Replica Types and Pinning Strategies

Arie Shoshani

March 2001

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). We also discuss the pinning functions, from simple to sophisticated pinning.

1. A “permanent” type
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 archive. Typically, it is the location that files are stored immediately after they are created. This usually corresponds to a file stored in “tier 0” in tier architecture. A "permanent file is a physical file that can only be created and removed by the owner of the dataset (or the data collection).

2. A “durable” type
An administrator who can make decisions on where replicas should reside creates this type of replica based on his/her knowledge of the expected use of a file. This file is "durable” in that it is likely to stay in the assigned location for a 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. A "durable" file is a physical replica of a file that can only be created and removed by the administrator of the disk cache.

3. A “volatile” type
"Volatile" replicas are created dynamically because of users’ requests to process the files. A volatile file stays in the cache 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. A volatile file is a physical replica of a file that is subject to removal by the DRM according to preset policies.

2. “Pinning” and “two-phase pinning”

The concept of *pinning* is similar to locking. While locking is associated with the *content* of a file to coordinate reading and writing, pinning is associated with the *location* of the file to insure that a file stays in that location. Unlike a lock, which has to be released, a "pin" is temporary, in that it has a time-out period associated with it, and the "pin" is automatically released at the end of that time-out period. The action of “pinning a file” results in a “soft guarantee” that the file will stay in a disk cache for a pre-specified length of time. The length of the “pinning time” is a policy determined by the disk cache

manager. The need for pinning stems from the inherently unreliable behavior of the data grid (because of system failures, network failures, or irresponsible clients). Since we cannot count on pins to be released, we use the pinning time as a way to avoid pinning a file forever.

Pinning is useful in order to make it highly likely that a file needed to be transferred is available at the time of transfer. Suppose that a client at site x finds out that a certain file exists at site y, and wishes this file to be replicated at its site. By the time the file transfer request is issued the file may be removed from site y (because space is needed). This can occur because of network delays, or the system at site y being temporarily busy. To avoid this situation, we use a technique we call "two-phase pinning". First, the client requests that the file will be pinned. After the file is pinned by the storage manager, it is transferred to its destination, and then released by the client. As mention above, given that a client can be unreliable (it may be irresponsible, it may crash, or the network connection may fail) the time-out is used to avoid permanent pinning of the file. After the time-out period elapses there is no guarantee that the file will be kept in the disk cache, although it may be kept longer if another client is using it, or space is not immediately needed.

Two-phase pinning is akin to the well known "Two-phase locking" technique used extensively in database systems, except that the lock is temporary. While two-phase locking is used very successfully to synchronize writing of files and avoiding deadlocks, two-phase pinning is used to prevent files from being removed from a disk cache prematurely. Pinning can also be used to synchronize the requests for multiple files *concurrently*; that is, if the client needs several files at the same time, it can first these files, and only then executing the transfers for all files, releasing them as soon as each is transferred.

We note, that under the widely accepted assumption that file replicas are only read, there is no need for detecting and eliminating deadlocks between the pinned files.

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

The usefulness of the distinction of a replica type is in that if a replica is "permanent" or "durable", 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 is transferred or accessed. Further, there is some chance that the replica will be removed in the middle of its transfer to another site.

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 durable, the likelihood of it being removed at the time it is needed is small. A simple policy that avoids even this possibility is to never remove a durable file, but instead change it into a "volatile" type. For volatile replicas pinning reduces 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 sites 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.

4. Pinning strategies

As a practical matter, SRM implementation is simpler if the SRM does not have to keep track of which files are pinned and by whom. We discuss below several pinning strategies, from simple to sophisticated. The simplest strategy requires the least amount of information on file pinning. The more sophisticated strategies keep track of more information but provide additional functionality. We refer to the pinning strategies from simple to sophisticated as level-0-pinning, level-1-pinning, etc. We identify 4 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 longer.

Level-1-pinning

In this case, pinning requests are made, but the SRM does not keep track which client made the pinning request. Each file has a single time-stamp associated with it. 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. If current time minus the time stamp is larger than the time-out period, then the SRM can remove that file.

Similar to level-0-pinning, level-1-pinning insures that files accessed often remain in the cache longer. 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.

Note that 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 the file will be kept in cache for the length of the time-out policy increasing the probability of a successful transfer.

Level-2-pinning

This strategy keeps track of which client requested the pin, and when the request was made. 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. Also, it makes it possible to service clients according to a policy (such as round robin service) rather than in the order files are requested. This is necessary to prevent a single client from issuing hundreds of requests taking over the disk cache. Thus, requests to pin a large number of files can be queued and allocated according to a fair policy. Another advantage to queuing pinning requests, is that they do not have to be refused when the disk system is overloaded.

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 coordinate the pinning of files for a certain time period, the reservation of network bandwidth (QOS), and the reservation of 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" does not have to be checked. Its main advantage over level-0 is that it provides the requestor with pinning capability for the duration of the time-out.

Level-2-pinning is needed if request queuing is to be performed by the SRM and fair service policies are to be applied. 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 durable 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 and durable 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 or durable 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. This is another aspect of the type of replica that can be recorded in the replica catalog.

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.