values.
> - For the filesystem testing, the partition is not reformatted. I can
> change this behaviour...
> - For the settings of the guest I tried to use the default settings (I
> had to change some openvz guest settings) just following the HOWTO on
> vserver or openvz site.
> For the kernel parameters, did you mean kernel config file tweaking?
> - Cron are stopped during tests.
> - All binaries are always build in the test node.
>
> Feel free to provide me different scenario which you think are more
> relevant.
>
> --

>
> ____
> Vserver mailing list
> Vserver@list.linux-vserver.org
> http://list.linux-vserver.org/mailman/listinfo/vserver

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by Herbert Poetzl on Thu, 06 Jul 2006 10:54:58 GMT

View Forum Message <> Reply to Message

On Wed, Jul 05, 2006 at 02:43:17PM +0400, Kirill Korotaev wrote:
> >>>- All binaries are always build in the test node.
> >>>

> >>
> >>I assuming you are doing your tests on the same system (i.e. same
> >>compiler/libs/whatever else), and you do not change that system over
> >>time (i.e. you do not upgrade gcc on it in between the tests).
> >
> >
> >I hope! :)
>
> All binaries should be built statically to work the same way inside

I'm against that, IMHO statically built binaries (except
for dietlibc and uClibc) are not really realistic

> host/guest or you need to make sure that you have exactly the same
> versions of glibc and other system libraries. At least glibc can
> affect perforamnce very much :/

yep, indeed, I'd suggest to use the very same filesystem
for tests on the host as you use for the guests ...

best,
Herbert

> Kirill
> ____
> Vserver mailing list
> Vserver@list.linux-vserver.org
> http://list.linux-vserver.org/mailman/listinfo/vserver

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by dev on Thu, 06 Jul 2006 11:30:47 GMT

View Forum Message <> Reply to Message

Herbert Poetzl wrote:
> sidenote: on a 'typical' Linux-VServer guest, tmp
> will be mounted as tmpfs, so be careful with that
> OVZ might do similar as might your host distro :)
>
good point. Can we document all these issues somewhere?

Kirill

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by Gerrit Huizenga on Thu, 06 Jul 2006 14:12:28 GMT

View Forum Message <> Reply to Message

On Thu, 06 Jul 2006 14:44:23 +0400, Kirill Korotaev wrote:
> Gerrit,
>
> >>>>I assuming you are doing your tests on the same system (i.e. same
> >>>>compiler/libs/whatever else), and you do not change that system over
> >>>>time (i.e. you do not upgrade gcc on it in between the tests).
> >>>
> >>>I hope! :)
> >>
> >>All binaries should be built statically to work the same way inside host/guest or
> >>you need to make sure that you have exactly the same versions of glibc and other
> >>system libraries. At least glibc can affect perforamnce very much :/
> >
> >
> > Ick - no one builds binaries statically in the real world. And,
> > when you build binaries statically, you lose all ability to fix
> > security problems in base libraries by doing an update of that library.
> > Instead, all applications need to be rebuilt.
> >
> > Performance tests should reflect real end user usage - not contrived
> > situations that make a particular solution look better or worse.
> > If glibc can affect performance, that should be demonstrated in the
> > real performance results - it is part of the impact of the solution and
> > may need an additional solution or discussion.
>
> What I tried to say is that performance results done in different
> environments are not comparable so have no much meaning. I don't want us
> to waste our time digging in why one environment is a bif faster or slower than another.
> I hope you don't want too.

I *do* want to understand why one patch set or another is significantly
faster or slower than any other. I think by now everyone realizes that
what goes into mainline will not be some slice of vserver, or OpenVZ
or MetaCluster or Eric's work in progress. It will be the convergance
of the patches that enable all solutions, and those patches will be added
as they are validated as beneficial to all participants *and* beneficial
(or not harmful) to mainline Linux. So, testing of large environments
is good to see where the overall impacts are (btw, people should start
reading up on basic oprofile use by about now ;-) but in the end, each
set of patches for each subsystem will be judged on their own merits.
Those merits include code cleanliness, code maintainainability, code
functionality, performance, testability, etc.

So, you are right that testing which compares roughly similar environments
is good. But those tests will help us identify areas where one solution
or another may have code which provides functionality in some way which
has lower impact.

I do not want to have to dig into those results in great detail if the
difference between two approaches is minor. However, if a particular
area has major impacts to performance, we need to understand how the
approaches differ and why one solution has greater impact than another.
Sometimes it is just a coding issue that can be easily addressed. Sometimes
it will be a design issue indicating that one solution or another has
a design issue which might have been better addressed by another solution.

The fun thing here (well, maybe not for each solution provider) is that
we get to cherry pick the best implementations from each solution, or
create new ones as we go which ultimate allow us to have application
virtualization, containers, or whatever you want to call them.

