



z/TPF V1.1

TPF Users Group Fall 2008

Title: *z/TPF Support for MySQL*

Name: Mark Cooper
Venue: Open Source Subcommittee

AIM Enterprise Platform Software
IBM z/Transaction Processing Facility Enterprise Edition 1.1.0

Any reference to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

© 2008 IBM Corporation



- What is MySQL?
 - Popular high-performance relational database
 - Cost-effective and reliable.
 - Relational database – collection of “relations” (tables)
 - Table data is “related”
 - Data in a specific column is in the same “domain” (same data type and conform to the same restraints)
- Why use MySQL on z/TPF?
 - Port applications that use an SQL interface to z/TPF
 - Use a relational database on z/TPF for new applications
 - Transport data on/off z/TPF easily

Install z/TPF support for MySQL:

- See the base/mysql/readme.txt file in the z/TPF source code
- www.mysql.org
- Open source version of MySQL Version 5.0.27 or later is available to the development community.
- For version 5.0.27, go to <http://downloads.mysql.com/archives/mysql-5.0/> and click on mysql-5.0.27.tar.gz. Create a TPF_ROOT/opensource directory and download the source archive.
- From the TPF_ROOT/opensource directory, extract the source archive:
`tar -zxf mysql-5.0.27.tar.gz`
- From the TPF_ROOT/opensource directory, create a symbolic link: `ln -s mysql-5.0.27 mysql`
- Build and load the MySQL

Create MySQL directory and mount a thread-safe file system such as one of the processor unique file systems (PFS, FFS or MFS):

```
ZFILE mkdir /usr/mysql
```

```
CSMP0097I 15.18.19 CPU-B SS-BSS SSU-HPN IS-01
```

```
FILE0003I 10.51.35 mkdir /usr/... COMPLETED SUCCESSFULLY. NO OUTPUT TO DISPLAY+
```

```
ZFILE mount -t pfs pfsrecl /usr/mysql
```

```
CSMP0097I 15.18.43 CPU-B SS-BSS SSU-HPN IS-01
```

```
FILE0003I 15.18.43 mount -t pf... COMPLETED SUCCESSFULLY. NO OUTPUT TO DISPLAY+
```

Note: If not present, need to define and initialize a new pool or fixed file system. For example, **ZAVFS BUILD PFS PFSRECL CRID FC8A DRID FC8B**

Verify TCP/IP connection:

```
zttcp display all
```

```
CSMP0097I 13.35.34 CPU-B SS-BSS SSU-HPN IS-01 _  
TTCP0184I 13.35.34 IP CONNECTIONS DISPLAY
```

| OSA NAME | CURRENT STATUS | DESIRED STATUS | LOCAL IP ADDR | TRACE | READ | DATA |
|----------|----------------|----------------|---------------|-------|------|------|
| OSA1 | ACTIVE | ACTIVE | 9.057.013.086 | ALL | 2186 | 21AC |

END OF DISPLAY+

Verify thread support:

```
zctka display thread
```

```
CSMP0097I 13.42.24 CPU-B SS-BSS  SSU-HPN  IS-01  
CTKA0020I 13.42.24  STORAGE ALLOCATIONS ON PROCESSOR B
```

```
KEYWORD          ALLOCATION  
MTHD              50  
TSTK              256  
END OF DISPLAY+
```

If values set less than these values, use the `zctka alter` command to change them and `zripl`.

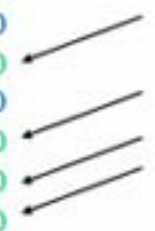
Verify adequate heap size:

```

zctka display heap
CSMP0097I 13.44.02 CPU-B SS-BSS  SSU-HPN  IS-01
CTKA0020I 13.44.02  STORAGE ALLOCATIONS ON PROCESSOR B

KEYWORD          ALLOCATION
PEH              16
HAVL1            64
HAVL2            16
HAVL3            16
HAVL4             2
HAVSZ1           64
HAVSZ2           256
HAVSZ3           1024
HAVSZ4           4096
PES              16
SHA              0
SHP              20
EMPS             200
ESPS             200
MMES             200
XMMES           100
MAXCMMES         200
EHTRACE          64
END OF DISPLAY+

```



If EMPS and MMES are not at least 40, use the zctka alter command to change them and zripl. The value of MAXXMMESS and XMMES need to be configured for your system. Start by using the values above and then a more accurate value can be determined by observing 64-bit heap usage in data collection.

