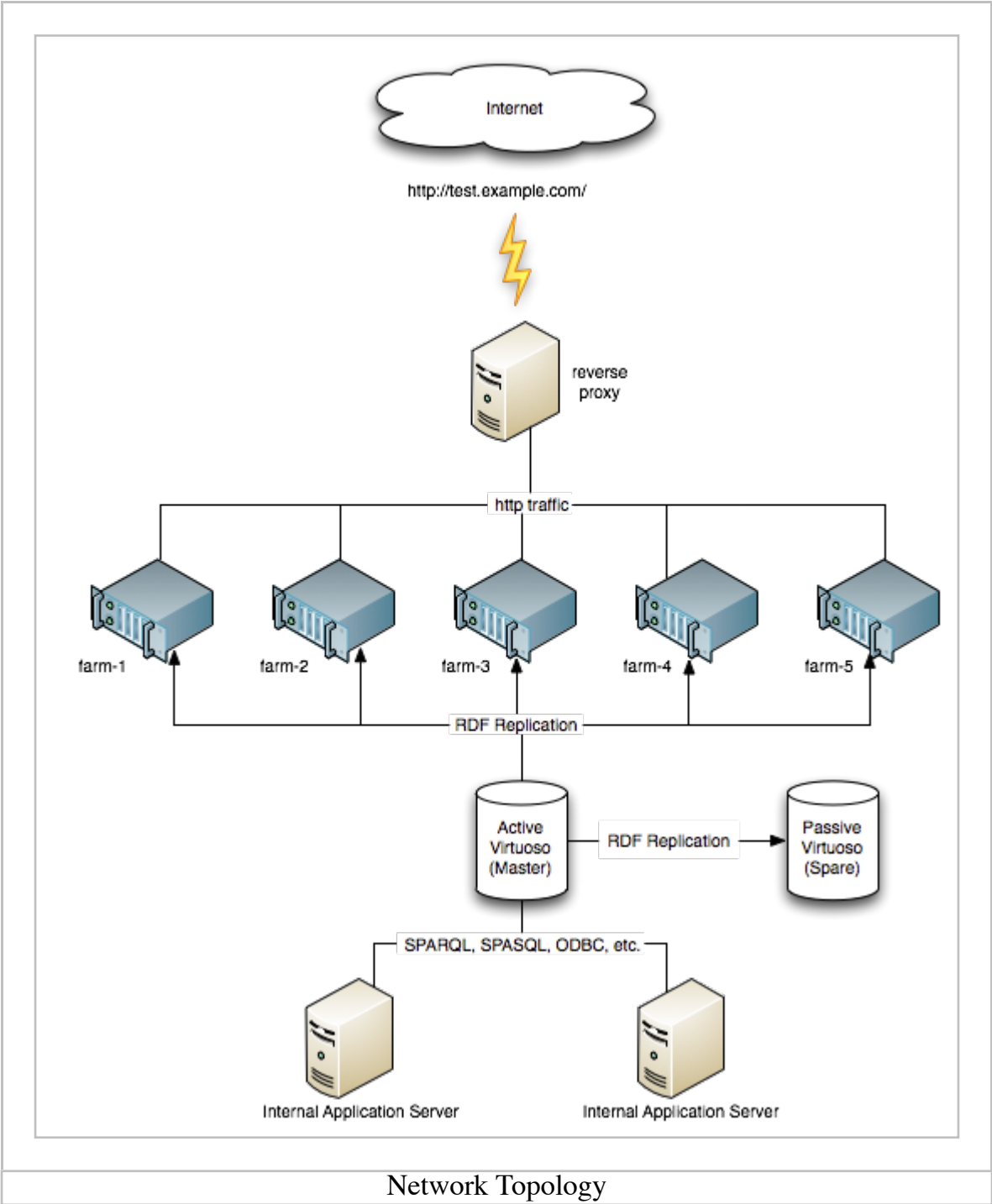


Virtuoso RDF Replication scenarios

Table of Contents

- [Introduction](#)
- [Setup](#)
 - [Installing Virtuoso](#)
 - [Setting up MASTER](#)
 - [Setup SPARE master](#)
 - [Setup FARM-1](#)
 - [Setup FARM-2 from scratch](#)
 - [Setup FARM-3 using blueprint from FARM-1 installation](#)
 - [Setup FARM-4 using clone of FARM-1](#)
- [Further reading](#)



Network Topology

Introduction

In this document we will examine a proposed setup for a back-end server called MASTER which publishes a number of graphs to a set of front-end machines called FARM-1 .. FARM-n and discuss a couple of common scenarios like adding an extra machine to the farm, or replacing a broken instance of MASTER.