> Now, to have the same environment there are at least 2 ways:
> - make static binaries (not that good, but easiest way)

This is a case where "easiest" is just plain wrong. If it doesn't match
how people will use their distros and solutions out of the box it has
no real relevence to the code that will get checked in.

> - have exactly the same packages in host/VPS for all test cases.
>
> BTW, I also prefer 2nd way, but it is harder.

Herbert's suggestion here is good - if you can use exactly the same
filesystem for performance comparisons you remove one set of variables.

However, I also believe that if the difference between any two filesystems
or even distro environements doing basic performance tests (e.g.
standardized benchmarks) then there is probably some other problem that
we should be aware of. Most of the standardized benchmarks elimininate
the variance of the underlying system to the best of their ability.
For instance, kernbench carries around a full kernel (quite backlevel)
as the kernel that it builds. The goal is to make sure that the kernel
being built hasn't changed from one version to the next. In this case,
it is also important to use the same compiler since there can be
extensive variation between versions of gcc.

gerrit

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by dev on Mon, 10 Jul 2006 08:16:53 GMT

View Forum Message <> Reply to Message

Gerrit,

Great! this is what I wanted to hear :) Fully agree.

Thanks,
Kirill

> On Thu, 06 Jul 2006 14:44:23 +0400, Kirill Korotaev wrote:
>
>>Gerrit,
>>
>>
>>>>>>I assuming you are doing your tests on the same system (i.e. same
>>>>>>compiler/libs/whatever else), and you do not change that system over
>>>>>>time (i.e. you do not upgrade gcc on it in between the tests).
>>>>>
>>>>>I hope! :)
>>>>
>>>>All binaries should be built statically to work the same way inside host/guest or
>>>>you need to make sure that you have exactly the same versions of glibc and other
>>>>system libraries. At least glibc can affect perforamnce very much :/
>>>
>>>
>>>Ick - no one builds binaries statically in the real world. And,
>>>when you build binaries statically, you lose all ability to fix
>>>security problems in base libraries by doing an update of that library.
>>>Instead, all applications need to be rebuilt.
>>>
>>>Performance tests should reflect real end user usage - not contrived
>>>situations that make a particular solution look better or worse.
>>>If glibc can affect performance, that should be demonstrated in the
>>>real performance results - it is part of the impact of the solution and
>>>may need an additional solution or discussion.
>>
>>What I tried to say is that performance results done in different
>>environments are not comparable so have no much meaning. I don't want us
>>to waste our time digging in why one environment is a bif faster or slower than another.
>>I hope you don't want too.
>
>
> I *do* want to understand why one patch set or another is significantly
> faster or slower than any other. I think by now everyone realizes that
> what goes into mainline will not be some slice of vserver, or OpenVZ
> or MetaCluster or Eric's work in progress. It will be the convergance
> of the patches that enable all solutions, and those patches will be added
> as they are validated as beneficial to all participants *and* beneficial
> (or not harmful) to mainline Linux. So, testing of large environments
> is good to see where the overall impacts are (btw, people should start
> reading up on basic oprofile use by about now ;-) but in the end, each
> set of patches for each subsystem will be judged on their own merits.
> Those merits include code cleanliness, code maintainainability, code

> functionality, performance, testability, etc.
>
> So, you are right that testing which compares roughly similar environments
> is good. But those tests will help us identify areas where one solution
> or another may have code which provides functionality in some way which
> has lower impact.
>
> I do not want to have to dig into those results in great detail if the
> difference between two approaches is minor. However, if a particular
> area has major impacts to performance, we need to understand how the
> approaches differ and why one solution has greater impact than another.
> Sometimes it is just a coding issue that can be easily addressed. Sometimes
> it will be a design issue indicating that one solution or another has
> a design issue which might have been better addressed by another solution.
>
> The fun thing here (well, maybe not for each solution provider) is that
> we get to cherry pick the best implementations from each solution, or
> create new ones as we go which ultimate allow us to have application
> virtualization, containers, or whatever you want to call them.
>
>
>>Now, to have the same environment there are at least 2 ways:
>>- make static binaries (not that good, but easiest way)
>
>
> This is a case where "easiest" is just plain wrong. If it doesn't match
> how people will use their distros and solutions out of the box it has
> no real relevence to the code that will get checked in.
>
>
>>- have exactly the same packages in host/VPS for all test cases.
>>
>>BTW, I also prefer 2nd way, but it is harder.
>
>
> Herbert's suggestion here is good - if you can use exactly the same
> filesystem for performance comparisons you remove one set of variables.
>
> However, I also believe that if the difference between any two filesystems
> or even distro environements doing basic performance tests (e.g.
> standardized benchmarks) then there is probably some other problem that
> we should be aware of. Most of the standardized benchmarks elimininate
> the variance of the underlying system to the best of their ability.
> For instance, kernbench carries around a full kernel (quite backlevel)
> as the kernel that it builds. The goal is to make sure that the kernel
> being built hasn't changed from one version to the next. In this case,
> it is also important to use the same compiler since there can be
> extensive variation between versions of gcc.