If necessary, define local domain name to allow TCP/IP loopback support to work with MySQL (MySQL default = localhost):

```
zdtcp dom def-yourlocaldomainname.com
CSMP0097I 08.44.35 CPU-B SS-BSS SSU-HPN IS-01
DTCP0067I 08.44.35 TPF DEFAULT DOMAIN NAME IS DEFINED +
```

Note: If you do not have the local domain name defined and attempt to access the local MySQL database without specifying the IP address, you may receive the following error:

```
ERROR 2005 (HY000): Unknown MySQL server host 'localhost' (4)
```


Verify FTP client is enabled (APARs PJ31266 and PJ31296):

zdmmap curl

CSMP0097I 10.26.35 CPU-B SS-BSS SSU-HPN IS-01

DMAP0003I 10.26.35 LINK MAP DATA DISPLAY

CURL ACTIVE IN LOADSET BASE IN SUBSYSTEM BSS

PROGRAM ADDRESS 0000000380BE07E0

PROGRAM SIZE 0002F004

base64 - OBJ FILE AT ADDR 0000000380BE07E0

OBJECT FILE SIZE 000007A0

COMPILED ON 2008/09/15 AT 21.21.28

connect - OBJ FILE AT ADDR 0000000380BE0F80

OBJECT FILE SIZE 00000FD8

COMPILED ON 2008/09/15 AT 21.21.29

cookie - OBJ FILE AT ADDR 0000000380BE1F58

OBJECT FILE SIZE 00001CA8

COMPILED ON 2008/09/15 AT 21.21.31

easy - OBJ FILE AT ADDR 0000000380BE3C00

MORE.... TPFXA1

If the function does not exist, then use FTP commands to manually transfer the necessary MySQL files to TPF instead of using the bootstrap command on the next slide. The **ZMSQL bootstrap** command lists the commands to do it manually.

Create MySQL configuration file for the bootstrap process and run bootstrap:

```
zfile echo yourlinux.com linuxuser linuxpasswd > /etc/ftp.mysql.conf
```

```
CSMP0097I 08.51.20 CPU-B SS-BSS SSU-HPN IS-01
```

```
FILE0003I 08.51.20 echo linux... COMPLETED SUCCESSFULLY. NO OUTPUT TO DISPLAY+
```

```
zmsql bootstrap path-/usr
```

```
CSMP0097I 08.22.38 CPU-B SS-BSS SSU-HPN IS-01
```

```
MSQL0001I 08.22.38 ZMSQL BOOTSTRAP PROCESSING STARTED+
```

```
CSMP0097I 08.22.46 CPU-B SS-BSS SSU-HPN IS-01
```

```
CYC00003I 08.22.46 POOL TYPE 4DP DEVICE DEVA DIRECTORIES
```

```
1073 THRU 1073 COUNTS 8000 IN USE+
```

```
CSMP0097I 08.22.55 CPU-B SS-BSS SSU-HPN IS-01
```

```
MSQL0004I 08.22.55 ZMSQL BOOTSTRAP PROCESSING COMPLETED+
```

Add MySQL server to internet daemon:

```
ZINET ADD s-MYSQL pgm-CMYS model-DAEMON user-tpfuser1 state-NORM act-OPER
CSMP0097I 15.16.16 CPU-B SS-BSS SSU-HPN IS-01
INET0011I 15.16.16 SERVER MYSQL ADDED TO THE
                INETD CONFIGURATION FILE+
```

Note: The user parameter sets the default user running on the MySQL server.

Start MySQL server (output here is for the first time mysql is started):

```
zinet start s-mysql
CSMP0097I 08.58.41 CPU-B SS-BSS  SSU-HPN  IS-01
INET0017I 08.58.41 SERVER MYSQL STARTED+
CSMP0097I 08.58.41 CPU-B SS-BSS  SSU-HPN  IS-01
INET0017I 08.58.41 SERVER MYSQL          STARTED+
CSMP0097I 08.58.43 CPU-B SS-BSS  SSU-HPN  IS-01
InnoDB: The first specified data file ./ibdata1 did not exist:+
CSMP0097I 08.58.43 CPU-B SS-BSS  SSU-HPN  IS-01
InnoDB: a new database to be created!+
CSMP0097I 08.58.43 CPU-B SS-BSS  SSU-HPN  IS-01
080729 8:59:43 InnoDB: Setting file ./ibdata1 size to 10 MB+
CSMP0097I 08.58.43 CPU-B SS-BSS  SSU-HPN  IS-01
InnoDB: Database physically writes the file full: wait...+
CSMP0097I 08.58.45 CPU-B SS-BSS  SSU-HPN  IS-01
080729 8:59:45 InnoDB: Log file ./ib_logfile0 did not exist: new to be created
+
CSMP0097I 08.58.45 CPU-B SS-BSS  SSU-HPN  IS-01
InnoDB: Setting log file ./ib_logfile0 size to 5 MB+
CSMP0097I 08.58.45 CPU-B SS-BSS  SSU-HPN  IS-01
InnoDB: Database physically writes the file full: wait...+
CSMP0097I 08.58.46 CPU-B SS-BSS  SSU-HPN  IS-01
080729 8:59:46 InnoDB: Log file ./ib_logfile1 did not exist: new to be created
+
CSMP0097I 08.58.46 CPU-B SS-BSS  SSU-HPN  IS-01
InnoDB: Setting log file ./ib_logfile1 size to 5 MB+
CSMP0097I 08.58.46 CPU-B SS-BSS  SSU-HPN  IS-01
InnoDB: Database physically writes the file full: wait...+
CSMP0097I 08.58.47 CPU-B SS-BSS  SSU-HPN  IS-01
InnoDB: Doublewrite buffer not found: creating new+
CSMP0097I 08.58.48 CPU-B SS-BSS  SSU-HPN  IS-01
InnoDB: Doublewrite buffer created+
CSMP0097I 08.58.48 CPU-B SS-BSS  SSU-HPN  IS-01
InnoDB: Creating foreign key constraint system tables+
CSMP0097I 08.58.48 CPU-B SS-BSS  SSU-HPN  IS-01
InnoDB: Foreign key constraint system tables created+
CSMP0097I 08.58.48 CPU-B SS-BSS  SSU-HPN  IS-01
080729 8:59:48 InnoDB: Started; log sequence number 0 0+
CSMP0097I 08.58.48 CPU-B SS-BSS  SSU-HPN  IS-01
080729 8:59:48 Note" /usr/libexec/mysqld: ready for connections.+
CSMP0097I 08.58.48 CPU-B SS-BSS  SSU-HPN  IS-01
Version: '5.0.27' socket: '' port: 3306 Source distribution+
```

