



Oracle TimesTen in-memory database integration

September 2014

Author:
Jakub Žitný

Supervisor:
Miroslav Potocký

CERN openlab Summer Student Report 2014

Project Specification

The objective is to build an rpm/script/puppet module that will easily deploy TimesTen in-memory database on existing server/cluster. Create script configuring TimesTen in-memory database for usage with specific database/RAC and creating step-by-step document (Twiki+Snow KB) on how to get required data cached in a simple way. Ultimate outcome will be to have a new service to deploy TT caching easily on any puppetized DB server.

Abstract

TimesTen is in-memory database from Oracle with ability to be attached as a cache to existing Oracle Database. The installation process of TimesTen requires a lot of configuration to be done and although Oracle provides some installation scripts to simplify that, one still needs to take a lot of steps and time to set everything up. This work explores the TimesTen configuration options, proposes the solution for automating as much of the setup as possible and presents the easiest ways to build a working in-memory cache layer for Oracle.

Table of Contents

1	Introduction	4
2	TimesTen installation procedure	5
2.1	Pre-install procedures	5
2.2	Main installation	6
2.3	Post-install procedures	7
2.4	Cache setup	7
2.5	Uninstall	8
3	Automated installation	8
3.1	Wrapper 'ttdeploy'	9
3.1.1	Usage	9
3.1.2	Supported platforms	10
3.1.3	Internals	10
3.1.4	Development	10
3.2	RPM package	10
3.3	Puppet	11
4	Conclusion	11
A	Manpages of 'ttdeploy'	13
B	Procedures for cache setup for Oracle DB	16

1 Introduction

One of the tools that manages most of the data at CERN is Oracle Database. The IT-DB group at CERN works on various improvements for Oracle Database and one of the performance ones is using TimesTen in-memory database as a caching layer.

TimesTen itself is standalone database product from Oracle. It is an in-memory relational SQL database that can be used as regular database for applications, can be backed up and can be replicated. The main advantages of TimesTen are simplicity and speed. Oracle claims the simple design of TimesTen is due to the fact that all data resides in memory which leads to the promised speed up, even more significant than fully-cached Oracle Database. This leads to the most interesting use-case of TimesTen that is TimesTen as a caching layer for Oracle Database. In this case, TimesTen instance is deployed on separate server (or servers) and application communicates only with TimesTen. When TimesTen is deployed on the same machine as application server, significant network latency is lost for even faster application – database communication. TimesTen caching layer can be read-only and writethrough.

Although TimesTen is supposed to be simple and easy to use, the installation and configuration resembles more the complex setup of Oracle Database than just MySQL, MongoDB or Redis.

This work dives into the configuration options of TimesTen and proposes a solution for automating TimesTen integration for CERN environment as well as generic setup for Red-Hat Linux derived distributions.

2 TimesTen installation procedure

TimesTen installation requires several prerequisites to be met. System administrator has to set up kernel parameters to customise shared memory, semaphore and large page behaviour, manually create directory for TimesTen instance information and also create system user and group for administration of TimesTen instance. After preparing these, system administrator can use bundled installation script from Oracle to install TimesTen. This installation script can be executed in interactive or batch mode and requires answers for several questions. When installation script finishes, TimesTen daemon is launched. Before customising the instance or setting up caching it is encouraged to set environment variables for TimesTen instance administrating user.

2.1 Pre-install procedures

Considering the instance needs, system administrator might need to set up 4 kernel parameters – *SHMMAX*, *SHMALL*, *SEM* and *NR_HUGEPAGES*⁸.

SHMMAX and *SHMALL* are related to shared memory. *SHMMAX* is the maximum size of a single shared memory segment in bytes and should be calculated by estimating the following:

$$\text{PermSize} + \text{TempSize} + \text{LogBufMB} + 64 \text{ MB}$$

These are TimesTen connection attributes that set the sizes of the TimesTen permanent memory partition, temporary memory partition, and log buffer. The sum of these when using default values is *136 MB* plus *64 MB* allowance for overhead^{6,7}. However, this might need to be recalculated if *PermSize* or *LogBufMB* is changed.

SHMALL is system-wide total size of shared memory segments in pages and the value is similar to the same value on system for Oracle database. It should be lower than the physical memory on the system. Usually there should be enough space for OS memory operations, e.g. 1 GB for Linux, and the rest can be dedicated to *SHMALL* for TimesTen⁵.

