Univention Corporate Server

Performance guide

Table of Contents

Chapter 1. Introduction

By default UCS is suitable for environments with up to 5,000 users. This document describes configuration modifications which can increase performance in larger environments.

Chapter 2. OpenLDAP and listener/notifier domain replication

As a core element in the operation and administration of a UCS domain, the performance of the LDAP server plays a central role in the overall performance.

2.1. Indexes

Feedback

Comparable with other database systems, OpenLDAP uses indexes about commonly requested attributes. For indexed attributes a search is not performed via the full database contents, but over an optimized subsection.

With newer UCS versions, the indexes are occasionally expanded and automatically activated. The automatic activation can be deactivated using the UCR variable `ldap/index/autorebuild`. In this case, the indexes should be set manually to ensure that there is no loss of performance as a result. The indexes are controlled by the UCR variables `ldap/index/eq`, `ldap/index/pres`, `ldap/index/sub` and `ldap/index/approx`. Once the variables have been changed, the OpenLDAP server must be stopped and the `slapindex` command run.

To determine whether not-indexed variables are used, you can activate OpenLDAP debug level -1 and search for the string 'not indexed' in the log file `/var/log/syslog`. For example:

```
ucr set ldap/debug/level=-1
systemctl restart slapd
grep 'not indexed' /var/log/syslog
```

2.2. Configuration of the database backend

Feedback

The memory mapped database (MDB) has been used for new installations since UCS 4.0. If BDB is still in use, a migration to MDB should be performed for amd64 systems. The database backend can be controlled via the UCR variable `ldap/database/type`. A migration can be performed as follows:

```
systemctl stop slapd
slapcat -l ldif
mkdir /var/lib/univention-ldap/ldap.BACKUP
mv /var/lib/univention-ldap/ldap/* /var/lib/univention-ldap/ldap.BACKUP
ucr set ldap/database/type=mdb
slapadd -l ldif
systemctl start slapd
```

By default the memory mapped database needs more I/O operations than the BDB backend. With the Univention Configuration Registry variable `ldap/database/mdb/envflags` this behavior can be configured. The following flags can be set (multiple values are separated by spaces):

- `nosync` specify that on-disk database contents should not be immediately synchronized with in memory changes. Enabling this option may improve performance at the expense of data security. In particular, if the operating system crashes before changes are flushed, some number of transactions may be lost. By default, a full data flush/sync is performed when each transaction is committed.

- `nometasync` Flush the data on a commit, but skip the sync of the meta page. This mode is slightly faster than doing a full sync, but can potentially lose the last committed transaction if the operating system crashes. If both `nometasync` and `nosync` are set, the `nosync` flag takes precedence.

- `writemap` Use a writable memory map instead of just read-only. This speeds up write operations but makes the database vulnerable to corruption in case any bugs in `slapd` cause stray writes into the memory mapped region.
- `mapasync` When using a writable memory map and performing flushes on each commit, use an asynchronous flush instead of a synchronous flush (the default). This option has no effect if `writemap` has not been set. It also has no effect if `nosync` is set.
- `nordahead` Turn off file read-ahead. Usually the OS performs read-ahead on every read request. This usually boosts read performance but can be harmful to random access read performance if the system's memory is full and the DB is larger than RAM.

2.3. OpenLDAP ACLs

Feedback

Access to the information contained in the LDAP directory is controlled by access control lists (ACLs) on the server side. General information on the configuration of ACLs in UCS can be found in the LDAP chapter of the [ucs-manual].

Nested groups are also supported. The Univention Configuration Registry variable `ldap/acl/nested-groups` can be used to deactivate the nested groups function for LDAP ACLs, which will result in a speed increase for directory requests.

2.4. Univention Directory Listener

Feedback

The Univention Directory Listener can perform safety checks to prevent a user name being added into a group twice. These checks add some overhead to replication and can be deactivated by setting the Univention Configuration Registry variables `listener/memberuid/skip` and `listener/uniquemember/skip` to `no`. Starting with UCS 3.1 the variables are not set and the checks are not activated any longer by default.

Chapter 3. Name Service Cache Daemon (NSCD)

Name resolutions can be cached by the *Name Service Cache Daemon* (NSCD) in order to speed up frequently recurring requests for unchanged data. Thus, if a repeated request occurs, instead of querying the LDAP server, the data are simply drawn directly from the cache.

The size of the cache held by the NSCD is preconfigured for an environment with 5,000 users. If more users or hosts are created, the cache should be enlarged as otherwise it will not be possibly to cache enough entries.