MySQL server is started ...

zinet disp s-mysql

CSMP0097I 09.03.20 CPU-B SS-BSS SSU-HPN IS-01

INET0031I 09.03.20 START OF ZINET DISPLAY OF ACTIVE SERVER

SERVER - MYSQL PROCID - B ACTIVATION - OPER

PGM - CMYS PARM -

PROTOCOL - PORT - 00000 MODEL - DAEMON _

SERVERRORS - 00000 SERVETIME - 00000 USER - youruser

MAXPROC - 00001 TIMEOUT - 00000 STATE - NORM

AORLENGTH - 00000 BACKLOG - 00000

IP - ANY SOCKET - 00000000/00000000 COUNT - 0000000001

TOTAL COUNT - 0000000001

END OF DISPLAY+

MySQL server is started (port 3306 is MySQL) ...

zsock sum proto-tcp

CSMP0097I 09.17.54 CPU-B SS-BSS SSU-HPN IS-01

SOCK0021I 09.17.54 SOCKET SUMMARY INFORMATION

| SOCKET DESC | LOCAL IP | LOCAL PORT | REMOTE IP | REMOTE PORT | PROT | STATE |
|-----------------|---------------|---------------|---------------|----------------|------|-------------|
| 00C00003 | | 21 | | | TCP | LISTEN |
| 00C00006 | | 1414 | | | TCP | LISTEN |
| 00C00007 | 9.057.013.086 | 21 | 9.056.224.021 | 39007 | TCP | ESTABLISHED |
| 00C00043 | | 3306 | | | TCP | LISTEN |
| SUMMARY TOTAL | | 4 | | | | |
| END OF DISPLAY+ | | | | | | |

Display all databases on MySQL server

```
zmsql exec-'show databases;'
```

```
CSMP0097I 15.49.12 CPU-B SS-BSS SSU-HPN IS-01  
MSQL0006I 15.49.12 BEGIN DISPLAY OF SQL QUERY RESULTS
```

```
-----  
show databases  
-----
```

```
-----  
| Database |  
-----  
| information_schema |  
| mysql |  
| test |  
-----
```

```
3 rows in set
```

```
Bye
```

```
END OF DISPLAY+
```

Display 1-MB frames used by MySQL

```

ZSTAT OWNER NAME-IMYSQLD BLOCK-FRM1MB
CSMP0097I 14.48.19 CPU-B SS-BSS  SSU-HPN  IS-01
STAT0023I 14.48.19 BLOCK OWNER DISPLAY
                IOB   FRAME  COMMON      SWB      ECB      FRM1MB
ALLOCATED                2704    5000      250    1252    150        500
AVAILABLE                2704    4807      247    1118    124        242
-
                IOB   FRAME  COMMON      SWB      ECB      FRM1MB
IMYSQLD                                82
END OF DISPLAY+

```

The purpose of this slide is to get a feeling for the memory usage of a MySQL server.

