IBM Information Management software

Getting Started with DB2® Express-C 9.5 and Parallels™ Virtuozzo Containers 4.0

Jeff Shantz, IBM Toronto Lab
Irina Delidjakova, IBM Toronto Lab

Contents

1. Introduction

IBM DB2® Express-C 9.5 is a no-charge community edition of the DB2 data server, completely free for development, deployment, and distribution. Its quick setup, ease of use, and powerful self-managing capabilities help increase productivity while reducing administration time and costs.

Embodying all of the core features of the more scalable DB2 editions, DB2 Express-C will use up to 2 processor cores and 2 GB of RAM. Additionally, it can be used with any storage solution, and there are no restrictions on database size or any other artificial limitations.

Combined with Parallels™ Virtuozzo Containers 4.0 for Linux, DB2 Express-C can be a powerful addition to your information management infrastructure. Virtuozzo Containers offers a high performance, space efficient operating system virtualization solution that allows new virtual machines – or *containers* – to be created within seconds.

To simplify DB2 deployment for users of Virtuozzo Containers for Linux, IBM has made available EZ templates for DB2 Express-C 9.5, optimized specifically for Virtuozzo. Leveraging the power of the Virtuozzo File System (VZFS), you can have a DB2 Express-C data server fully installed, configured, and tuned to your needs within minutes, allowing new test and production servers to be provisioned as needed.

This paper will help you get up and running with DB2 Express-C and Virtuozzo Containers for Linux, enabling you to:

- Install the DB2 Express-C EZ templates onto a Virtuozzo hardware node
- Create one or more repositories to host DB2 RPM and Debian packages
- Install and configure DB2 Express-C inside a Virtuozzo container
- Troubleshoot and tune container settings to optimize DB2 performance

2. Prerequisites

Before proceeding with this paper, please ensure that you have a Virtuozzo Containers hardware node provisioned and ready for use. Additionally, please read this section carefully to ensure that your system is adequately prepared for DB2 installation.

2.1 Operating System Templates

DB2 Express-C 9.5 EZ templates are available for both 32- and 64-bit versions of the following Linux distributions:

- Asianux 3
- Redhat AS 4
- Redhat EL 5
- SuSE ES 9 SP3
- SuSE ES 10 SP1
- Ubuntu 7.04
- Ubuntu 7.10

Prior to creating a container into which DB2 Express-C 9.5 will be installed, be sure to cache one or more of these operating system EZ templates with the noted service packs applied to your repository.

2.2 Package Dependencies

Containers based on Asianux 3, Redhat AS 4, Redhat EL 5, SuSE ES 9, SuSE ES 10, and 32-bit versions of Ubuntu 7.04 and Ubuntu 7.10 require the following three packages:

- `libstdc++5`
- `libaio`
- `ksh`

The DB2 Express-C 9.5 EZ templates will automatically resolve these dependencies so long as you have the packages that provide them present in your operating system repository or installed in your container. The following lists the package names that provide the aforementioned dependencies for each operating system:

- **Asianux 3 x86**
  - `compat-libstdc++-33-3.2.3-61.1AX.i386.rpm`
  - `libaio-0.3.106-3.2.i386.rpm`
  - `ksh-20070328-2.i386.rpm`

- **Asianux 3 x86_64**
  - compat-libstdc++-33-3.2.3-61.1AX.x86_64.rpm
  - libaio-0.3.106-3.2.x86_64.rpm
  - ksh-20070328-2.x86_64.rpm

- **Redhat AS 4 x86**
  - compat-libstdc++-33-3.2.3-47.3.i386.rpm
  - libaio-0.3.105-2.i386.rpm
  - pdksh-5.2.14-30.3.i386.rpm

- **Redhat AS 4 x86_64**
  - compat-libstdc++-33-3.2.3-47.3.x86_64.rpm
  - libaio-0.3.105-2.x86_64.rpm
  - pdksh-5.2.14-30.3.x86_64.rpm

- **Redhat EL 5 x86**
  - compat-libstdc++-33-3.2.3-61.i386.rpm
  - libaio-0.3.106-3.2.i386.rpm
  - ksh-20060214-1.4.i386.rpm