SEM parameter specifies kernel semaphore settings. *SEM* itself is made of four values. The first value is *SEMMSL*, the maximum number of semaphore per array. The second one, *SEMMNS*, is the maximum number of semaphores system-wide. The third value, *SEMOPM*, is the maximum number of operations per semop call, and the last one is the maximum number of arrays¹⁰. TimesTen itself uses 155 semaphores, plus one for each connection⁴ so *SEMMSL* has to reflect this. *SEMMNS* might satisfy the product of *SEMMSL* and *SEMMNI* although it is not necessary¹⁰.

NR_HUGEPAGES is the number of huge pages for the system. This value depends on hugepage size and of course on the size of physical memory available on the system. We showed how to calculate the total size of shared memory segments. In order to use huge pages for all shared memory, the number of hugepages should be the *SHMALL* divided by the size of huge pages which is usually *2 MB* by default. To use huge pages for the whole system simply divide the size of physical memory by huge page size.

All of the custom kernel parameter values are on Red Hat distributions to be entered into */etc/sysctl.conf* and applied with command *sysctl -p*.

After preparing the kernel, the system needs user account and group for TimesTen administration. This user has to be the owner of newly created directory */etc/TimesTen/* and he has to have all per-

missions on it. The installation script will use this to store information about the TimesTen instances, as there may be more than one of them on one system.

2.2 Main installation

Inside the TimesTen installation package from Oracle, there is bundled installation script which prepares all the scripts, configuration files and other files on the system. The script offers interactive and batch installation mode and is optimised for several operating systems. Besides Linux, it is Solaris, HP-UX and AIX. Figure below shows all possible options for the installation script.

```
setup.sh [-install] [-uninstall [doc]] [-batch [<filename>]]
[-record <filename>] [-installdoc] [-quickstart]
[-help] [-verbose]
    -install          Install TimesTen
    -uninstall        Un-install TimesTen
    -batch            Install TimesTen without having to respond to prompts
                    If filename is specified, read from the file
    -record           Install TimesTen and record response to prompts
    -installDoc       Install documentation
    -quickstart       Install the TimesTen Quick Start directory
    -help             Display this help message
    -verbose          Display extra installation information
```

During interactive installation, user needs to address options to customise it. Among them he needs to choose the instance name for TimesTen instance, the location of TimesTen files, daemon files, log files and data files, the port where TimesTen will listen for incoming connections, whether he wants to install documentation and Quick Start Sample Programs, whether he wants to set up Oracle Clusterware, whether he wants to enable PL/SQL for TimesTen and several others. For automated batch installation, he needs to record this configuration to answer file. An example for answer file is shown below. The installation script is also able to record these answers interactively. The complete list of options and their descriptions is available in *Installing TimesTen* section of Oracle TimesTen In-Memory Database Installation Guide³.

```
Please choose an instance name for this installation:node
Is this correct:y
Which would you like to install:1
Where would you like to install the tt1122 instance of TimesTen:/home/ttadmin
Where would you like to create the daemon home directory:/home/ttadmin/TimesTen/tt1122/info
Would you like to specify a different location for the daemon logs:n
Do you want to use the default port number for the TimesTen daemon:y
Please enter a unique port number for the TimesTen daemon:53496
Restrict access to the the TimesTen installation to the group 'ttadmin':y
Would you like to enable PL/SQL for this instance:y
Please enter a value for TNS_ADMIN (s=skip):/usr/lib/oracle
What is the TCP/IP port number that you want the TimesTen Server to listen on:53497
Do you want to install the Quick Start Sample Programs and the TimesTen Documentation:n
Would you like to install the documentation (without the Quick Start Sample Programs):y
Where would you like to create the doc directory:/home/ttadmin/TimesTen/tt1122/doc
Would you like to use TimesTen Replication with Oracle Clusterware:n
```

Under the hood, script *setup.sh* is just wrapper around perl script *install.pl* that executes most of the installation. Script *setup.sh* is under 200 lines of code long, it checks if it's executed on supported OS and passes all arguments to *install.pl* if so. There is also an option to uninstall TimesTen from the system. Uninstall procedure is further described in section 3.5. Script *install.pl* is script almost 7000 LOC long written in Perl, supporting multiple shells and multiple operating systems.