In this example we will assume each virtuoso instance running on its own machine, so they can use the same port numbers for both the main server (default 1111) as well as the http port (default 8890) as each machine has an unique IP addresses. In the example we use MASTER-IP and FARM-x-IP which should be replaced by either the real IP address or the DNS name of the machine in question.

Since there will be a reverse-proxy service in front of the farm, all virtuoso instances should have the URIQA Default host set to the outside name for this service. In this example we will use <http://test.example.com> as the web service we are trying to setup.

Setup

Installing Virtuoso

All machines in this setup should be installed with similar installation paths like:

- /opt/virtuoso
- /dbs/virtuoso
- /virtuoso
- ...

The partition should be big enough to have room for the Virtuoso binaries and libraries, the transaction logs, backups and, if you do not want to use the striping feature of Virtuoso, it will need to have room for the main database files as well.

Here are the quick installation steps

1. Login as root
2. Create local user called virtuoso using the chosen installation path as home directory
3. Login as virtuoso
4. Extract `virtuoso-universal-server-6.1.tar` in home directory
5. Run `sh install.sh` to install Virtuoso
6. Remove the file `install.sh` `virtuoso-universal-server-6.1.tar` `virtuoso-server.taz` if not otherwise needed
7. Run `bin/virtuoso-stop.sh` to shutdown this Virtuoso instance
8. Install `virtuoso.lic` for this system in `$HOME/bin` directory

As the replication process needs to make an ODBC connection to the MASTER machine, all machines should have the following information in the `$HOME/bin/odbc.ini`:

```

[ODBC Data Sources]
..
MASTER_DSN = OpenLink Virtuoso

..
[MASTER_DSN]
Driver = OpenLink Virtuoso
Address = MASTER_IP:1111

```

Setting up MASTER

The MASTER machine is the back-end server machine. Various applications feed SPARQL data into this machine it publishes a set of graphs using RDF Replication.

The MASTER machine should ideally be equipped with multiple redundant disks in **RAID-1** or **RAID-6** mode to minimize the risk that a single bad disk takes down the system. From a Virtuoso point of view we will use a combination of online backups combined with checkpoint audit trail to backup the content of the database in a safe way. The online backups, the checkpoint audit trail as well as the replication logs can also be copied to secondary storage using the `rsync` command and can be easily scripted as a cron job.

Changes to database/virtuoso.ini

```

...
[Parameters]
SchedulerInterval      = 1      ; run the internal scheduler every
minute
CheckpointAuditTrail  = 1      ; enable audit trail on transaction
logs
CheckpointInterval    = 60     ; perform an automated checkpoint
every 60 minutes
...
[URIQA]
DefaultHost = test.example.com
...
[Replication]
ServerName   = MASTER
ServerEnable = 1
QueueMax    = 5000000
...

```

Once the MASTER is started using the `bin/virtuoso-start.sh` script we must enable RDF replication before we start add data to the graphs we wish to replicate, so every record is accounted for by the replication process. If there is existing data in the graphs to be published, then this data would need to be added to a subscriber manually since the replication process creates a delta set of changes since publishing was enabled.

To enable publishing of the graph we use the `isql` program to connect to the MASTER

instance:

```
$ isql MASTER-IP:1111
```

and run the following commands:

```
-- enable this instance as a publisher
rdf_repl_start();

-- add graphs to replication list
rdf_repl_graph_ins('http://test.example.com');
```

Next we create a backup directory inside the database directory and setup the online backup, again using the `isql` program:

```
$ cd database
$ mkdir backup
$ isql MASTER_IP:1111
```

and run the following commands:

```
-- clear any previous context
backup_context_clear();

-- start the backup
backup_online ('bkup-#', 1000000, 0, vector ('backup'));
```

The following files can now be backed up using `rsync` or similar tool to another machine:

Files	Description
database/backup/*.bp	the incremental backup files
database/virtuoso.trx	the main transaction log containing the most recent updates to the database that have not been checkpointed into the database
database/virtuosoTIMESTAMP.trx	all the previous transaction logs which can be used to reconstruct the database
database/[]rdf_repl*.log	all the replication logs containing the changes to the published graph

NOTE: Since the database is constantly modified during operation, it is of NO use to backup the `virtuoso.db` using an `rsync` script unless the `virtuoso` instance was shutdown beforehand, or certain extra precautions are taken which we will explain later on.