- **Redhat EL 5 x86_64**
  - compat-libstdc++-33-3.2.3-61.x86_64.rpm
  - libaio-0.3.106-3.2.x86_64.rpm
  - ksh-20060214-1.4.x86_64.rpm

- **SuSE ES 9 SP3 x86**
  - libstdc++-3.3.3-43.41.i586.rpm*
  - libaio-0.3.102-1.2.i586.rpm
  - pdksh-5.2.14-780.7.i586.rpm

- **SuSE ES 9 SP3 x86_64**
  - libstdc++-3.3.3-43.41.x86_64.rpm*
  - libaio-0.3.102-1.2.x86_64.rpm
  - pdksh-5.2.14-780.7.x86_64.rpm

- **SuSE ES 10 SP1 x86**
  - compat-libstdc++-5.0.7-22.2.i586.rpm
  - libaio-0.3.104-14.2.i586.rpm
  - ksh-93r-12.28.i586.rpm

- **SuSE ES 10 SP1 x86_64**
  - compat-libstdc++-5.0.7-22.2.x86_64.rpm
  - libaio-0.3.104-14.2.x86_64.rpm
  - ksh-93r-12.28.x86_64.rpm

- **Ubuntu 7.04 x86**
  - libstdc++5_1%3a3.3.6-15ubuntu1_i386.vz.deb
  - libaio1_0.3.106-3ubuntu2_i386.vz.deb
  - pdksh_5.2.14-20_i386.vz.deb

- **Ubuntu 7.10 x86**
    - libstdc++5_1%3a3.3.6-15ubuntu2_i386.vz.deb
    - libaio1_0.3.106-5ubuntu2_i386.vz.deb
    - pdksh_5.2.14-20build1_i386.vz.deb

> * Please note that the SuSE ES 9 SP3 EZ operating system templates cache these packages, and thus they will not get pulled from the operating system repository during DB2 installation unless you have previously removed them from the container.

2.3 Package Dependencies for 64-bit Ubuntu Containers

Containers based on 64-bit versions of Ubuntu 7.04 and Ubuntu 7.10 require the following packages:

- ia32-libs
- libaio1
- libc6
- libc6-i386
- libstdc++5
- pdksh

.
As before, the DB2 EZ templates will automatically resolve these dependencies if they are installed in your operating system repository. The following lists the names of packages that provide the aforementioned dependencies for each operating system:

- **Ubuntu 7.04 x86_64**
    - ia32-libs_1.5ubuntu9_amd64.vz.deb
    - libaio1_0.3.106-3ubuntu2_amd64.vz.deb
    - libc6_2.5-0ubuntu14_amd64.vz.deb*
    - libc6-i386_2.5-0ubuntu14_amd64.vz.deb
    - libstdc++5_1%3a3.3.6-15ubuntu1_amd64.vz.deb
    - pdksh_5.2.14-20_amd64.vz.deb

- **Ubuntu 7.10 x86_64**
    - ia32-libs_2.1ubuntu4_amd64.vz.deb
    - libaio1_0.3.106-5ubuntu2_amd64.vz.deb
    - libc6_2.6.1-1ubuntu10_amd64.vz.deb*
    - libc6-i386_2.6.1-1ubuntu10_amd64.vz.deb
    - libstdc++5_1%3a3.3.6-15ubuntu2_amd64.vz.deb
    - pdksh_5.2.14-20build1_amd64.vz.deb

> * Please note that 64-bit versions of the Ubuntu 7.04 and 7.10 operating system templates cache these packages, and thus they will not get pulled from the operating system repository during DB2 installation unless you have previously removed them from the container.

3. Installing DB2 EZ Templates on a Hardware Node

Before DB2 can be installed in a Virtuozzo container, you must install one or more DB2 EZ templates on the hardware node. These templates are packaged as RPM files and are available for download at:

ftp://ftp.software.ibm.com/software/data/db2/express/templates/V9.5.