2.3 Post-install procedures

After the main installation finishes, it is required to set up environment for TimesTen administration user. Besides making sure that environment for Oracle connections and paths for binaries and dynamic libraries are properly set, TimesTen-specific variables should be added by sourcing script *ttenv.sh* that is part of the installed scripts and binaries in bin directory. Script *ttenv.sh* is a wrapper for perl script *envcfg* that prepares and exports all needed variables. All tools for administering TimesTen, *ttisql* command-line interface and possible connections to external Oracle databases should be available.

2.4 Cache setup

TimesTen has to be paired with Oracle database in order to set up the caching-layer. Cache users, groups and permissions and additional cache settings must be set up properly inside both databases.

Firstly, the character set for TimesTen instance should be the same as in Oracle database. The character set of Oracle database can be checked by running the following SQL query:

```
select VALUE from NLS_DATABASE_PARAMETERS where PARAMETER='NLS_CHARACTERSET';
```

To set it up for TimesTen instance, the *DatabaseCharacterSet* setting in *info/sys.odbci.ini* has to be edited.

There should be cache manager and possibly cache data user inside the TimesTen instance. Let's say we'll create a cache manager user *cacheadm* and cache data user *cachedata*. We want to grant 'admin' permissions to *cacheadm* and 'create session' permissions to *cachedata*. This is done from the *ttisql* utility connected to TimesTen instance.

```
create user cacheadm identified by ek$3mELY.har0-Pw0;  
grant admin to cacheadm;  
create user cachedata identified by 3v3N.har03r-Pw0;  
grant create session to cachedata;
```

Oracle database has to have the same users as these with appropriate grants as well as properly configured global cache schema. Instructions for this are listed in Appendix C.

To move further a new connection as *cacheadm* to TimesTen instance has to be made using *ttisql*. This time the connection will be connected also remotely to Oracle database. The TNS alias of specific Oracle database is to be entered into *info/sys.odbci.ini* as *OracleNetServiceName*. For this to work, *ORACLE_HOME* has to be set and properly point to a directory with *network/admin/tnsnames.ora* file. Following command is used for the connection:

```
connect "dsn=cachedb1_1122;uid=cacheadm;oraclepwd=cacheadm";
```

The *dsn* argument specifies local TimesTen instance DSN, the *uid* is the name of cache administration user that is the same on TimesTen and Oracle database, *pwd* is the TimesTen cache administration user's password and the *oraclepwd* is the Oracle database's paired cache administration user's password.

To actually pair these users, the following command should be executed:

```
call ttcacheuidpwdset ('cacheadm','cacheadm');
```

TimesTen cache grid provides users with Oracle databases a means to horizontally scale out cache groups across multiple systems with read and write data consistency across the TimesTen databases and predictable latency for database transactions. A cache grid contains one or more grid members that collectively manage application data using the relational data model⁹. At least one cache grid has to be present and associated with the TimesTen instance. For newly created TimesTen instance, following commands do the job:

```
call ttgridcreate ('samplegrid');
call ttgridnameset ('samplegrid');
```

2.5 Uninstall

The default uninstall procedure is even more intertwined than installation. The installation package contains a script called *uninst.sh*, it's almost 1300 lines of Bash code long, but after executing it, it says only the following:

```
$ ./uninst.sh
To uninstall TimesTen, run setup.sh with -uninstall option.
```

However, running *setup.sh* with *uninstall* option requests running it from the location where all the binaries for installed instance are located.

```
$ ./setup.sh -uninstall
To uninstall a specific instance of TimesTen, run the setup.sh script located
within the installation directory that you wish to uninstall.
For example :
/opt/TimesTen/giraffe/bin/setup.sh -uninstall
... will uninstall the instance 'giraffe', located in '/opt/TimesTen/giraffe'.

NOTE: Prior to performing an uninstallation, make sure that your working
      directory is not within the path of the instance you wish to remove.
```

Finally, after executing the same script located inside the bin directory in TimesTen instance location, the procedure seems to work. However, the same *uninst.sh* as the one from installation package is located nearby, in the bin directory, too. Inspecting the working *setup.sh* with *uninstall* option showed that it is this one that does the actual uninstallation.

3 Automated installation

The bottom line of previous chapters was to show the complexity of TimesTen installation. Anyone interested in fast tryout of this database might be repelled by that. Moreover, setting up a cache layer for Oracle database requires some knowledge of proper caching techniques.