The following Univention Configuration Registry variables can be set:

- `nscd/hosts/size` should be at least the same as the number of all the computers entered in the DNS.
- `nscd/passwd/size` should be at least the same as the number of users.

To allow an efficient cache allocation, the value selected should always be a prime number, in case of doubt the next highest prime number should be selected.

A script can be downloaded from https://updates.software-univention.de/download/scripts/nscdCachesize.sh which suggests corresponding values based on the objects currently included in the system.

Chapter 4. Performance issues during the join process

The size of the UCS domain can have an impact on the duration of the join process. Here is some information how to deal with such problems.

4.1. Samba

Feedback

One of the join scripts for samba requires that the samba connector has synchronized all domain objects into samba. This script has a timeout of 3h (from UCS 4.4-7 on). This is sufficient for normal sized environments. But in large environments this script may hit the timeout and abort the join process. To increase the timeout the Univention Configuration Registry variable `create/spn/account/timeout` can be set prior to the join process.

Chapter 5. Local group cache

By default the group cache is regenerated every time changes are made to a group. This avoids cache effects whereby group memberships only become visible for a service after the next scheduled group cache rewrite (by default once a day and after 15 seconds of inactivity in the Univention Directory Listener. In larger environments with a lot of group changes, this function should be deactivated by setting the Univention Configuration Registry variable `nss/group/cachefile/invalidate_on_changes` to `false`. This setting takes effect immediately and does not require a restart of the Univention Directory Listener.

When the group cache file is being generated, the script verifies whether the group members are still present in the LDAP directory. If only the Univention Management Console is used for the management of the LDAP directory, this additional check is not necessary and can be disabled by setting the Univention Configuration Registry variable `nss/group/cachefile/check_member` to `false`.

Chapter 6. UCS management system

6.1. Disabling automatic search

Feedback

By default all objects are automatically searched for in the domain management modules of the Univention Management Console. This behavior can be disabled by setting the Univention Configuration Registry variable `directory/manager/web/modules/autosearch` to `0`.

6.2. Imposing a size limit for searches

Feedback

The Univention Configuration Registry variable `directory/manager/web/sizelimit` is used to impose an upper limit for search results. If, e.g., this variable is set to 2000 (as is the default), searching for more than 2000 users would not be performed and instead the user is asked to refine the search.

6.3. Adjusting the limit on open file descriptors

Feedback

The Univention Configuration Registry variable `umc/http/max-open-file-descriptors` is used to impose an upper limit on open file descriptors of the ***univention-management-console-web-server***. The default is 65535.

6.4. Vertical performance scaling

Feedback

A single Univention Management Console instance does not use multiple CPU cores by design, therefore it can be beneficial to start multiple instances. Set the following Univention Configuration Registry variables `umc/server/processes` and `umc/http/processes` and restart the Univention Management Console:

```
systemctl restart apache2 \
    univention-management-console-web-server \
    univention-management-console-server
```

The number of instances to configure depends on the workload and the server system. As a general rule of thumb these should not be higher than the machines CPU cores. Good throughput values had resulted in tests with the following combinations:

- 6 CPU cores: `umc/http/processes=3` and `umc/server/processes=3`
- 16 CPU cores: `umc/http/processes=15` and `umc/server/processes=15`
- 32 CPU cores: `umc/http/processes=25` and `umc/server/processes=25`

Note that the number of Apache processes may also need to be increased for the customization to take effect.

Chapter 7. Further services and components

7.1. Apache

Feedback

In environments with many simultaneous accesses to the web server or Univention Portal and Univention Management Console, it may be advisable to increase the number of possible Apache processes or reserve processes. This can be achieved via the UCR variables apache2/server-limit, apache2/start-servers, apache2/min-spare-servers and apache2/max-spare-servers. After setting, the Apache process must be restarted via the command systemctl restart apache2.

Detailed information about useful values for the UCR variables can be found at https://httpd.apache.org/docs/2.4/en/mod/mpm_common.html#serverlimit and https://httpd.apache.org/docs/2.4/en/mod/mpm_common.html#startservers.

7.2. SAML

Feedback

By default, SAML assertions are valid for 300 seconds and must be renewed by clients no later than then to continue using them. In scenarios where refreshing SAML assertions at such short intervals is too expensive (for clients or servers), the lifetime of SAML assertions can be increased via the UCR variable umc/saml/assertion-lifetime. This can be achieved on each UCS system with the role master domain controller or backup domain controller by executing the following commands:

```
ucr set umc/saml/assertion-lifetime=3600
cd /usr/share/univention-management-console/saml/
./update_metadata --binddn USERDN --bindpwdfile FILENAME
```

USERDN has to be replaced with a valid DN of a user, that is member of the group Domain Admins and the file specified by FILENAME has to contain the corresponding password of that user.