After downloading a template, install it with the following command:

```
# rpm –ivh <template_file>
```

where `<template_file>` is one of the following:

Platform	Template File
Asianux 3 x86	`db2exc-x86-asianux-3-x86-ez-9.5.0.0-1.noarch.rpm`
Asianux 3 x86_64	`db2exc-x86_64-asianux-3-x86_64-ez-9.5.0.0-1.noarch.rpm`
Redhat AS4 x86	`db2exc-x86-redhat-as4-x86-ez-9.5.0.0-1.noarch.rpm`
Redhat AS4 x86_64	`db2exc-x86_64-redhat-as4-x86_64-ez-9.5.0.0-1.noarch.rpm`
Redhat EL5 x86	`db2exc-x86-redhat-el5-x86-ez-9.5.0.0-1.noarch.rpm`
Redhat EL5 x86_64	`db2exc-x86_64-redhat-el5-x86_64-ez-9.5.0.0-1.noarch.rpm`
SLES 9 SP3 x86	`db2exc-x86-sles-9-x86-ez-9.5.0.0-1.noarch.rpm`
SLES 9 SP3 x86_64	`db2exc-x86_64-sles-9-x86_64-ez-9.5.0.0-1.noarch.rpm`
SLES 10 SP1 x86	`db2exc-x86-sles-10-x86-ez-9.5.0.0-1.noarch.rpm`
SLES 10 SP1 x86_64	`db2exc-x86_64-sles-10-x86_64-ez-9.5.0.0-1.noarch.rpm`
Ubuntu 7.04 x86	`db2exc-x86-ubuntu-7.04-x86-ez-9.5.0.0-1.noarch.rpm`
Ubuntu 7.04 x86_64	`db2exc-x86_64-ubuntu-7.04-x86_64-ez-9.5.0.0-1.noarch.rpm`
Ubuntu 7.10 x86	`db2exc-x86-ubuntu-7.10-x86-ez-9.5.0.0-1.noarch.rpm`
Ubuntu 7.10 x86_64	`db2exc-x86_64-ubuntu-7.10-x86_64-ez-9.5.0.0-1.noarch.rpm`

You can verify the installation of your DB2 EZ templates with the following command:

```
# rpm -qa | grep db2
```

This command will list all DB2 EZ templates that you have installed on the hardware node.

4. Configuring a DB2 Repository

Once you have the desired DB2 EZ templates installed on the hardware node, you will need to configure Virtuozzo to pull DB2 packages from a repository. For this, you will need to download all necessary DB2 packages, setup a local repository, and then configure Virtuozzo to use this repository. This section will detail all steps required to get up and running with a DB2 repository.

4.1 Obtaining DB2 Packages

The first step in creating a DB2 repository is to obtain the packages for each platform on which you will install DB2. The following packages are available at **ftp://ftp.software.ibm.com/software/data/db2/express/templates/V9.5**:

Platform	Filename
32-bit versions of Asianux, Redhat, SuSE	`db2exc-x86-9.5.0.0-1.i586.rpm`
64-bit versions of Asianux, Redhat, SuSE	`db2exc-x86_64-9.5.0.0-1.x86_64.rpm`
32-bit versions of Ubuntu	`db2exc-x86_9.5.0-2_i386.deb`
64-bit versions of Ubuntu	`db2exc-x86-64_9.5.0-2_amd64.deb`

You should download all packages corresponding to the platforms on which you wish to use DB2. For example, if you want to be able to install DB2 Express-C 9.5 in both 32- and 64-bit versions of Ubuntu, you should obtain both `db2exc-x86_9.5.0-2_i386.deb` and `db2exc-x86-64_9.5.0-2_amd64.deb`.

4.2 Configuring a DB2 Repository for RPM-Based Operating Systems

If you wish to install DB2 into Asianux, Redhat, or SuSE containers, you will need to obtain the DB2 RPM packages (see section 4.1). Once you have these, you are ready to configure a local repository.

This section will detail the configuration of a DB2 RPM repository in a container using the `sles-10-x86_64` EZ operating system template. If you choose to use another operating system in your repository container, please note that, while the process will be similar, certain paths may differ.