CreateDatabases.txt – Example file containing SQL commands:

```
CREATE DATABASE food;
USE food;
CREATE TABLE fruit (Fruit VARCHAR(15),calories INTEGER, percent_water TINYINT(3),
fiber TINYINT(4), fat TINYINT(3), protein INTEGER, sugar INTEGER) ENGINE = MYISAM;
INSERT INTO fruit VALUES ('apple', 49, 84, 2.3, 0, 0.4, 11.8);
INSERT INTO fruit VALUES ('avocado', 126, 81, 0.2, 10, 2.0, 7.0);
INSERT INTO fruit VALUES ('blueberry', 48, 80, 8.4, 0, 1.0, 11.0);
INSERT INTO fruit VALUES ('banana', 88, 76, 2.7, 0, 1.2, 20.4);
INSERT INTO fruit VALUES ('grapes', 64, 83, 2.2, 0, 0.6, 15.5);
INSERT INTO fruit VALUES ('orange', 47, 87, 1.8, 0, 1.0, 10.6);
CREATE TABLE veggies (Vegetable VARCHAR(15), calories INTEGER, fiber TINYINT(4), fat TINYINT(3),
protein INTEGER, sugar INTEGER) ENGINE = INNODB;
INSERT INTO veggies VALUES ('bell pepper', 30, 2, 0, 1, 4);
INSERT INTO veggies VALUES ('broccoli', 45, 5, 0, 5, 3);
INSERT INTO veggies VALUES ('brussel sprouts', 40, 3, 0, 2, 2);
INSERT INTO veggies VALUES ('carrot', 35, 2, 0, 1, 5);
INSERT INTO veggies VALUES ('spinach', 40, 5, 0, 2, 0);
CREATE DATABASE golfers;
USE golfers;
CREATE TABLE pga (Name VARCHAR(20), Majors TINYINT(2));
INSERT INTO pga VALUES ('Tiger', 11);
INSERT INTO pga VALUES ('Jack', 18);
INSERT INTO pga VALUES ('Monty', 0);
CREATE TABLE 2006DCamateur (Position VARCHAR(3), Name VARCHAR(20), Score SMALLINT(3))
ENGINE=MYISAM;
:
:
```

Create a file on linux and ftp it to TPF:

```
markcoo@linuxtpf:~> ftp 9.57.13.86
```

```
Connected to 9.57.13.86.
```

```
220 TPF FTP server (Version 1.01) ready.
```

```
331 Guest login ok, type your name as password.
```

```
230 Guest login ok, access restrictions apply.
```

```
Remote system type is UNIX.
```

```
Using binary mode to transfer files.
```

```
ftp> ascii
```

```
200 Type set to A.
```

```
ftp> cd test
```

```
250 CWD command successful.
```

```
ftp> put CreateDatabases.txt
```

```
local: CreateDatabases.txt remote: CreateDatabases.txt
```

```
500 'EPSV': command not understood.
```

```
227 Entering Passive Mode (9,57,13,86,4,59)
```

```
150 Opening ASCII mode data connection for 'CreateDatabases.txt'.
```

```
100% |*****| 3990 6.29 MB/s --:-- ETA
```

```
226 Transfer complete.
```

```
3990 bytes sent in 00:00 (70.53 KB/s)
```

NOTE: I created a /test directory on my TPF system and gave it permission to allow a client to write to it:

```
zfile cd ~ zfile mkdir /test zfile chmod -R 0777 /test
```

Command to create/populate databases & tables:

```
zmsql file-/test/CreateDatabases.txt
CSMP0097I 15.51.44 CPU-B SS-BSS SSU-HPN IS-01 _
MSQL0006I 15.51.44 BEGIN DISPLAY OF SQL QUERY RESULTS
```

```
-----
CREATE DATABASE golfers
-----
```

```
Query OK, 1 row affected
```

```
-----
CREATE TABLE pga (Name VARCHAR(20), Majors TINYINT(2))
-----
```

```
Query OK, 0 rows affected
```

```
-----
INSERT INTO pga VALUES ('Tiger', 11)
-----
```

```
Query OK, 1 row affected
```

```
MORE DATA AVAILABLE, ENTER ZPAGE TO CONTINUE+
```

Display the newly created databases:

```
zmsql exec-'show databases;'
```

```
CSMP0097I 09.49.31 CPU-B SS-BSS SSU-HPN IS-01
```

```
MSQL0006I 09.49.31 BEGIN DISPLAY OF SQL QUERY RESULTS
```

```
-----  
show databases
```

```
-----
```

```
.-----.  
| Database |  
.-----.  
| information_schema |  
| food |  
| golfers |  
| mysql |  
| softball |  
| test |  
| tpf |  
.-----.
```

```
7 rows in set
```

```
MORE DATA AVAILABLE, ENTER ZPAGE TO CONTINUE+
```

Display all tables in database "food":

```
zmsql exec-'use food; show tables;'  
CSMP0097I 15.32.20 CPU-B SS-BSS  SSU-HPN  IS-01  
MSQL0006I 15.32.20 BEGIN DISPLAY OF SQL QUERY RESULTS  
-----  
show tables  
-----  
  
.-----.  
| Tables_in_food |  
.-----.  
| fruit          |  
| veggies       |  
.-----.  
2 rows in set  
  
Bye  
END OF DISPLAY+
```

Display data in the table "fruit":

```
zmsql exec-'use food; select * from fruit;'
CSMP0097I 15.33.46 CPU-B SS-BSS SSU-HPN IS-01 _
MSQL0006I 15.33.46 BEGIN DISPLAY OF SQL QUERY RESULTS
```

```
-----
select * from fruit
```

```
-----
```

| Fruit | calories | percent_water | fiber | fat | protein | sugar |
|-----------|----------|---------------|-------|-----|---------|-------|
| apple | 49 | 84 | 2 | 0 | 0 | 12 |
| avocado | 126 | 81 | 0 | 10 | 2 | 7 |
| blueberry | 48 | 80 | 8 | 0 | 1 | 11 |
| banana | 88 | 76 | 3 | 0 | 1 | 20 |
| grapes | 64 | 83 | 2 | 0 | 1 | 16 |
| orange | 47 | 87 | 2 | 0 | 1 | 11 |

```
6 rows in set
```

```
Bye _
```

```
MORE DATA AVAILABLE, ENTER ZPAGE TO CONTINUE+
```

