

# CyberCluster 1.0

## Technical Documentation

Bringing replication to the masses

# What is CyberCluster?

CyberCluster is a multi-master, synchronous replication solution based on PostgreSQL 8.2. It is the first enterprise-scale software providing high-end replication technology to companies running PostgreSQL.

## What is synchronous replication?

Synchronous replication means that data is kept consistent across all database nodes at any point in time.

This is especially important for many kinds of applications.

Internally Cybercluster synchronizes writes to the database to make sure that all database nodes inside the cluster contain the same data. In addition to that read access is load balanced to make sure that all databases nodes can work highly efficient at the same time. In practice this means that read access can be scaled almost infinitely.

## License

Cybercluster is free software. It can be downloaded and used freely under the terms of the BSD license. This means that no license costs have to be paid.

## Professional services

Cybertec Geschwinde & Schöning GmbH ([www.postgresql-support.de](http://www.postgresql-support.de)) provides professional services for Cybercluster:

- Technical support (24x7)
- Professional training
- Consulting
- Implementation of applications on top of Cybercluster
- Performance Tuning
- Cluster setup and implementation

If your need information about this product or if you are in trouble don't hesitate to visit our website or to send us an email: [office@cybertec.at](mailto:office@cybertec.at)

## Features of CyberCluster

Cybercluster provides a rich set of features. The following overview contains of list of the most important functionalities and gives an insight into what Cybercluster can do for you.

Functionality	Load Balancing	CyberCluster provides load balancing algorithms to make sure that all database nodes can be used during normal operation. This is especially important when it comes to high-performance reading.
	Error Handling	When the cluster software detects an error Cybercluster automatically detaches the broken nodes from the working system and continues its normal operation on the remaining database machines.  As soon as the broken node is repaired, it can be added to the database system safely again. No downtime is needed to perform this operation if at least 3 database nodes are available (1 active node, 1 node for syncing, 1 broken node).
	Recovery	When a cluster DB is started in recovery mode it will automatically resync using an active master.
Replication	COPY	Full support available
	NOW( )	Full support available
	NEXTVAL( ), SETVAL( )	Full support available
	Serial type	Full support available
	Stored procedure	Full support available
	Large Object	Full support available

## Restrictions

(1) When replicating a large object, it needs to be placed in a directory which can be read by all database nodes.

# System Composition

CyberCluster consists of a Load Balancers, Cluster DB Nodes, and Replication Servers.

## Cluster DB nodes

Cluster nodes are the machines processing the actual requests coming in. Whenever a request is received from the client a Cluster DB has to determine whether it is facing a read or a write request. Reads are directly executed by the database and the result is sent to the client. In case a writing operation is detected, the cluster DB sends a message to the replication manager to make sure that data is sent to all remaining database nodes inside the cluster.

## Replication Server

The replication server is the central component of the system. It takes incoming requests from single database nodes and replicates those changes to all database nodes in the system.

If the replication manager detects a problem on a database node it will remove the machine from the list of active database nodes and a log containing a detailed error description will be written to disk. This has to be done to make sure that database nodes can never be out of sync in case of failure. Based on the logs the system administrator can then decide whether it is possible to add the machine to the cluster again.

## Load Balancing Server

Cybercluster contains its own load balancer which can be used to distribute the load inside the cluster. The load on the system is determined by the number of active queries. The machine with the lowest number of active queries will be chosen to perform a new request.

If the load balancer detects a problem on an active database node it automatically detaches the database node from the active system to make sure that no more requests will be sent to a broken machine.

Using load balancers with Cybercluster is optional. If no load balancers are available applications can connect to any database node inside the system. The load balancer itself is in no way involved in replication - it is just used to distribute load.

# Installation of Cybercluster

Two options to install Cybercluster are available:

- Installation from source: Cybertec provides tarballs so that users can build their system themselves
- Binary distributions: For featured platforms binary packages are provided.

