Replica Types and Pinning Strategies

Arie Shoshani

November 2000

1. Replica types

Replicas in a data grid have three possible status types depending on their expected usage and function. This status could be used to simplify the interaction with Storage Resource Managers (SRMs), in that pinning of files can be avoided. We also discuss the pinning functions, from simple to sophisticated pinning.

1. A "permanent" status
This type of replica refers to a file stored in a location that is intended to be "permanent", and is usually a location on some tape system. Typically, it is the location that files are stored immediately after they are created. This corresponds to a file stored in "tier 0" in tier architecture.

2. A "quasi-permanent" status
A "replica administrator" who decides where replicas should reside creates this type of replica based on his/her knowledge of the expected use of a file. This file is "quasi-permanent" in that it is likely to stay in the assigned location for long periods of time, but can be removed by the administrator if expected use diminishes. This type of replica will usually resides in "tier 1" in a tier architecture, but can be stored in a "tier 2" level as well.

3. A "volatile" status
This type of replica is created dynamically because of users' requests to process the file. It will stay in place if the demand for it is high, and otherwise will be removed when space is needed. This type of replica is primarily stored in "tier 2" in a tier architecture, but can also be residing in "tier 1" according to the dynamic policy of replica management.

2. To pin or not to pin, that is the question

The usefulness of this distinction of a replica type is in that if a replica is "permanent", it is highly likely to be available for file transfer (i.e. copy from one site to another). A "volatile" replica is more likely to be removed if access demand to it is low. Thus, there is a possibility that the replica is removed in between the time that the client finds about its existence from the replica catalog, and the time that the replica needs to be transferred or accessed. Further, there is some chance that the replica will be removed in the middle of its transfer to another site.

Pinning of a file is a request to a Storage Resource Manager (SRM) at a source site that has that file to keep it for a period of time, or until the file is transferred to a target site.

An SRM can associate a time-out with the pinning request. The length of the time that a file will be pinned is a policy of each SRM, not a grid-wide policy. If the time-out period expires, the file may be marked for removal. After a file is transferred the requestor should normally release the pin. If it is not released, the file may be released by the SRM after the file time-out expires. Similarly, if a file is pinned and never transferred, the file may be released after the time-out expires. Release of a file by the requestor does not necessarily mean that the file is removed; other requestors may still need it, or the SRM may choose to keep it longer.

Regardless of which type of replica is accessed, and whether it is pinned there is a need for the requester to detect and recover from failures. This is because there is no absolute guarantee that a pinned replica will be available at the time of transfer, or that the transfer will complete properly. However, there is a higher probability that a pinned file will be available at the time it is needed. Letting an SRM know that a file will be needed increases the probability that the SRM will keep the file until it is transferred. Letting the SRM know that a file was released, helps the SRM manage its storage resource more effectively.

Obviously, if a replica is permanent, there is no need to pin it. Even if the replica is quasi-permanent, the likelihood of it being removed at the time it is needed is small. However, for volatile replicas pinning would reduce avoidable errors (such as "file not found" or "transfer incomplete"). Thus, pinning is needed only when the probability of failure is sufficiently high, which is the case with volatile replicas.

Volatile replicas are essential in order to manage dynamic storage allocation of replicas; that is, the ability for SRMs to determine dynamically which files to keep in their system at any one time. The goal is to dynamically and automatically migrate replicas to the site where they are used the most. This requires the registration of volatile replicas in the replica catalog soon after they are replicated, so that they are globally known. Similarly, if a file is removed by an SRM, the replica catalog entry will have to be removed as well before the file is physically removed.

Another reason for pinning is for advanced reservation and planning, what is referred to as the "quality-of-service" (QOS). In such a scenario, one may want to reserve space, pin several files, reserve network bandwidth, and transfer the files during this QOS window. For this to work, it is necessary that the source replicas are either permanent, or there are pinned.

3. Pinning strategies

As a practical matter, SRM implementation is simpler if the SRM does not have to keep track of pinning. However, an SRM can still perform "simple pinning" strategies. We'll call the pinning strategies from simple to sophisticated as level-0-pinning, level-1-pinning, etc. We identify 4 such levels such levels.

Level-0-pinning

This is the case that pinning requests are not made at all. SRMs do not mark files as pinned even when a file is being transferred. Rather the SRM finds out if a file is in use or was recently in use to determine which files to remove when space is needed. Using this strategy, the SRM looks for the "oldest" file (or files), usually by a call to the underlying file system to check for files "last touched". A time-out may be associated with this, where a file can be removed if it was not touched for a certain time period. In this case, a release of a file is meaningless. This strategy is sufficient to insure that files that are accessed a lot remain in the cache.

