

Table of Contents

Managing a Large Site.....	1
User Management.....	1
Web Management.....	1
Clear web ownership.....	1
Faster web listing.....	1
Decrease administrative help.....	1
Self-service web creation/deletion/rename.....	1
Eliminating Impractical Operations.....	2
In all public webs.....	2
SiteChanges.....	2
Statistics script use from browser.....	2
Multiple servers.....	2
Locking down the Main web.....	3
Rotating Trash and Sandbox.....	3
User subwebs.....	3
User Masquerading.....	3
Multiple Disks.....	3

Managing a Large Site

There is no limitation with the number of webs and users a TWiki site can have. But there are several considerations needed to run a site having thousands of webs and tens of thousands of users. This topic discusses those.

User Management

The default user management scheme using `TWikiUserMappingContrib` and `TWiki::Users::HtPasswdUser` is not suitable for tens of thousands of users because of the following factors.

- `TWikiUserMappingContrib` maintains the list of users on the `TWiki.TWikiUsersTemplate` topic
- `TWiki::Users::HtPasswdUser` stores user account data on a text file

You need to use e.g. `LdapContrib` for `TWiki::Users::LdapUserMapping` and `TWiki::LdapPasswdUser`. Or you need to implement your own user mapping manager which is scalable. If your environment provides intranet single sign-on and user directory (e.g. LDAP or ActiveDirectory based), it's worth considering getting rid of user registration.

Web Management

Clear web ownership

On a large site, keeping track of web ownership is important for the following reasons.

- To contact the owner about the web's content or configuration
- To prevent abandoned webs from accumulating -- by deleting webs which become orphan

`MetadataRepository` is one way to keep web ownership data. TWiki can enforce existence of web metadata when a web is created.

Faster web listing

If there are thousands of webs, getting the list of all webs takes a long time by directory traversal. It can be made faster by using `MetadataRepository` and enforcing existence of web metadata when a web is created. This prevents directory traversal to get the list of webs. Instead, the metadata repository's webs table is referred to.

Decrease administrative help

The more webs there are on a TWiki site, the more help request the TWiki admins get. To minimize TWiki admins intervention, you can make webs autonomous following the instruction on `AutonomousWebs`. This does not decrease the number of questions from web owners, but TWiki admins can hand off web administrative responsibility to the web owner that way.

Self-service web creation/deletion/rename

Usually, only `TWikiAdminGroup` members can create/delete/rename top level webs, which may generate a good amount of TWiki admin work. By properly implementing `canCreateWeb($cUID, $web)` and `canRenameWeb($cUID, $oldWeb, $newWeb)` of the user mapping handler your TWiki installation use, you can make top level web creation/deletion/rename self-service.

Assuming your TWiki configuration requires web metadata when a new web is created, if you make web creation self-service, you need to make it possible to create metadata of a new web in MetadataRepository.

Eliminating Impractical Operations

If you have thousands of webs, some operations take too long. Here are those costly operations and how to suppress them.

In all public webs

Setting the `{NoInAllPublicWebs}` configuration parameter to true has the following effects

- On the "More topic actions" page, "in all public webs" links are suppressed since they are likely to time out.
- On the WebSearch and WebSearchAdvance topics on all webs, the "All public webs" checkbox is suppressed.

SiteChanges

TWiki.SiteChanges, the topic showing all recent changes across all webs, should be deleted.

Statistics script use from browser

This is not about the number of webs, but about the number of accesses. If there are millions of page views in a month, the `statistics` script takes too long and a time out would occur if it's invoked from browser.

Setting `{Stats}{DisableInvocationFromBrowser}` configuration parameter to true disable invocation of the `statistics` script from browser.

Multiple servers

For higher performance and availability, you may have multiple TWiki servers behind a load balancer for a single TWiki site. By having `$TWiki::cfg{DataDir}` and `$TWiki::cfg{PubDir}` on NFS or other file sharing mechanisms, you can have multiple servers for a single TWiki site easily. If a topic is saved simultaneously by two or more people, on different servers sharing `$TWiki::cfg{DataDir}`, something may break - cases of broken RCS files are reported though their causes haven't been identified.

Even if `$TWiki::cfg{DataDir}` and `$TWiki::cfg{PubDir}` are shared by multiple servers, log files should not be because of the frequency they are updated. For example:

```
use Sys::Hostname;
$TWiki::cfg{LogFile} = '/var/twiki/logs/log%DATE%' . hostname . '.txt';
```

```
logYYMM.SERVER_HOSTNAME.txt
```

If each server has its own log file, the `statistics` script needs to see log files of all the servers to provide real data. If `{Stats}{LogFileGlob}` configuration parameter is set as shown below, the `statistics` script reads access log files matching the file glob (wildcard) **instead of** the file specified by `{LogFileName}`.

```
$TWiki::cfg{Stats}{LogFileGlob} = "/var/twiki/logs/log%DATE%*.txt";
```

Locking down the Main web

If you have tens of thousand of users, to prevent unaccounted-for topics from accumulating in the Main web, you should lock it down. Specifically, you should allow users to have only FirstLast, FirstLastLeftBar, and FirstLastBookmarks. This can be achieved by forbidding ordinary users CHANGE operation in the %USERWEB% web while customizing the isAdmin() method of the user mapping manager to make the user admin of their topics.

Rotating Trash and Sandbox

The Trash web accumulates deleted topics, attachments, and webs. To prevent the Trash web from growing indefinitely, it needs to be cleaned up periodically. One way to achieve it is to rotate Trash just like you do with log files. For example, a new Trash is created from the Trash template after Trash is moved to Trash1 after Trash1 is moved to Trash2 ... after Trash9 is moved to Trash10 after Trash10 is deleted. If you do this rotation daily, you keep deleted topics, attachments, and webs for 10 days.

In addition to Trash, the Sandbox web should be rotated as well. Otherwise, users may accumulate random stuff in Sandbox. Some of them may depend on that random stuff. Rotating Sandbox once a week in the following manner should be appropriate - create a new Sandbox from the Sandbox template after moving Sandbox to Sandbox1 after moving Sandbox1 to Sandbox2 after deleting Sandbox2.

User subwebs

If you have many users, a good number of them may want to have a web for their own use rather than for a team use. In that case, providing them with their own subweb in the Main web might be a good idea.

You can see how to do it at UserSubwebs.

User Masquerading

If you have thousands of webs, TWiki administrators (typically TWikiAdminGroup members) have a big power. Their administrative operations may need to be audited.

UserMasquerading provides a means to minimize the amount of time an administrator exercising the privilege and audit their activities.

In addition, UserMasquerading enables web owners to check access restriction settings on their own.

Multiple Disks

A single disk may not be able to house all webs. UsingMultipleDisks provides a way to use multiple disks.

Related Topics: AdminDocumentationCategory, MetadataRepository, AutonomousWebs, UsingMultipleDisks, UserSubwebs, UserMasquerading

This topic: TWiki > LargeSite

Topic revision: r3 - 2017-01-23 - TWikiContributor



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

LargeSite < TWiki < TWiki

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback

Note: Please contribute updates to this topic on TWiki.org at TWiki:TWiki.LargeSite