This work brings a several simplifications to the installation process of TimesTen. First of all, it presents a wrapper script for installing TimesTen with all dependencies, system-level settings and configuration. The script is called *ttdeploy* and can be use interactively or in batch mode to deploy TimesTen for testing purposes faster. For even more convenient installation, we packed the script and original installation archive into RPM package.

At CERN, the automated deployment of services is crucial for running the organisation and operation of experiments. This work therefore proposes a complete integration of TimesTen into CERN's Puppet/LDAP environment.

3.1 Wrapper ‘ttdeploy’

As described, the default TimesTen installation script handles only the basic tasks of moving needed files to needed locations for TimesTen daemon and administration utilities to function properly. This work presents a wrapper *ttdeploy* to address also pre-install and post-install tasks as well as TimesTen caching setup and uninstallation.

Program *ttdeploy* prepares the whole system for TimesTen deployment, including the setup of kernel parameters, user and group account, permissions, TimesTen installation, environment variables and also desired configuration of TimesTen caching. It is written in Bash and optimised for Linux distributions derived from Red-Hat.

The TimesTen installation archive, however, still has to be downloaded from Oracle website manually.

3.1.1 Usage

As figure below shows, *ttdeploy* can be run in interactive mode, automated batch mode or in uninstall mode. Each of them offers logging, verbose output and also debugging for future enhancements or possible problems.

```
Usage
  ttdeploy [-dvh] [-i tt.archive [-b batch.conf] | -u /path/to/tt.loc]

Examples
  ttdeploy -i tt.archive           interactive installation
  ttdeploy -i tt.archive -b batch.conf  automated batch installation
  ttdeploy -u /path/to/tt_loc        uninstallation from given location

For the list of all options and detailed info see 'man ttdeploy'.
```

Interactive mode is launched by following command, where *tt.package* is the official TimesTen package from Oracle. The package can be zipped tar archive as downloaded from Oracle website (*.tgz* or *.tar.gz*), decompressed archive (*.tar*) or unpacked directory (*linux8664* or other name).

```
ttdeploy -i tt.package
```

Batch mode needs additional argument specifying path to configuration file.

```
ttdeploy -i tt.package -b batch.conf
```

The configuration file is text file with Bash variables. There are mandatory parameters that have to be present in the file and also several additional parameters that are optional. All of these options as well as configuration files are described in Appendix B.

Uninstallation requires the path to TimesTen home directory. The reasons for this are described in section 3.5. So, if TimesTen has been installed to */home/ttadmin/* then it should be uninstalled as follows:

```
ttdeploy -u /home/ttadmin/
```

Verbose execution is supported in all modes with option *-v*. Debug execution is supported with *-d*. Debug passes the *-x* parameter to Bash script. During debug execution all external programs, subshells, functions and commands are displayed to the user².

Exit codes of *ttdeploy* are described in Appendix B.

3.1.2 Supported platforms

Program *ttdeploy* is optimised for Linux distributions derived from Red-Hat, that is Red-Hat Enterprise Linux, CentOS, Scientific Linux, Oracle Linux and others. These operating systems are most used at CERN. Support for other popular operating systems used for Oracle Database, such as Solaris, might be added in the future. Support for systems that might be suitable for particular application server use-cases should be added too. These systems might be preferred for the case where TimesTen and application server are on the same machine.

Program *ttdeploy* is a Bash script, currently without support for other popular shells like *zsh*, *tcsch* or *ksh*.

3.1.3 Internals

As mentioned earlier, program *ttdeploy* is written as Bash script. It uses Bash functions for structured execution, effectiveness and better code readability. The script checks system configuration and tries to find all possible problems before executing anything. There are trap procedures that ensure proper cleanup in case of unexpected termination or termination signals.

During the development, the use of external programs was eliminated in as many cases as possible. Inside the functions the script uses local variables so the environment is not full of temporary or unneeded items. It prints necessary information in colour about execution to *stdout* and error messages to *stderr*. The code is commented and all details are provided in man pages attached in Appendix B.

3.1.4 Development

Program *ttdeploy* together with documentation, example configuration files and rpmbuild scripts are versioned with Git. The code, history and changes are available at <https://github.com/jakubzitzny/ttdeploy> and CERN's internal Git repository with Gitweb web interface at <https://git.cern.ch/web/ttdeploy.git>.

There is a lot of issues that might need to be resolved in the code of *ttdeploy* and other scripts, future developers are welcome to address them. Some of them are noted in *TODO* comments in the code.