Display data in the table “fruit” from linux:

```
markcoo@linuxtpf:~> mysql -h 9.57.13.86
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

```
Your MySQL connection id is 9 to server version: 5.0.27
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql> use food; select * from fruit;
```

```
Reading table information for completion of table and column names
```

```
You can turn off this feature to get a quicker startup with -A
```

```
Database changed
```

| Fruit | calories | percent_water | fiber | fat | protein | sugar |
|-----------|----------|---------------|-------|-----|---------|-------|
| apple | 49 | 84 | 2 | 0 | 0 | 12 |
| avocado | 126 | 81 | 0 | 10 | 2 | 7 |
| blueberry | 48 | 80 | 8 | 0 | 1 | 11 |
| banana | 88 | 76 | 3 | 0 | 1 | 20 |
| grapes | 64 | 83 | 2 | 0 | 1 | 16 |
| orange | 47 | 87 | 2 | 0 | 1 | 11 |

```
6 rows in set (0.00 sec)
```

Stop the MySQL server:

```
zinet stop s-mysql
```

```
CSMP0097I 09.59.54 CPU-B SS-BSS SSU-HPN IS-01
INET0019I 09.59.54 SERVER MYSQL STOPPED+
CSMP0097I 09.59.54 CPU-B SS-BSS SSU-HPN IS-01
INET0051I 09.59.54 MYSQL IS NO LONGER ACCEPTING CONNECTIONS ON
IP - PORT - 00000 PID - 5E590012+
CSMP0097I 09.59.54 CPU-B SS-BSS SSU-HPN IS-01
080729 10:00:54 Note" /usr/libexec/mysqld: Normal shutdown+
CSMP0097I 09.59.54 CPU-B SS-BSS SSU-HPN IS-01
+
CSMP0097I 09.59.55 CPU-B SS-BSS SSU-HPN IS-01
080729 10:00:55 InnoDB: Starting shutdown...+
CSMP0097I 10.00.20 CPU-B SS-BSS SSU-HPN IS-01
080729 10:01:20 InnoDB: Shutdown completed; log sequence number 0 50431+
CSMP0097I 10.00.20 CPU-B SS-BSS SSU-HPN IS-01
080729 10:01:20 Note" /usr/libexec/mysqld: Shutdown complete+
CSMP0097I 10.00.20 CPU-B SS-BSS SSU-HPN IS-01
+
```


Backup a database:

```
zfile cd /test
```

```
CSMP0097I 09.38.06 CPU-B SS-BSS SSU-HPN IS-01
```

```
FILE0003I 09.38.06 cd /test COMPLETED SUCCESSFULLY. NO OUTPUT TO DISPLAY+
```

```
zfile mysqldump food >backup_food.sql
```

```
CSMP0097I 09.38.30 CPU-B SS-BSS SSU-HPN IS-01
```

```
FILE0003I 09.38.30 mysqldump f... COMPLETED SUCCESSFULLY. NO OUTPUT TO DISPLAY+
```

```
zmsql exec-'use food; drop table fruit;'
```

```
CSMP0097I 10.06.18 CPU-B SS-BSS SSU-HPN IS-01
```

```
MSQL0006I 10.06.17 BEGIN DISPLAY OF SQL QUERY RESULTS
```

```
-----  
drop table fruit
```

```
-----  
Query OK, 0 rows affected
```

```
Bye
```

```
END OF DISPLAY+
```

```
zmsql exec-'use food; select * from fruit;'
```

```
CSMP0097I 10.06.26 CPU-B SS-BSS SSU-HPN IS-01
```

```
MSQL0006I 10.06.25 BEGIN DISPLAY OF SQL QUERY RESULTS
```

```
-----  
select * from fruit
```

```
-----  
ERROR 1146 (42S02) at line 1: Table 'food.fruit' doesn't exist
```

```
Bye
```

```
END OF DISPLAY+
```

Restore a database:

```

zfile cat backup_food.sql | dd conv=tounix | mysql food
CSMP0097I 10.26.55 CPU-B SS-BSS  SSU-HPN  IS-01
FILE0002I 10.26.55 START OF ERROR DISPLAY FROM cat backup_food.sql | dd conv...
5.1 records in
5.1 records out
2649 bytes transferred in 1 secs (2649 bytes/sec)
END OF DISPLAY+
zmsql exec-'use food; select * from fruit; '
CSMP0097I 10.26.06 CPU-B SS-BSS  SSU-HPN  IS-01 _
MSQL0006I 10.26.05 BEGIN DISPLAY OF SQL QUERY RESULTS
-----
select * from fruit
-----

```

| Fruit | calories | percent_water | fiber | fat | protein | sugar |
|-----------|----------|---------------|-------|-----|---------|-------|
| apple | 49 | 84 | 2 | 0 | 0 | 12 |
| avocado | 126 | 81 | 0 | 10 | 2 | 7 |
| blueberry | 48 | 80 | 8 | 0 | 1 | 11 |
| banana | 88 | 76 | 3 | 0 | 1 | 20 |
| grapes | 64 | 83 | 2 | 0 | 1 | 16 |
| orange | 47 | 87 | 2 | 0 | 1 | 11 |