>
> gerrit
>

Subject: Container Test Campaign
Posted by Clement Calmels on Tue, 11 Jul 2006 08:45:35 GMT
View Forum Message <> Reply to Message

Some updates on
http://lxc.sourceforge.net/bench/

New design, results of the stable version of openvz added, clearer
figures.

--
Clément Calmels <clement.calmels@fr.ibm.com>

Subject: Re: Container Test Campaign
Posted by Kirill Korotaev on Tue, 11 Jul 2006 09:18:57 GMT
View Forum Message <> Reply to Message

> Some updates on
> http://lxc.sourceforge.net/bench/
>
> New design, results of the stable version of openvz added, clearer
> figures.
>

1. are 2.6.16 OVZ results still for CFQ disk scheduler?
2. there is definetely something unclean in your testing as
  vserver and MCR makes dbench faster than vanilla :))
  have you took into account my notice about partition size?
  and that disk partition on which dbench works should be reformatted
  each time before test case?

Kirill

Subject: Re: Container Test Campaign
Posted by Clement Calmels on Wed, 12 Jul 2006 16:31:25 GMT
View Forum Message <> Reply to Message

Le mardi 11 juillet 2006 à 13:18 +0400, Kirill Korotaev a écrit :
> > Some updates on
> > http://lxc.sourceforge.net/bench/

> >
> > New design, results of the stable version of openvz added, clearer
> > figures.
> >
>
> 1. are 2.6.16 OVZ results still for CFQ disk scheduler?

This tests are currently in progress... for the moment, it seems that
the anticipatory io scheduler improves performance a lot.

> 2. there is definetely something unclean in your testing as
>   vserver and MCR makes dbench faster than vanilla :))

Couldn't some test be faster inside a container than with a Vanilla? For example if I want to dump all files in /proc, obviously inside a light container it will be faster because /proc visibility is limited to the container session. Just to be clear:

```
r3-21:~ # find /proc/ | wc -l
4213
r3-21:~ # mcr-execute -j1 -- find /proc/ | wc -l
729
```

I'm not sure and I'm still investigating. I'm now adding Oprofile to all tests to have more information. If you know technical reasons that imply different results, let me know. Help welcome!

--
Clément Calmels <clement.calmels@fr.ibm.com>

Subject: Re: [Vserver] Re: Container Test Campaign
Posted by Herbert Poetzl on Thu, 13 Jul 2006 02:07:04 GMT

View Forum Message <> Reply to Message

> > > Some updates on
> > > http://lxc.sourceforge.net/bench/
> > >
> > > New design, results of the stable version of openvz added, clearer
> > > figures.
> > >
> >
> > 1. are 2.6.16 OVZ results still for CFQ disk scheduler?
>
> This tests are currently in progress... for the moment, it seems that
> the anticipatory io scheduler improves performance a lot.

>
> > 2. there is definetely something unclean in your testing as
> >   vserver and MCR makes dbench faster than vanilla :))

that's not really unusual ...

> Couldn't some test be faster inside a container than with a Vanilla?

yes, they definitely can, and some very specific ones
are constantly faster regardless of how many tests
and/or setups you have ...

> For example if I want to dump all files in /proc, obviously inside a
> light container it will be faster because /proc visibility is limited
> to the container session. Just to be clear:
>
> r3-21:~ # find /proc/ | wc -l
> 4213
> r3-21:~ # mcr-execute -j1 -- find /proc/ | wc -l
> 729
>
> I'm not sure and I'm still investigating. I'm now adding Oprofile to all
> tests to have more information. If you know technical reasons that imply
> different results, let me know. Help welcome!

yes, the 'isolation' used in Linux-VServer already
gave that 'at first glance' strange behaviour that
some tests are 'faster' inside a guest than on the
real/vanilla system, so for us it is not really new
but probably it is still confusing, here are a few
reasons _why_ some tests are better than the 'original'

 - structures inside the kernel change, relations
   between certain structures change too, some of
   those changes cause 'better' behaviour, just
   because cache usage or memory placement is different

 - many checks walk huge lists to find a socket or
   process or whatever, some of them use hashes to
   speed up the search, the lightweight guests often
   provide faster access to 'related' structures

 - scheduler and memory management are tricky beasts
   sometimes it 'just happens' that certain operations
   and/or sequences are faster than other, although
   they give the same result

HTC,

Herbert

> --

>
> ____
> Vserver mailing list
> Vserver@list.linux-vserver.org
> http://list.linux-vserver.org/mailman/listinfo/vserver