It should be noted that increasing the lifetime has security implications that should be carefully considered.

7.3. Squid

Feedback

If the Squid proxy service is used with NTLM authentication, up to five running NTLM requests can be processed in parallel. If many proxy requests are received in parallel, the Squid user may occasionally receive an authentication error. The number of parallel NTLM authentication processes can be configured with the Univention Configuration Registry variable squid/ntlmauth/children.

7.4. BIND

Feedback

BIND can use two different backends for its configuration: OpenLDAP or the internal LDB database of Samba/AD. The backend is configured via the Univention Configuration Registry variable dns/backend. On UCS Directory Nodes running Samba/AD, the backend must not be changed to OpenLDAP.

When using the Samba backend, a search is performed in the LDAP for every DNS request. With the OpenLDAP backend, a search is only performed in the directory service if the DNS data has changed. For this reason, using the OpenLDAP backend can reduce the load on a Samba/AD domain controller.

7.5. Kernel

Feedback

In medium and larger environments the maximum number of open files allowed by the Linux kernel may be set too low by default. As each instance requires some unswappable memory in the Linux kernel, too many objects may lead to a resource depletion and denial-of-service problems in multi-user environments. Because of that the number of allowed file objects is limited by default.

The maximum number of open files can be configured on a per-user or per-group basis. The default for all users can be set through the following Univention Configuration Registry variables:

`security/limits/user/default/hard/nofile`
The hard limit defines the upper limit a user can assign to a process. The default is 32768.

`security/limits/user/default/soft/nofile`
The soft limit defines the default settings for the processes of the user. The default is 32768.

A similar problem exists with the *Inotify* sub-system of the kernel, which can be used by all users and applications to monitor changes in file systems.

`kernel/fs/inotify/max_user_instances`
The upper limit of inotify services per user ID. The default is 511.

`kernel/fs/inotify/max_user_watches`
The upper limit of files per user which can be watched by the inotify service. The default is 32767.

`kernel/fs/inotify/max_queued_events`
The upper limit of queued events per inotify instance. The default is 16384.

7.6. Samba

Feedback

Samba uses its own mechanism to specify the maximum number of open files. This can be configured through the Univention Configuration Registry variable `samba/max_open_files`. The default is 32808.

If the log file `/var/log/samba/log.smbd` contains errors like `Failed to init inotify - Too many open files`, the kernel and Samba limits should be increased and the services should be restarted.

7.7. System statistics

Feedback

The log file `/var/log/univention/system-stats.log` can be checked for further performance analyses. The system status is logged every 30 minutes. If more regular logging is required, it can be controlled via the UCR variable `system/stats/cron`.

7.8. Dovecot high-performance mode

Feedback

Univention Corporate Server configures Dovecot to run in "High-security mode" by default. Each connection is served by a separate login process. This security has a price: for each connection at least two processes must run.

Thus installations with 10.000s of users hit operating system boundaries. For this case Dovecot offers the "High-performance mode". To activate it, login processes are allowed to serve more than one connection. To configure this run

```
ucr mail/dovecot/limits/imap-login/service_count=0
```

If client_limit=1000 and process_limit=100 are set, only 100 login processes are started, but each serves up to 1000 connections – a total of 100.000 connections.

The cost of this is that if a login process is compromised, an attacker might read the login credentials and emails of all users this login process is serving.

To distribute the load of the login processes evenly between CPU cores, mail/dovecot/limits/imap-login/process_min_avail should be set to the number of CPU cores in the system.

7.9. UDM REST API performance scaling

Feedback

A single Univention Directory Manager REST API instance does not use multiple CPU cores by design, therefore it can be beneficial to start multiple instances. By setting the Univention Configuration Registry variable directory/manager/rest/processes the number of processes can be increased. Afterwards the Univention Directory Manager REST API needs to be restarted via

```
systemctl restart univention-directory-manager-rest
```

. The number of instances to configure depends on the workload and the server system. As a general rule of thumb these should not be higher than the machines CPU cores. With directory/manager/rest/processes=0 all available CPU cores are used.

Bibliography

[ucs-manual] Univention GmbH. 2021. *Univention Corporate Server - Manual for users and administrators*. https://docs.software-univention.de/manual-5.0.html.