```

6 rows in set

Bye _CSMP0097I 10.14.49 CPU-B SS-BSS  SSU-HPN  IS-01cd
FILE0003I 10.14.49 cat /test/b... COMPLETED SUCCESSFULLY.  NO OUTPUT TO DISPLAY+

```

Logs

Create file /etc/my.cnf to turn on various logs:

```
[mysqld]
log-bin
log
log-error
log-slow-queries
```

```
zfile chown tpfuser1 /etc/my.cnf
```

```
CSMP0097I 11.13.45 CPU-B SS-BSS SSU-HPN IS-01
FILE0003I 11.13.45 chown tpfus... COMPLETED SUCCESSFULLY. NO OUTPUT TO DISPLAY+
```

Start MySQL server (first stop it if already active)

```
zinet start s-mysql
```

```
zfile ls /usr/mysql/data
```

```
CSMP0097I 11.37.20 CPU-B SS-BSS SSU-HPN IS-01
FILE0001I 11.37.20 START OF DISPLAY FROM ls /usr/mysql/data
food                mysql                CO0901B.log
golfers             softball            CO0901B.pid
ib_logfile0         test                CO0901B-bin.index _
ib_logfile1         tpf                 CO0901B-bin.000001
ibdata1             CO0901B.err        CO0901B-slow.log
END OF DISPLAY+
```

Display logs using zfile cat or ftp them over to your linux system. For the Binary log, use the mysqlbinlog utility: **zfile /bin/mysqlbinlog CO0901B-bin.000001**

MySQL users:

- MySQL users are created and authorized with the GRANT statement:
`ZMSQL EXEC-'GRANT ALL ON *.* TO user IDENTIFIED BY "mypass" WITH GRANT OPTION;' IP-ip`

Note the use of two single quotes before and after the password.
- MySQL always will create two users for root and the user ID/host specified in the /etc/ftp.mysql.conf
- ZMSQL and ZFILE commands - Defaults to the z/TPF user. To switch default, use the `USER` parameter on the `ZINET` command.
- Local/remote client program calls - User ID can be specified.

Since MySQL users will sometimes default to the z/TPF user, here are some commands that manage the z/TPF user:

- `ZOVFS DISPLAY ALLUSERS` - Will indicate if security is being used.
- `ZOVFS INIT` - Initializes file system security support.
- `ZOVFS ENABLE` - Enables file system security support.
- `ZOVFS MKUSR userid PASSWD password UID uid` - Creates a userid
- `ZPVFS LOGIN userid passwd` and `ZPVFS LOGOUT` - Logs in and logs out of the file system.

Make new MySQL user and grant authority:

```
markcoo@linuxtpf:~> mysql -u newuser -h 9.57.13.86
ERROR 1045: Access denied for user 'newuser1'@'linuxtpf.pok.ibm.com' (using
password: NO)
```

```
ZMSQL EXEC-'GRANT ALL ON *.* TO newuser WITH GRANT OPTION; '
CSMP0097I 21.06.20 CPU-B SS-BSS SSU-HPN IS-01
MSQL0006I 21.06.20 BEGIN DISPLAY OF SQL QUERY RESULTS
-----
GRANT ALL ON *.* TO newuser WITH GRANT OPTION
-----
```

```
Query OK, 0 rows affected
```

```
Bye
END OF DISPLAY+
```

```
markcoo@linuxtpf:~> mysql -u newuser -h 9.57.13.86
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13 to server version: 5.0.27

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql>
```

Do a display from linux with the new user:

```
markcoo@linuxtpf:~> mysql -u newuser -h 9.57.13.86
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11 to server version: 5.0.27
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql> use food; select * from fruit;
```

```
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

```
Database changed
```

| Fruit | calories | percent_water | fiber | fat | protein | sugar |
|-----------|----------|---------------|-------|-----|---------|-------|
| apple | 49 | 84 | 2 | 0 | 0 | 12 |
| avocado | 126 | 81 | 0 | 10 | 2 | 7 |
| blueberry | 48 | 80 | 8 | 0 | 1 | 11 |
| banana | 88 | 76 | 3 | 0 | 1 | 20 |
| grapes | 64 | 83 | 2 | 0 | 1 | 16 |
| orange | 47 | 87 | 2 | 0 | 1 | 11 |

```
6 rows in set (0.00 sec)
```

MySQL application programs:

1. Modify (or create) `makepf.env_my_appl` to point to the `include/` directory for MySQL. Typically, this will be the `opensource/mysql/include` directory.
`ROOTINCDIRS +=(foreach d,$(TPF_ROOT),$d/opensource/mysql/include)`
2. Modify (or create) `my_appl.mak` and allow the application program to pickup MySQL header files by including the environment: `makepf_env :=my_appl`
3. Add the following line to the `.mak` file to link the MySQL functions:
`LIBS :=CMYL`
4. Create source program in ASCII mode or EBCDIC mode.
 - ASCII mode – Add the following to the `.mak` file to include the ASCII environment (sets flag for ASCII mode): `makepf_env +=ctoe`
 - EBCDIC mode – Add the following to the `.mak` file to only include ASCII library to use conversion routines (does not set flag for ASCII mode):
`LIBS += CTOE`. Any text passed to MySQL needs to be converted to ASCII and any text returned from MySQL will be in ASCII.