Setup SPARE master

The SPARE machine is a replica of the MASTER machine. This machine subscribes to the publication of the MASTER to keep an exact match of the RDF graphs, but also publishes

this data without any initial subscribers.

The SPARE machine should ideally be equipped similar to the MASTER machine, with multiple redundant disks in **RAID-1** or **RAID-6** mode to minimize the risk that a single bad disk takes down the system. From a Virtuoso point of view we will use a combination of online backups combined with checkpoint audit trail to backup the content of the database in a safe way. The online backups, the checkpoint audit trail as well as the replication logs can also be copied to secondary storage using the `rsync` command and can be easily scripted as a cron job.

Changes to database/virtuoso.ini

```
...
[Parameters]
SchedulerInterval      = 1      ; run the internal scheduler every
minute
CheckpointAuditTrail  = 1      ; enable audit trail on transaction
logs
CheckpointInterval    = 60     ; perform an automated checkpoint
every 60 minutes
...
[URIQA]
DefaultHost = test.example.com
...
[Replication]
ServerName      = SPARE
ServerEnable    = 1
QueueMax       = 5000000
...
```

We must enable RDF replication before we start add data to the graphs we wish to replicate, so every record is accounted for by the replication process. If there is existing data in the graphs to be published, then this data would need to be added to a subscriber manually since the replication process creates a delta set of changes since publishing was enabled.

To enable publishing of the graph, as well as subscribing to the MASTER, we first start up this Virtuoso instance with `bin/virtuoso-start.sh` and then use the `isql` program to connect to the SPARE instance:

```
$ bin/virtuoso-start.sh
$ isql SPARE-IP:1111
```

and run the following commands:

```
-- enable this instance as a publisher
rdf_repl_start();

-- add graphs to replication list
rdf_repl_graph_ins('http://test.example.com');
```

```

-- connect to master
repl_server ('MASTER', 'MASTER_DSN');

-- start subscribing to __rdf_repl
repl_subscribe ('MASTER', '__rdf_repl', 'dav', 'dav', 'dba', 'dba');

-- start initial replication
repl_sync_all ();

-- add subscription to scheduler
DB.DBA.SUB_SCHEDULE ('MASTER', '__rdf_repl', 1);

```

Next we create a backup directory inside the database directory and setup the online backup, again using the `isql` program:

```

$ cd database
$ mkdir backup
$ isql SPARE_IP:1111

```

and run the following commands:

```

-- clear any previous context
backup_context_clear();

-- start the backup
backup_online ('bkup-#', 1000000, 0, vector ('backup'));

```

The following files can now be backed up using rsync or similar tool to another machine:

Files	Description
database/backup/*.bp	the incremental backup files
database/virtuoso.trx	the main transaction log containing the most recent updates to the database that have not been checkpointed into the database
database/virtuosoTIMESTAMP.trx	all the previous transaction logs which can be used to reconstruct the database
database/□_rdf_repl*.log	all the replication logs containing the changes to the published graph

Note: Since the database is constantly modified during operation, it is of NO use to backup the virtuoso.db using an rsync script unless the virtuoso instance was shutdown beforehand, or certain extra precautions are taken which we will explain later on.

Setup FARM-1

The FARM-1 machine is the first front-end server machine. It subscribes to the publication

of the MASTER instance to keep up-to-date.

The FARM-1 machine can be run on simpler hardware than the MASTER instance. It does not require the same level of redundancy in terms of hard disks etc, as there are a number of these machines running in parallel each capable of returning results to the proxy. If one FARM machine dies, it can simply be taken from the reverse-proxy list, repaired or replaced with a fresh machine before it is added to the list of servers in the reverse proxy. As such it does not need to be backed up separately, although we could make a backup of this installation to quickly install the rest of the identical FARM boxes.