After making changes to man pages file *ttdeploy.1* it may be checked with following command.

```
groff -Tascii -man ttdeploy.1 | less
```

3.2 RPM package

To simplify the installation even more, we packed the *ttdeploy* program with configuration files and TimesTen installation archive into RPM package. RPM is a package manager for Linux distributions derived from Red-Hat. RPM packages are easily installed by simple command (below) and the package manager keeps track of everything, allowing easier uninstallation procedure, dependency management and other things.

```
rpm -i package.rpm
```

The RPM packages are compiled with specific configuration files. Three of them are already prepared for the most common use-cases. These are distributed only for CERN internal purposes, as the license agreement for TimesTen does not allow to redistribute it.

There are some use-cases that should be addressed with different configuration files if needed. Updated RPMs may be easily created using a script that is part of the *ttdeploy* Git repository. After changing man pages, *ttdeploy* script or a configuration file RPM can be rebuilt using the script *buil-drpm.sh*.

3.3 Puppet

At CERN, the database servers for development and production purposes are managed by Puppet. Puppet is automation software that defines and enforces the state of server infrastructure, simplifying configuration, provisioning, orchestration and monitoring¹. Database instances at CERN are installed from RPM packages and customised from LDAP .

The Puppet environment is specified in modules, hostgroups and parameters in Hiera. Puppet deals with OS-level configuration. Complex services, for example Oracle databases, are then installed from RPM packages and later customised with scripts and variables from LDAP service for specific purpose. This way, all the different instances of Oracle Database have the same base on every system.

Integrating TimesTen into Puppet environment for wider adoption at CERN has not been decided yet, but when the time comes, we propose using similar deployment structure as the one used for Oracle. There are OS-level prerequisites to be specified. These will be written into Puppet manifests. The stripped-down ‘vanilla’ RPM will be provided for preparing the instance of TimesTen. Post-install procedures and further customisation will be done from LDAP or manually by the database administrator.

The manifest file will have to include the TimesTen administrating user *timesten*, his password, shell, home directory and group with the same name. The existing RPM from previous chapter will have to be stripped down of pre and post-install procedures and it will have to be made upgradable. The database instance DSNs, cache-settings, replication and possible load balancing setting will have to be placed to LDAP service. Other service-level customisations, for example database users and permissions, are to be dealt with by assigned database administrator.

4 Conclusion

This work prepared a foundation for further research on integration of TimesTen into database infrastructure at CERN. It was preceded by the work of Endre Andras Simon that focused on comparing the performance of Oracle Database with and without TimesTen caching¹⁰. The comparison clearly showed that having TimesTen cache layer for high-load Oracle applications is promising. However, there is need for more complex benchmarks comparing fully-cached Oracle Database with TimesTen cache layer in various use-cases that are present at CERN.

Thanks to the tools this work has brought, the installation and cache layer setup is dramatically simplified. Therefore, any further research can focus solely on benchmarks and testing.

Besides simplifying the installation for testing, created installation packages can be used for integrating TimesTen caching into production infrastructure at CERN, if members of database group

decide. Additional details for plugging this service in CERN Puppet environment and LDAP has been provided.

Glossary

Oracle Oracle Corporation (company).

Oracle Database Oracle RDBMS (software product).

TimesTen Oracle TimesTen in-memory database (software product).

TimesTen caching using TimesTen as cache layer for Oracle database.

TimesTen instance deployed TimesTen.

TNS the TNS protocol alias representing the hostname, port, etc. of remote Oracle database, sometimes referred to as Oracle Net Service Name (NSN).

A Manpages of 'ttdeploy'

TTDEPLOY(1)

TTDEPLOY(1)

NAME

ttdeploy - automated deployment of Oracle TimesTen in-memory database

SYNOPSIS

```
ttdeploy [-vd] -i <tt.package>
ttdeploy [-vd] -i <tt.package> -b <batch.conf>
ttdeploy [-vd] -u </path/to/ttloc>
ttdeploy -h
```

DESCRIPTION

TimesTen is in-memory database from Oracle with ability to be attached as a cache to existing Oracle database. The installation process requires a lot of configuration to be done and although Oracle provides some installation scripts to simplify that, one still needs to go through a lot of hassle to set everything up. Program ttdeploy enables the installation to be done interactively or automatically with single command. Ttdeploy prepares the whole system for TimesTen deployment, including the setup of kernel parameters, user and group account, permissions, environment variables and also desired configuration of TimesTen cache, connection to "big" Oracle database and so on.