ASCII mode

```
#include <stdio.h>
#include <stdlib.h>
#include <mysql.h>

int main(int argc, char ** argv)
{
    MYSQL *connection;
    MYSQL mysql_conn;
    int rc, i;

    mysql_init(&mysql.conn);

    /*Establish a connection based on the first input parm (argv[0] is the program name)*/
    connection = mysql_real_connect(&mysql_conn,argv[1],NULL,NULL,NULL,,0,NULL,0);
    if (connection == NULL)
        printf("Error connecting to MySQL server: %s\n",mysql_error(&mysql_conn));

    rc = mysql_query(connection,"CREATE DATABASE IF NOT EXISTS tpftest");
    if (rc != 0)
        printf("Error issuing CREATE DATABASE: %s\n",mysql_error(connection));

    rc = mysql_query(connection, "CREATE TABLE IF NOT EXISTS tpftest.t1 (t2 int unsigned, PRIMARY KEY
    (t2));");
    if (rc != 0)
        printf("Error issuing CREATE TABLE: %s\n",mysql_error(connection));

    for (i=0; i<10; i++)
    {
        char insertbuffer[128];
        sprintf(insertbuffer,"INSERT INTO tpftest.t1 values (%d);",i);
        rc = mysql_query(connection,insertbuffer);
        if (rc != 0)
            printf("Error issuing INSERT INTO: %s\n",mysql_error(connection));
    }
    exit(0);
}
```


EBCDIC mode

```
#include <atoc.h>

int ABCD()
{
/* MY EXISTING Z/TPF CALLS.. */

iconv_init();
{
MYSQL * connection;
MYSQL mysql_conn;
int rc;
int i;
char *hostname="my.tpf.host.com";
char * ascii_hostname=e2a_string(hostname);

mysql_init(&mysql_conn);

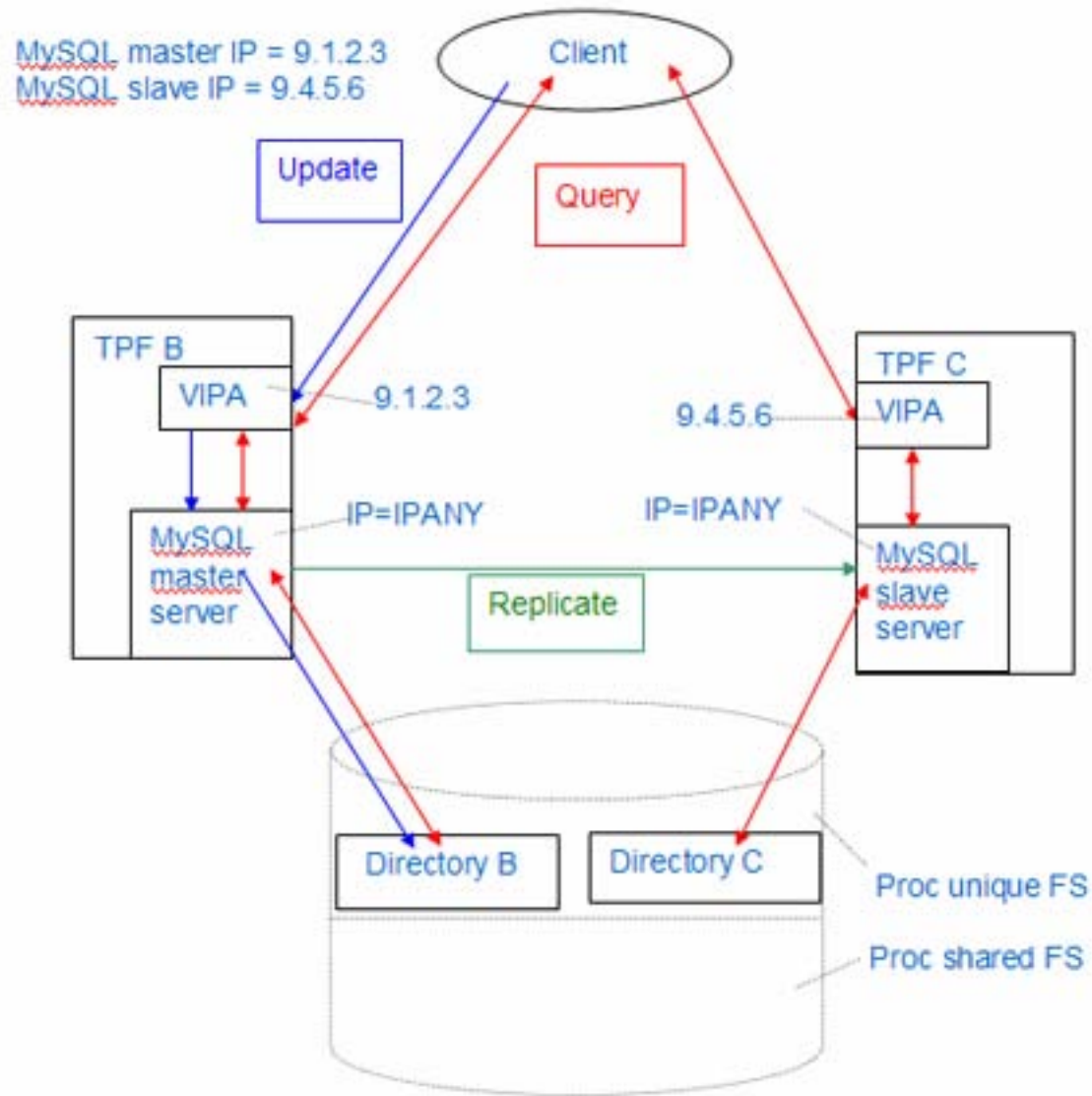
/* Establish a connection based on the first input parm (argv[0] is the program name)*/
connection = mysql_real_connect(&mysql_conn,ascii_hostname,NULL,NULL,NULL,,0,NULL,0);
if (connection == NULL)
    printf("Error connecting to MySQL server: %s\n",a2e_string(mysql_error(&mysql_conn)));

rc = mysql_query(connection,e2a_string("CREATE DATABASE IF NOT EXISTS tpfctest;"));
/* The rest of the mysql code .... */
}

/* MY EXISTING Z/TPF calls.. */
}
```