1. On the hardware node, create the container that will host the repository:

```
# vzctl create 500 \
        --ostemplate sles-10-x86_64 \
        --ipadd 172.16.50.50 \
        --hostname DB2RPMREPO
```

   This will create a container with VEID `500` and will assign it the IP address `172.16.50.50`. If VEID `500` is already in use in your system, you will need to select a different VEID. Similarly, if the IP address `172.16.50.50` is already in use by another container, or if you are using a different subnet than 172.16.0.0, you will need to modify the IP address accordingly in the command above.

2. Set the `root` password for the container, and assign it adequate disk space to host the DB2 packages:

```
# vzctl set 500 \
        --userpasswd root:password \
        --diskspace 1048576:2097152 \
        --save
```

   Here, the `root` password for the container has been set to `password`, and the container has been given a soft disk space limit of 1GB, and a hard disk space limit of 2 GB.

3. Start the container:

```
# vzctl start 500
```

4. Create the directories in the container that will house the DB2 packages by issuing the following command on the hardware node:

```
# mkdir -p /vz/root/500/srv/www/htdocs/download/db2exc_x86/base
# mkdir -p /vz/root/500/srv/www/htdocs/download/db2exc_x86/updates
# mkdir -p /vz/root/500/srv/www/htdocs/download/db2exc_x86_64/base
# mkdir -p /vz/root/500/srv/www/htdocs/download/db2exc_x86_64/updates
```

   Note that `/srv/www/htdocs/` is the default path to the public web root in SLES 10 SP1. If you are using a different operating system in your repository container, you will need to change the paths in the commands above accordingly.

5. Copy the DB2 packages obtained in section 4.1 to the appropriate locations:

```
# cp db2exc-x86-9.5.0.0-1.i586.rpm \
     /vz/root/500/srv/www/htdocs/download/db2exc_x86/base

# cp db2exc-x86_64-9.5.0.0-1.x86_64.rpm \
     /vz/root/500/srv/www/htdocs/download/db2exc_x86_64/base
```

6. Download the latest version of the `createrepo` package by searching for it on **http://rpmfind.net** and install it on the hardware node:

```
# rpm –Uvh createrepo-0.4.4-9.3.noarch.rpm
```

   Note that the version of `createrepo` that you obtain may differ from the one listed in the command above.

7. Create the repository metadata:

```
# createrepo /vz/root/500/srv/www/htdocs/download/db2exc_x86/base
# createrepo /vz/root/500/srv/www/htdocs/download/db2exc_x86/updates
# createrepo /vz/root/500/srv/www/htdocs/download/db2exc_x86_64/base
# createrepo \
  /vz/root/500/srv/www/htdocs/download/db2exc_x86_64/updates
```

8. Edit the file `/vz/template/conf/vztt/url.map` and add the variable $DB2SERVER pointing to the IP address of the repository container (see step 1). For example, if your repository container uses the IP address `172.16.50.50`, you would add the following line to `/vz/template/conf/vztt/url.map`:

```
$DB2SERVER      http://172.16.50.50
```

9. Start the web server in the container and set it to start automatically at boot time, by issuing the following commands on the hardware node:

```
# vzctl exec2 500 rcapache2 start
# vzctl exec2 500 chkconfig --set apache2 on
```

   Note that, depending on the operating system used in the repository container, the commands to start Apache and set it to start automatically at boot time may differ from those shown above.

You now have a properly configured local repository serving DB2 RPM packages for both 32- and 64-bit versions of the Asianux, Redhat, and SuSE Linux distributions.

4.3 Configuring a DB2 Repository for Debian-Based Operating Systems

If you wish to be able to install DB2 into Ubuntu 7.04 or 7.10 containers, you will need to obtain the DB2 Debian packages (see section 4.1). Once you have obtained these, you are ready to configure a local repository.

This section will detail the configuration of a DB2 Debian repository in a container using the `ubuntu-7.04-x86_64` EZ operating system template. If you choose to use another operating system to configure the repository, please note that, while the process will be similar, certain paths may differ.