OPTIONS

```
-i tt.package
    interactive installation from tt.package file
    file can be gz, tar or unpacked directory

-b <batch.conf>
    automated batch installation
    gets answers from given config file e.g. batch.conf
    used in combination with -i
    see below for config file contents

-u /path/to/ttloc
    uninstalls TimesTen from given location (e.g. /home/ttadmin)

-v
    turns on verbose mode

-d
    turns on debug mode
```

-h shows help message

FILES

<batch.conf>
configuration file for batch mode
specified with option -b
the contents of the file is set of bash variables
- mandatory settings:
ORACLE_TNSNAME ORACLE_USERNAME ORACLE_PASSWORD DATABASE_SIZE
- optional settings (recommended):
ADMIN_USERNAME ADMIN_PASSWORD ADMIN_GROUP TT_INSTANCENAME
TT_ADMINNAME TT_ADMINPASS TT_USERNAME TT_USERPASS TT_CACHE_GRIDNAME
- optional settings (needed only in special cases):
TT_VERSION TT_PORT ORACLE_HOME TT_DATASTORE_DIR TT_LOGS_DIR
ADMIN_HOME KERNEL_SEM SHMMAX SHMALL
TTCONFIG_PERM_SIZE TTCONFIG_TEMP_SIZE TTC_SERVER_BASE
each setting (variable) is described in ENVIRONMENT section

<tt.package>
TimesTen installation package
downloaded from <http://bit.ly/1kKkz14>

ENVIRONMENT

ORACLE_TNSNAME
The net service name of Oracle DB where cache will be connected to.

ORACLE_USERNAME
Username for dba connection to Oracle DB.

ORACLE_PASSWORD
Password for dba connection to Oracle DB.

DATABASE_SIZE
The expected size of the database to be cached in GB.
This option will be used to set up nr_hugepages kernel parameter.

ADMIN_USERNAME
The username for system user for TimesTen administration.

ADMIN_PASSWORD
The password for system user for TimesTen administration.

ADMIN_GROUP
The group for system user for TimesTen administration.

TT_INSTANCENAME
The unique instance name for current TimesTen instance.

TT_ADMINNAME
The username for TimesTen db administration user.
This user has to be the same as cache admin in Oracle.

TT_ADMINPASS
The password for TimesTen cache administration user.

TT_USERNAME
The username for TimesTen cache data user.
This user has to be the same as cached data owner in Oracle.

TT_USERPASS
The password for TimesTen db data user.

TT_CACHE_GRIDNAME

The name for TimesTen cache grid.

TT_VERSION

Default value is 11.2.2 - the current release of TimesTen.

TT_PORT

Default value is 53396.

For more instances on single machine this has to be changed.

ORACLE_HOME

This variable is expected to be taken from environment.

If not, it needs to be specified to point to directory tree with tnsnames.ora. ORACLE_HOME=/home/user/ if there is tnsname.ora in /home/user/network/admin/. If not found, ttdeploy will try to find tnsnames.ora in /ORA/dbs01/oracle/product/rdbms.

TT_DATASTORE_DIR

The location for TimesTen data.

The default location is /ORA/dbs03/data.

For each instance, new directory will be created (e.g. /ORA/dbs03/data/tt1122). It is recommended for this location not to be under TimesTen directory tree.

TT_LOGS_DIR

The location for TimesTen logs.

The default location is /ORA/dbs02/logs.

It is recommended for this location not to be under TimesTen directory tree. It is recommended for this location to be on different device as TT_DATASTORE_DIR.

ADMIN_HOME

The home folder for system user for TimesTen administration.

Default is /home/\$ADMIN_NAME.

KERNEL_SEM

Kernel semaphore settings consisting of SEMMSL SEMMNS SEMOPM SEMMNI.

SEMMSL has to be changed for increased number of connections.

TimesTen uses 155 SEMMSL + one for each connection. Minimum value is 128.

After deciding on SEMMSL, you have to recalculate the other values.

More information is at <http://bit.ly/1u0q7aB>.

SHMMAX Maximum size of shared memory segment.

The value is calculated as PermSize + TempSize + LogBufMB + 64 MB.

More information is at <http://bit.ly/1u0qMZH>.