## Installation from source

The source package can be compiled and installed the same way as PostgreSQL:

```
tar xzf cybercluster-1.0.1.tar.gz
cd cybercluster-1.0.1
./configure [options ...]
make
make install
```

## Installation of binaries

The binary packages (e.g. for Debian) can be installed with the Linux distribution's usual package management software.

## Configuration

Setting up a cluster is easy and can be done in just a few steps. Some basic requirements have to be taken into consideration, however:

1. All machines inside the cluster (DB nodes, replication servers, load balancer) should have consistent DNS information. In other words, machines must have a consistent name inside the cluster.
2. The database nodes must be allowed to log into each other's „template1“ databases without password.
3. The database nodes must be allowed to log into each other via ssh without password, i.e. via the secure key mechanism.

To setup a cluster it is wise to ask for professional support to make sure that all corner cases can be covered in a professional way.

## Sample Configuration

This section contains a simple sample configuration you can use to getting started quickly.

/etc/hosts file:

```
127.0.0.1      localhost.localdomain  localhost
192.168.0.1   ws001
192.168.0.2   ws002
192.168.0.3   ws003
```

cluster.conf under \$PGDATA (on database node #1):

```
<Replicate_Server_Info>
  <Host_Name>      ws001 </Host_Name>
  <Port>           8001 </Port>
```

```

    <Recovery_Port> 8101 </Recovery_Port>
</Replicate_Server_Info>
<Host_Name> ws002 </Host_Name>
<Recovery_Port> 7001 </Recovery_Port>
<Rsync_Path> /usr/bin/rsync </Rsync_Path>
<Rsync_Option> ssh </Rsync_Option>
<Rsync_Compress> yes </Rsync_Compress>
<Pg_Dump_Path> /usr/bin/pg_dump </Pg_Dump_Path>
<When_Stand_Alone> read_write </When_Stand_Alone>
<Replication_Timeout> 1min </Replication_Timeout>
<LifeCheck_Timeout> 3s </LifeCheck_Timeout>
<LifeCheck_Interval> 11s </LifeCheck_Interval>

```

cluster.conf under \$PGDATA (on database node #2):

```

<Replicate_Server_Info>
    <Host_Name> ws001 </Host_Name>
    <Port> 8001 </Port>
    <Recovery_Port> 8101 </Recovery_Port>
</Replicate_Server_Info>
<Host_Name> ws003 </Host_Name>
<Recovery_Port> 7001 </Recovery_Port>
<Rsync_Path> /usr/bin/rsync </Rsync_Path>
<Rsync_Option> ssh </Rsync_Option>
<Rsync_Compress> yes </Rsync_Compress>
<Pg_Dump_Path> /usr/bin/pg_dump </Pg_Dump_Path>
<When_Stand_Alone> read_write </When_Stand_Alone>
<Replication_Timeout> 1min </Replication_Timeout>
<LifeCheck_Timeout> 3s </LifeCheck_Timeout>
<LifeCheck_Interval> 11s </LifeCheck_Interval>

```

pgreplicate.conf on the replication server:

```

<Cluster_Server_Info>
    <Host_Name> ws002 </Host_Name>
    <Port> 5432 </Port>
    <Recovery_Port> 7001 </Recovery_Port>
</Cluster_Server_Info>
<Cluster_Server_Info>
    <Host_Name> ws003 </Host_Name>
    <Port> 5432 </Port>
    <Recovery_Port> 7001 </Recovery_Port>
</Cluster_Server_Info>
<Host_Name> ws001 </Host_Name>
<Replication_Port> 8001 </Replication_Port>
<Recovery_Port> 8101 </Recovery_Port>
<RLOG_Port> 8301 </RLOG_Port>

```

```
<Response_Mode> normal          </Response_Mode>
<Use_Replication_Log> no        </Use_Replication_Log>
<Replication_Timeout> 1min      </Replication_Timeout>
<LifeCheck_Timeout> 3s         </LifeCheck_Timeout>
<LifeCheck_Interval> 15s       </LifeCheck_Interval>
```

## Starting Cybercluster

Once the cluster has been configured database nodes can be started using the following command (this is the same as PostgreSQL's startup command):

```
pg_ctl -D <datadir> start
```

The replication server can be started before or after all nodes are started up. The following command can be used:

```
pgreplicate -D <datadir>
```

The load balancing server can be started using the following command:

```
pglb -D <datadir>
```

If the PGDATA environment variable is set, the „-D <datadir>” option can be safely omitted as PostgreSQL gets the desired information from the shell directly.