1. On the hardware node, create the container that will host the repository:

   ```
   # vzctl create 501 \
           --ostemplate ubuntu-7.04-x86_64 \
           --ipadd 172.16.50.51 \
           --hostname DB2DEBREPO
   ```

   This will create a container with VEID `501` and will assign it the IP address `172.16.50.51`. If VEID `501` is already in use in your system, you will need to select a different VEID. Similarly, if the IP address `172.16.50.51` is already in use by another container, or if you are using a different subnet than 172.16.0.0, you will need to modify the IP address accordingly in the command above.

2. Set the `root` password for the container, and assign it adequate disk space to host the DB2 packages:

   ```
   # vzctl set 501 \
           --userpasswd root:password \
           --diskspace 1048576:2097152 \
           --save
   ```

   Here, the `root` password for the container has been set to `password`, and the container has been given a soft disk space limit of 1GB, and a hard disk space limit of 2 GB.

3. Start the container:

   ```
   # vzctl start 501
   ```

4. Create the directories in the container that will house the DB2 packages by issuing the following command on the hardware node:

   ```
   # mkdir -p /vz/root/501/var/www/download/db2exc/binary
   # mkdir -p /vz/root/501/var/www/download/db2exc/updates
   ```

Note that `/var/www/` is the default path to the public web root in Ubuntu 7.04. If you are using a different operating system in your repository container, you will need to change the paths in the commands above accordingly.

5. Copy the DB2 packages obtained in section 4.1 to the `binary/` directory:

```
# cp db2exc-x86_9.5.0-2_i386.deb \
     /vz/root/501/var/www/download/db2exc/binary

# cp db2exc-x86-64_9.5.0-2_amd64.deb \
     /vz/root/501/var/www/download/db2exc/binary
```

6. Install the `dpkg-dev` package into the repository container:

```
# vzpkg install -p 501 dpkg-dev
```

7. Create the repository with the `dpkg-scanpackages` command:

```
# vzctl exec2 501 "cd /var/www/download/db2exc && \
                   dpkg-scanpackages binary /dev/null | \
                   gzip -9c > binary/Packages.gz"

# vzctl exec2 501 "cd /var/www/download/db2exc && \
                  dpkg-scanpackages updates /dev/null | \
                  gzip -9c > updates/Packages.gz"
```

8. Edit the file `/vz/template/conf/vztt/url.map` and add the variable `$DB2DEBSERVER` pointing to the IP address of the repository container (see step 1). For example, if your repository container uses the IP address `172.16.50.51`, you would add the following line to `/vz/template/conf/vztt/url.map`:

```
$DB2DEBSERVER      http://172.16.50.51
```

9. Start the web server in the container and set it to start automatically at boot time, by issuing the following commands on the hardware node:

```
# vzctl exec2 501 /usr/sbin/apache2ctl start
# echo "NO_START=0" > /vz/root/501/etc/default/apache2
```

Note that, depending on the operating system used in the repository container, the commands to start Apache and set it to start automatically at boot time may differ from those shown above.

You now have a properly configured local repository serving DB2 Debian packages for both 32- and 64-bit versions of Ubuntu.

5. Installing and Configuring DB2 in a Container

Now that you have one or more repositories setup and configured, you can install DB2 inside your Virtuozzo containers. This section will detail the process of installing DB2 into a container, configuring DB2 for use, and optimizing your container settings to maximize DB2 performance.

5.1 Configuring a Container for DB2

Before installing DB2 into your container, there are a few adjustments that should be made to the configuration of the container to ensure optimal DB2 performance. This section details relevant container settings and recommended values for these settings.

Please note that the parameter values specified in this section are recommendations only. They provide a good starting point for DB2 and should be fine for most purposes. However, more intensive workloads or the use of advanced DB2 features may require higher allocations for certain parameters. See section 6 for details on how to determine if parameter limits need to be increased.

5.1.1 Disk Space

It is recommended that at least 1 – 2 GB of disk space be allocated to the container for DB2. As your database size increases, you may need to adjust this parameter periodically.

Disk space values passed to `vzctl` are specified in KB, so 1 – 2 GB will be in the range 1,048,576 – 2,097,152 KB. To allocate this to the container, issue the following command:

```
# vzctl set VEID --diskspace 1048576:2097152 --save
```

where `VEID` is the ID of the container being modified.

