Table of Contents

Using multiple disks for DataDir and PubDir

Motivation

A TWiki site may reach a point where a single disk drive cannot house all files. Having PubDir on a different disk from the others doesn't help a lot because on a large site, PubDir takes the most capacity by far. In that case, there is no option but use multiple disks for PubDir.

It's possible to have a single multi-terabyte file system. But that doesn't mean it's practical to use a file system of such scale in your environemnt. You may need to use storage provided by a team you don't have control over. And the size limit might be 1T or 500G bytes.

For reasons described later, it's not possible to utilize additional disks simply by symbolic links without enhancing TWiki code.

Overview

Using multiple disks means that having multiple directories for $TWiki::cfg{DataDir} and $TWiki::cfg{PubDir}. And there need to be a mechanism to specify each web is housed in which.

As you see below, utilizing multiple disks is not trivial. You need to cope with various implications. Because of that, even with this enhancement, it's better to seek ways to have a large single disk space. NOSQL databases may provide good alternatives.

Things to know

How to enable

Setting $TWiki::cfg{MultipleDisks} a true value enables this feature.

On which disk a web resides

This feature requires that metadata repository is in use. This is because a mechanism to specify which web resides on which disk is required. Since this feature is for a large site having thousands of webs, it's not practical to use a topic for the disk specification purposes.

Obviously, the webs metadata file needs to be populated properly. Specifically, each web record needs to have the `disk` field specifying which disk the web resides on. Its value is a non-zero positive integer in decimal or a zero width string. Those values are called disk IDs.

The disk whose ID is "" (a zero width string) is required. When you add a disk, its ID needs to be the current largest ID plus 1. You must not skip a number. You can have the following disk IDs on a TWiki:

- "", 1
- "", 1, 2, 3

But you cannot have the following

- "", 2 (1 is missing)
- "", 1, 2, 4, 5 (3 is missing)

Which disk corresponds to which path

Then, you need to have the data and pub directories specified for each disk. There are two ways to do it. One is by the sites metadata in MetadataRepository and the other is by $TWiki::cfg{DataDir}, $TWiki::cfg{DataDir}, $TWiki::cfg{DataDir1}, $TWiki::cfg{PubDir1}, ...

Sites metadata

If you have the sites metadata in the MetadataRepository, for each site, you are supposed to have the datadir, pubdir, datadir1, pubdir1, ... fields specifying the paths of the data and pub directories in each disk.

If you have the sites metadata and $TWiki::cfg{MultipleDisks} is true, $TWiki::cfg{DataDir} and $TWiki::cfg{PubDir} are not used. But just in case there are bugs in the TWiki core regarding those and there are plug-ins referring to those, you should set those.

$TWiki::cfg{...}

If you don't have the sites metadata in MetadataRepository, you are supposed to have $TWiki::cfg{DataDir}, $TWiki::cfg{DataDir}, $TWiki::cfg{DataDir1}, $TWiki::cfg{PubDir1}, ... set.

Non numeric disk IDs

Actually, non numeric disk IDs are allowed on purpose. You can have disk IDs such as `a`, `b`, and `c` for read-only webs. `TWiki::getDiskList()` doesn't return non numeric disk IDs except the zero width string one ("").

Non numeric disk IDs may be handy for migration to a newer version of TWiki. The new version can show as read-only and include webs not yet migrated without bothering them.

TrashxNx

Each disk need to have its own Trash, which is named TrashxNx, e.g. Trashx1x, Trashx2x, ... For topic, attachment, and web deletion, a proper TrashxNx needs to be selected automatically. %TRASHWEB% is no longer constant and expanded to the proper trash web name depending on the current web.

Every time you add a new disk to TWiki, you need to register and create the trash web of the disk.

You may think Trash1, Trash2, ... are straightforward and desirable rather than Trashx1x, Trashx2x, ... The reason for the naming convention is the Trash web might be aged - every week or every day the new Trash web is created after Trash is renamed to Trash1 after Trash1 is renamed to Trash2, ... after Trash10 is deleted. The aged Trash webs would clash their names with trashes for extra disks.

Attachment URLs

They must be stable

Using multiple disks and exposing them in URLs are two different matters. Topic URLs are not affected by the disk on which topics reside. But attachment URLs may if attachment retrieval is handled by the web server directly without TWiki involved.