## Recovery of a database node

If a database node has fallen out of the cluster because of a crash (i.e. hardware failure, OS related problems, etc.), it must be re-synchronized with the operating cluster before it is usable again. The following commands do that:

```
pg_ctl -D <datadir> start -o "-R"
pg_ctl -D <datadir> start -o "-u"
pg_ctl -D <datadir> start -o "-U"
```

Any of those three commands can be used. The difference between those command line options are listed below:

1. „-R” is telling Cybercluster to use rsync-based recovery. This method is also called „cold-recovery” because the database node providing the data for recovery will be stopped during recovery. The advantage of this method is that rsync will not see dirty data which is about to be modified by the source database.
2. „-u” forces rsync-based recovery without backing up the original data directories. It speeds up recovery considerably but it should only be used when rsync is already configured properly (e.g. passwordless ssh login is working, etc). This is also a „cold recovery” method.
3. „-U” provides the option to use a pg\_dump-based recovery. Depending on the size of your database this can take quite a long time. However, unlike the other recovery methods, this is a „hot recovery” method, meaning that the source database node will not be stopped during recovery as „pg\_dump” behaves just like a normal database client. This is the only method which can be used if Cybercluster is used to replicate to a different system architecture (e.g. some other CPU architecture, etc.)

## Adding and removing database nodes

Database nodes can be added to a working cluster easily. The commands used for adding a node are the same as the recovery commands.

## Monitoring Cybercluster

After starting the replication server (by default it listens on port 8401) it is possible to monitor the cluster. The port can be changed by modifying the <RPLMonitor\_Port> directive in pgreplicate.conf. Here is an example:

```
<RPLMonitor_Port> NNNN </RPLMonitor_Port>
```

„NNNN“ is the port that will be used for monitoring requests. Monitoring can be performed with the following command:

```
pgrplmon <host> <port>
```

E.g. Using the sample configuration above and the default port:

```
pgrplmon ws001 8401
```

This produces a list of all active cluster nodes similar to the one shown below:

```
# pgrplmon ws001 8401
RPLMonitor v1.0
DB server #1
    Hostname: ws002
    Resolved host: 192.168.0.2
    Port: 5432
    # of started transactions: 0
    # of COMMITed transactions: 0
    # of ROLLBACKed transactions: 0
    # of INSERT statements: 3
    # of UPDATE statements: 1
    # of DELETE statements: 0
    # of DDL statements: 3
DB server #2
    Hostname: ws003
    Resolved host: 192.168.0.3
    Port: 5432
    # of started transactions: 0
    # of COMMITed transactions: 0
    # of ROLLBACKed transactions: 0
    # of INSERT statements: 0
    # of UPDATE statements: 1
    # of DELETE statements: 0
    # of DDL statements: 0
2 active cluster members were found
```

Number of statement per query type are collected for each database node.

NOTE: The information listed here is just for the local database node. It is no clusterwide count.

## Contact information

### **Cybertec Geschwinde & Schönig GmbH**

Gröhrmühlgasse 26

A-2700 Wiener Neustadt

Web: [www.postgresql.at](http://www.postgresql.at)

Phone: +43 / 664 / 3933 974

Email: [office@cybertec.at](mailto:office@cybertec.at)

Cybertec Geschwinde & Schöning GmbH

A-2700 Wiener Neustadt . Groehrmuehlgasse 26 . email: [office@cybertec.at](mailto:office@cybertec.at) . phone: +43 / 664 / 3933 974 . fax: +43 / 820 / 245 544 535