5.1.2 shmpages

Both the barrier and limit values for this parameter should be approximately equal to 90% of available disk space allocated to the container, divided by its page size. For example, if you have 2 GB available and your system has a 4 KB page size, then this parameter would be 90% x 2 GB = 1.8 GB. Divide this by the 4 KB page size to get 471,859.

```
# vzctl set VEID --shmpages 471859:471859 --save
```

5.1.3 privvmpages

The recommended values for this parameter are 1 GB for the barrier value and 2 GB for the limit value.

This parameter requires that its values be specified in pages, thus – assuming we are using a system having a 4 KB page size – we divide 1 GB by the 4 KB page size to obtain 266,144 for the barrier value. Similarly, dividing 2 GB by 4 KB yields 524,288 for the limit value.

```
# vzctl set VEID --privvmpages 266144:524288 --save
```

5.1.4 vmguarpages

The recommended value for this parameter is 64 KB (65,535 bytes) for the barrier value and the default value for the limit. The default limit value can be obtained by executing the following command on either the hardware node or in the container:

```
# cat /proc/user_beancounters
```

Look for the `vmguarpages` parameter, obtain the default value from its `limit` column, and then set the container's parameter accordingly:

```
# vzctl set VEID --vmguarpages 65535:9223372036854775807 --save
```

5.1.5 oomguarpages

The recommended value for the barrier value of this parameter is double the barrier value of the `vmguarpages` parameter, or 131,070. The limit value should be set to the default limit value of the `oomguarpages` parameter, which can be obtained by examining `/proc/user_beancounters`, as in section 5.1.4.

```
# vzctl set VEID --oomguarpages 131070:9223372036854775807 --save
```

5.1.6 kmemsize

The recommended values for this parameter are 1 GB and 2 GB for the barrier and limit values, respectively. Values for this parameter are specified in bytes.

```
# vzctl set VEID --kmemsize 1073741824:2147483648 --save
```

5.1.7 numothersock

The recommended value for this parameter is 250 for both the barrier and limit values.

```
# vzctl set VEID --numothersock 250:250 --save
```

5.2 Installing DB2 in a Container

The EZ templates created for DB2 Express-C make installing DB2 into your containers quick and painless. To install DB2 into a 32-bit container, simply issue the following command on the hardware node:

```
# vzpkg install VEID db2exc-x86
```

where `VEID` is the ID of the container into which DB2 should be installed.

Similarly, to install DB2 into a 64-bit container, issue the following command:

```
# vzpkg install VEID db2exc-x86_64
```

From there, `vzpkg` will take care of downloading the appropriate DB2 packages from the repositories, and will install DB2 into the specified container.

5.3 Configuring DB2 in a Container

While the DB2 Express-C EZ templates handle the process of installing DB2 in a container, they leave the process of creating users, groups, and instances, and performing other DB2 configuration up to you. This allows you full control over your DB2 environment.

1. Login to your container as `root` via SSH. For example, to connect to a container listening on IP `172.16.10.10`, you would issue the following command:

   ```
   # ssh root@172.16.10.10
   ```

2. Create the primary groups for the instance owner, fenced user, and DAS owner:

   ```
   # groupadd -g 999 db2iadm1
   # groupadd -g 998 db2fadm1
   # groupadd -g 997 dasadm1
   ```

3. Create the instance owner, fenced user, and DAS owner:

   ```
   # useradd -u 1004 -g db2iadm1 -m -d /home/db2inst1 db2inst1
   # useradd -u 1003 -g db2fadm1 -m -d /home/db2fenc1 db2fenc1
   # useradd -u 1002 -g dasadm1 -m -d /home/dasusr1 dasusr1
   ```

4. Set passwords for the instance owner, fenced user, and DAS owner (recommended):

```
# passwd db2inst1
# passwd db2fenc1
# passwd dasusr1
```