Attachment URLs must not be affected by disks they are housed in because if they are affected, an attachment's URL will change when the topic is moved to another disk. For example, an attachment's URL is /pub1/FooWeb/BarTopic/file.png), when FooWebweb moves to the disk 2, it will be /pub2/FooWeb/BarTopic/file.png. This is inconvenient.

If %ATTACHURL% or %PUBURL% is used to refer to the attachment and they are expanded to different paths depending on which disk the topic desides in, it keeps working. Still it is a bad idea because:

1. Users may not follow the practice of using %ATTACHURL% and %PUBURL%.
2. An attachment may be referred to from outside TWiki.

How to achieve it

To achieve attachment URL stability, you need to do either of the following

- Putting symbolic links under the directory $TWiki::cfg{PubDir}
- Rewriting /pub/WEB/TOPIC/FILE to /cgi-bin/viewfile/WEB/TOPIC/FILE

Other implications

So far, things you need to know to use multiple disks are discussed. You don't have know the following thing(s) to set it up, but still they are good to know.

%WEBLIST{...}%

With %WEBLIST{...}%, the "canmoveto" filter (introduced by ReadOnlyAndMirrorWebs) of the webs parameter eliminates webs further. The filter eliminates webs residing in a disk different from the current web.

Example
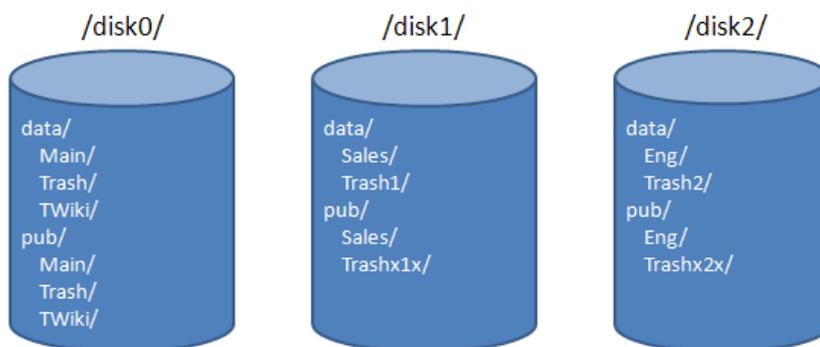
Metadata repository's webs file:

Name	Admin	Disk
Eng	EngAdminGroup	2
Main	TWikiAdminGroup	
Sales	SalesAdminGroup	1
TWiki	TWikiAdminGroup	
Trash	TWikiAdminGroup	
Trashx1x	TWikiAdminGroup	1
Trashx2x	TWikiAdminGroup	2

Non-federated site

If it's on a stand-alone, non-federated site:

```
$TWiki::cfg{DataDir}  = '/disk0/data';
```

They must be stable   3

```
$TWiki::cfg{PubDir}  = '/disk0/pub';

$TWiki::cfg{DataDir1} = '/disk1/data';
$TWiki::cfg{PubDir1}  = '/disk1/pub';

$TWiki::cfg{DataDir2} = '/disk2/data';
$TWiki::cfg{PubDir2}  = '/disk2/pub';
```

Federated sites

If it's on federated sites, Metadata repository's sites file would be:

Name	DataDir	PubDir	DataDir1	PubDir1	DataDir2	PubDir2
am	/disk0/data	/disk0/pub	/disk1/data	/disk1/pub	/disk2/data	/disk2/pub
eu	/d/twiki/data	/d/twiki/pub	/d1/twiki/data	/d1/twiki/pub	/d2/twiki/data	/d2/twiki/pub
as	/twiki/data	/twiki/pub	/twiki1/data	/twiki1/pub	/twiki2/data	/twiki2/pub

Why enhancement is required

Having additional disks and putting symbolic links under PubDir for some webs to off-load the primary disk doesn't work. This is because when a topic is deleted, `topic.txt` and `topic.txt,v` are moved from `DataDir/WEB` to the `DataDir/Trash`. And the `PubDir/WEB/topic` directory is moved to `PubDir/Trash`. If `PubDir/WEB` is a symbolic link a different disk, then moving `PubDir/WEB/topic` to `PubDir/Trash` fails.

For consideration on symbolic link based implementation, please read TWiki:Codev/UsingMultipleDisks#Considerations_on_symbolic_link

Related Topics: AdminDocumentationCategory, MetadataRepository, ReadOnlyAndMirrorWebs

This topic: TWiki > UsingMultipleDisks
Topic revision: r1 - 2012-11-22 - TWikiContributor