Level-1-pinning

In this case, pinning requests are made, but the SRM does not keep track which client made the pinning request. It has a single time-stamp associated with each file. Originally, the file is time stamped when it is first brought into the SRM space. This time stamp is updated every time some client requests a pin. When space is needed the SRM checks for the oldest time stamp. It then checks when this file was last touched. If it was touched in a time shorter than the time-out period, then the SRM updates the time-stamp and looks for another file with the oldest time stamp. Otherwise, it can be removed.

Similar to level-0-pinning, level-1-pinning insures that files accessed often remain in the cache. But it has the advantage that it is not necessary to check or update "last touched" for all the files. Only the files with the oldest time stamp (i.e. the oldest requested files) are checked.

Using this strategy, there is no requirement that files must be pinned before they are transferred. It is still possible to transfer a file without pinning. The advantage to the requestor to perform a pin is that it is like an advanced reservation, requesting that the replica will be kept in cache for the length of the time-out policy.

Level-2-pinning

This strategy keeps track of which client requested the pin, and when the pin was requested by each client. The main reason for keeping track of pin-per-client is to prevent clients from issuing repeated pins to the same file, thus keeping the file in disk cache indefinitely. Another advantage is that it makes it possible for the SRM to queue transfer requests, rather than refusing them when it is overloaded. This is very important feature, especially to SRMs that require transferring files from tape, such as an HRM. If the request is queued, an estimate can be provided as to the length of time before the file is ready for transfer.

Level-3-pinning

This is a request to pin a file for a certain time and duration. As mentioned above, this is the case where a client wishes to pin a collection of files for a certain time period, when it

has also reserved network bandwidth (QOS), and space in a target site where the files will be moved to. This level of pinning requires negotiations between the requestor and the SRM, as well as a cost model (or client priorities) assigned and managed.

4. What level to use?

Level-0-pinning is essential for the management of volatile replicas. There must be a strategy and a time out policy of which replicas to remove when space is needed. Otherwise, the disk cache will not be used efficiently. It favors files being accessed often. It keeps a file in cache as long as it is being accessed (or transferred). It is simple to implement, but it requires periodic update of the "last touched" time stamp in order to find out the oldest replica to remove.

Level-1-pinning is also simple to implement, but has the advantage that "last touched" need only to be checked for the file with the oldest time stamp.

Level-2-pinning is needed if request queuing is to be performed by the SRM. It is also necessary to prevent abuse by repeated pinning.

Level-3-pinning is needed for advanced reservation and advanced negotiation.

DRMs should support at least level-0-pinning or level-1-pinning. They need to support level-2-pinning if they provide request queuing. HRMs should support level-2-pinning because they need to stage files from tape. Level-3-pinning is much more complex, and needs to be provided only if other QOS services (space reservations and network reservations) are available.

5. Should replica types be registered in the replica catalog?

The knowledge that a replica is permanent or quasi-permanent can be used to avoid requests for pinning and subsequent release of replicas. This can reduce the amount of messages to SRMs. While it is useful to have the replica type in the replica catalog, it is not essential for a correct operation of the system. If this information is not available, pinning requests and releasing of files are simply ignored for permanent files by SRMs provided that they keep track of the type of replica.

A strong argument in favor of keeping replica type in the replica catalog is that if a replica on some disk is permanent one can plan on accessing it without advance reservations. Suppose that QOS network and space reservations were obtained for some files, then the permanent files can be counted on, so that even SRMs that support level-0-pinning only can participate. Further, at the time that a permanent replica is about to be transferred, the requestor may find another volatile replica that it prefers to access (being on a "closer" site or a site that it has a fast network connection to it), pin it, and proceed to access it instead.

The reservation scenario above suggests that it is worth distinguishing between types of permanent replicas: those that are on disk and those that are on tape. A permanent replica on tape cannot be relied on to be available when needed, as it may take a long time to get that replica from tape if there is a long request queue. Reserving a replica on tape requires that the request is made ahead of time, so they it can staged by the time they are needed.

In conclusion, while it is not essential for the replica type to be registered in the replica catalog, it can be useful in reducing message traffic to SRMs, and for better planning of reservations. The cost of maintaining the replica type in the catalog is minimal.