Change the database/virtuoso.ini file:

```
...
[Parameters]
SchedulerInterval      = 1      ; run the internal scheduler every
minute
CheckpointAuditTrail  = 0      ; disable audit trail on transaction
logs
CheckpointInterval    = 60     ; perform an automated checkpoint
every 60 minutes
...
[URIQA]
DefaultHost = test.example.com
...
[Replication]
ServerName      = FARM-1      ; each FARM machine needs to have a
unique replication name
ServerEnable    = 1
QueueMax        = 5000000
...
```

Next we start up the Virtuoso instance using the `bin/virtuoso-start.sh` command and use the `isql` program to subscribe to the MASTER:

```
$ bin/virtuoso-start.sh
$ isql FARM-1-IP:1111

-- connect to master
repl_server ('MASTER', 'MASTER_DSN');

-- start subscribing to __rdf_repl
repl_subscribe ('MASTER', '__rdf_repl', 'dav', 'dav', 'dba', 'dba');

-- start initial replication
repl_sync_all ();

-- add subscription to scheduler
DB.DBA.SUB_SCHEDULE ('MASTER', '__rdf_repl', 1);
```


At this point we can shutdown this Virtuoso instance using the `bin/virtuoso-stop.sh` command and make a copy of the whole virtuoso installation as a blueprint to copy to another FARM-x machine.

Setup FARM-2 from scratch

We can repeat the same steps we did for the FARM-1 machine, and just make sure we use FARM-2 as the replication name in the `database/virtuoso.ini` file and use `FARM-2-IP:1111` as an argument to the `isql` program.

Change `bin/virtuoso.ini`:

```
[Replication]
ServerName = FARM-2
```

Setup FARM-3 using blueprint from FARM-1 installation

Extract the tarred/zipped copy of the installation made at the end of the setup of FARM-1.

Before starting up the instance, we only need to give this instance a unique name for replication:

Change `bin/virtuoso.ini`:

```
[Replication]
ServerName = FARM-3
```

Next we start up the Virtuoso instance using the `bin/virtuoso-start.sh` command and since the subscription records and schedule are already performed in the previous step, we just use the `isql` program to perform a sync against the MASTER:

```
$ bin/virtuoso-start.sh
$ isql FARM-3-IP:1111

-- change replication name
DB.DBA.REPL_SERVER_RENAME ('FARM-1', 'FARM-3')

-- sync against master
repl_sync_all();
```

Setup FARM-4 using clone of FARM-1

If the system has been running for some time, it may not be practical to do a replication from start, so there is an alternative way to setup a new FARM-4 machine.

We can either restore the blue-print backup we make at the end of FARM-1 installation, or we do a fresh installation of virtuoso on the FARM-4 machine.

In both cases we shutdown the virtuoso instance and remove the database, as we are going

to replace this.

```
$ bin/virtuoso-stop.sh
$ cd database
$ rm virtuoso.db virtuoso.trx virtuoso.log virtuoso.pxa
```

Change the database/virtuoso.ini file:

```
...
[Parameters]
SchedulerInterval      = 1      ; run the internal scheduler every
minute
CheckpointAuditTrail  = 0      ; disable audit trail on transaction
logs
CheckpointInterval     = 60     ; perform an automated checkpoint
every 60 minutes
...
[URIQA]
DefaultHost = test.example.com
...
[Replication]
ServerName   = FARM-4          ; each FARM machine needs to have a
unique replication name
ServerEnable = 1
QueueMax     = 5000000
...
```

Next we are going to temporarily disable checkpointing on FARM-1 machine so we can copy its database without risking corruption:

```
$ isql FARM-1-IP:1111

-- disable automatic checkpointing
checkpoint_interval (-1);

-- and do an explicit checkpoint
checkpoint;
```

It is now safe to copy the database across using the rsync command:

```
$ rsync -avz virtuoso@FARM-1-IP:/path/to/virtuoso/database
/virtuoso.db database/virtuoso.db
```

Next we re-enable checkpoint interval on FARM-1:

```
$ isql FARM-1-IP:1111

-- re-enable checkpointing
checkpoint_interval(60);
```

The last step is to start the database:

```
$ bin/virtuoso-start.sh
$ isql FARM-4-IP:1111

-- change replication name
DB.DBA.REPL_SERVER_RENAME ('FARM-1', 'FARM-4')

-- sync against master
repl_sync_all();
```

Further reading

- [Virtuoso RDF Graph Replication Guide](#)
- [Howto setup RDF GRAPH Replication using the Virtuoso API](#)
- [Virtuoso API for replication](#)