SHMALL Total size of shared memory segments for the whole system.

More information is at <http://bit.ly/1u0qMZH>.

TTCONFIG_PERM_SIZE

Indicates the size in MB of the permanent memory region for the database.

More information is at <http://bit.ly/1u0rxSn>.

TTCONFIG_TEMP_SIZE

Indicates the total amount of memory in MB allocated to the temporary region. More information is at <http://bit.ly/1u0rFkT>.

TTC_SERVER_BASE

Default value is ttLocalHost.

EXIT STATUS

- 0 ok
- 1 script has been run under non-privileged account
- 2 script has been run from unsupported shell
please use bash

```
3      bad command-line option(s)
4      missing one or more dependency tools
5      invalid TimesTen installation package
      please download it from http://bit.ly/1kKkz14
6      invalid user/group/homedir settings given
7      invalid port for TimesTen
      use non-privileged port that is not used by other service
8      connection to Oracle database failed
      please fix the credentials or ORACLE_HOME env var
9      problem with configuration file
      provide mandatory settings (listed above)
10     main TimesTen installation failed
20     execution has been interrupted by SIGINT
50     insufficient amount of memory for chosen database size
127    internal error
```

DEVELOPMENT

Refer to [\url{https://github.com/jakubzitny/ttdeploy}](https://github.com/jakubzitny/ttdeploy).

AUTHOR

Jakub Zitny <[jakub.zitny\(at\)cern.ch](mailto:jakub.zitny@cern.ch)>

version 0.4

August 2014

TTDEPLOY(1)

B Procedures for cache setup for Oracle DB

For TimesTen to be able to cache Oracle DB properly, several procedures have to be run on Oracle DB side as well. A tablespace, a cache schema and cache users with privileges are necessary. These steps require *sysdba* authorisation. Example configuration is available at http://download.oracle.com/otn_hosted_doc/timesten/1121/quickstart/.

After logging in as *sysdba* following command will create a tablespace for managing all the IMDB cache objects.

```
create tablespace ttusers datafile 'ttusers.dbf' SIZE 40M;
```

After installing TimesTen there is *oraclescripts* directory available with PL/SQL scripts for setting up the Oracle DB for TimesTen caching. Script *initCacheGlobalSchema.sql* will init the global schema for cache. After *sysdba* sets up the timesten cache administrating user account, with default tablespace, with unlimited quota, script *grantCacheAdminPrivileges.sql* will grant proper privileges for this user for cache administration.

References

- [1] What is puppet? <http://puppetlabs.com/puppet/what-is-puppet>. Accessed: 2014-09-04.
- [2] bash(1) - linux man page. <http://linux.die.net/man/1/bash>, . Accessed: 2014-09-04.
- [3] Oracle timesten in-memory database installation guide: Installing timesten. http://docs.oracle.com/cd/E21901_01/doc/timesten.1122/e21632/install.htm#TTINS164, . Accessed: 2014-09-04.
- [4] Oracle timesten in-memory database installation guide: Semaphores (linux). http://docs.oracle.com/cd/E21901_01/doc/timesten.1122/e21632/install.htm#TTINS141, . Accessed: 2014-09-04.
- [5] Configuring shmmax and shmall for oracle in linux. <http://dbasolutions.wikispaces.com/SHMMAX+and+SHMALL>, . Accessed: 2014-09-04.
- [6] Oracle timesten in-memory database installation guide: Shared memory (linux). http://docs.oracle.com/cd/E21901_01/doc/timesten.1122/e21632/install.htm#TTINS142, . Accessed: 2014-09-04.
- [7] Oracle timesten in-memory database reference: Connection attributes. http://docs.oracle.com/cd/E21901_01/doc/timesten.1122/e21643/attribute.htm#TTREF114, . Accessed: 2014-09-04.
- [8] Oracle timesten in-memory database installation guide: Linux prerequisites. http://docs.oracle.com/cd/E21901_01/doc/timesten.1122/e21632/install.htm#TTINS139, . Accessed: 2014-09-04.
- [9] Oracle timesten application-tier database cache user's guide: Configuring a cache grid. http://download.oracle.com/otn_hosted_doc/timesten/1122/doc/timesten.1122/e21634/prereqs.htm#TTCAC191, . Accessed: 2014-09-04.
- [10] Endre Andras Simon. Evaluation of in-memory database timesten. 2013.