5. Create the DAS instance:

```
# /opt/ibm/db2/V9.5/instance/dascrt -u dasusr1
```

6. Create the `db2inst1` instance:

```
# /opt/ibm/db2/V9.5/instance/db2icrt -u db2fenc1 db2inst1
```

7. Add an entry to `/etc/services` specifying a service name and port number for the instance created in step 6. For example, to configure the `db2inst1` instance to use the service name `db2c_db2inst1`, and to listen on port `50001`, you would execute the following command:

```
# echo "db2c_db2inst1  50001/tcp" >> /etc/services
```

8. Configure DB2 to use the service name chosen in step 7:

```
# su - db2inst1 -c "db2 update dbm cfg using svcename
db2c_db2inst1"
```

9. Configure DB2 to use the TCP/IP protocol:

```
# su - db2inst1 -c "db2set DB2COMM=tcpip"
```

10. Stop and restart DB2 to allow the changes to take effect:

```
# su - db2inst1 -c db2stop
# su - db2inst1 -c db2start
```

5.4 Testing DB2 Configuration

Once you have DB2 installed and configured, it is time to begin using it. Before doing so, it is a good idea to run a quick test to ensure that everything works as it should.

1. Login to your container as `root` via SSH. For example, to connect to a container listening on IP `172.16.10.10`, you would issue the following command:

```
# ssh root@172.16.10.10
```

6. Tuning and Troubleshooting

If you receive any errors while using DB2 in a Virtuozzo container, or if you find performance to be lagging, it is a good idea to examine your container settings to determine if any parameters are exceeding their barrier or limit values.

There are two ways to check this:

1. Examine the contents of `/proc/user_beancounters` in the container as the

ensure that no resource allocation failures occur. See section 5.1 for details on modifying container settings.

To determine whether or not any resource allocation failures have already occurred, we are interested in the `failcnt` column. This column counts the number of failed requests for increased resources. For example, from the sample output above, we see that the limit for the privvmpages parameter is 69,632. If the container were to request more than this limit value, then the resource request would go unfulfilled, and the value of the `failcnt` column would be incremented.

If you notice a non-zero value in the `failcnt` column of a parameter, this signifies that resource requests are failing and that you need to immediately increase the barrier and limit values. Otherwise, container performance and stability may be affected.

However, it is also important to note that the `failcnt` column does not reset back to zero until a container is restarted. Thus, if you increase the barrier and limit values for a failing parameter, do not be surprised when its `failcnt` value remains the same. The important thing is that the value of `failcnt` does not increase.

7. Getting Assistance

We hope that this paper has been useful in helping you get up and running with DB2 Express-C 9.5 and Parallels Virtuozzo Containers 4.0 for Linux. We recognize that, despite best efforts, things may go wrong occasionally, and you may find yourself needing support.

The following DB2 resources are recommended:

- IBM DB2 9.5 Information Center for Linux, Unix, and Windows. This site houses all IBM documentation related to DB2 9.5 for LUW, and is a place to find quick information on DB2 syntax, commands, and data objects:

  http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp

- The developerWorks IBM DB2 Express Forum hosts a community of helpful DB2 users and experts alike, many of whom are willing to help answer any questions you may have:

  http://www.ibm.com/developerworks/forums/forum.jspa?forumID=805

Additionally, the following resources are recommended for Virtuozzo:

- Virtuozzo documentation is available online in PDF and HTML formats:

  http://www.parallels.com/en/products/virtuozzo/docs/

- A well-trafficked forum exists for Virtuozzo Containers, which is recommended for users experiencing problems with Virtuozzo configuration or usage:

  http://forum.swsoft.com/forumdisplay.php?f=65

Finally, questions and errata pertaining to this documentation can be directed to **db2validate@ca.ibm.com**.

IBM Canada
8200 Warden Avenue
Markham, ON
L6G 1C7
Canada

Printed in Canada
07/2008

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates. The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

The information in this publication is provided AS IS without warranty. Such information was obtained from publicly available sources, is current as of July 2008, and is subject to change. Any performance data included in the paper was obtained in the specific operating environment and is provided as an illustration. Performance in other operating environments may vary. More specific information about the capabilities of products described should be obtained from the suppliers of those products.