MySQL replication

Set up in my.cnf or command line



MySQL configuration options

- MySQL variables are set in a number of ways:
 - /etc/my.cnf – Server start time
 - The **XPARM** parameter on the **ZINET** command – Server start time
 - The **SET** statement – Any time server is active from MySQL client or on z/TPF using **ZMSQL EXEC-'SET variable_name=variable value'**.
- Thread Cache: Set by **thread_cache**. Controls the number of threads (ECBs) dedicated for server processing.
- Query Cache:
 - **query_cache_size** – Size of cache. Set to 0 disables cache. Minimum value about 40K.
 - **query_cache_limit** – Max size cached by single query.
 - **query_cache_type** – 0 = off; 1 = on (except if **SELECT SQL_NO_CACHE**); 2 = demand (only if **SELECT SQL_CACHE**).
- Max Connections: Set by **max_connections**. This also allows you to control the number of ECBs used by the server.

Timeslicing – If you want to change the defaults, use **ZTMLS ALTER IMYSQL**

ZTMSL display

CSMP0097I 16.00.15 CPU-B SS-BSS SSU-HPN IS-01

TMSL0002I 16.00.15

EXISTING TIME SLICE NAMES ON FILE

IBMHIPRI IBMLOPRI IBMINDEF IBMPARSE IMYSQL

END OF DISPLAY+

ZTMSL display IMYSQL

CSMP0097I 16.00.31 CPU-B SS-BSS SSU-HPN IS-01

TMSL0003I 16.00.31

TIME SLICE ATTRIBUTES FOR NAME IMYSQL ON FILE

MAXECB- 50 MAXTIME- 0 MINSUSP- 10 RUNTIME- 50 SLICES- 0

END OF DISPLAY+

Other MySQL topics:

- Storage engines:
 - MyISAM – maps databases to directories
 - InnoDB – maps tables to file space in fixed files
 - Federated – points to tables on a remote MySQL server
 - Memory – Non-persistent
- Stored procedures – consistent interface for applications.
- Triggers – update another table based on update to current table.
- User defined functions – extend MySQL functionality (can use with TPF data).
- z/TPF MySQL supported utilities:
 - `/bin/mysql` - MySQL client program
 - `/bin/mysqltest` - MySQL test program
 - `/bin/mysqldump` - Dumps MySQL tables to an SQL file
 - `/bin/mysqlshow` - Shows MySQL databases, tables, and columns
 - `/bin/mysqladmin` - Administers a local or remote MySQL server
 - `/bin/mysqlbinlog` - Displays the contents of a MySQL binary log file and relay log files
- Character set support

Trademarks

- **IBM** is a trademark of International Business Machines Corporation in the United States, other countries, or both.
- **UNIX** is a registered trademark of The Open Group in the United States and other countries.
- **Linux** is a trademark of Linus Torvalds in the United States, other countries, or both.
- **MySQL®** and the MySQL logo are registered trademarks of MySQL AB.
- Other company, product, or service names may be trademarks or service marks of others.
- **Notes**
- Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
- All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
- This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
- All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
- Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the 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.
- Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.
- This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.