Notes.net

Iris Today

Semaphores (Part 2)

by
Carol Zimmet and
Laura Rutherford

Level: Advanced
Works with: Domino 5.0
Updated 12/01/99

Inside this article:

Related links:

Optimizing performance: Semaphores (Part 1)

Analysis of a Sem.Timeouts line

Lotusphere99: Maximizing DB performance, reliability, availability & scalability

Domino 5 Administration Help

Technote: What Does SERVER_NAME_LOOKUP_NOUPDATE=1 Do?

Get the PDF:

A semaphore is to your system what a traffic signal is to an intersection: a mechanism that keeps things running smoothly. Specifically, semaphores ensure that a server completes certain tasks before it begins other tasks.

This article is part two in a two-part series on recognizing, troubleshooting, and preventing semaphore issues. Part one of this series discusses why semaphore timeouts occur and how to troubleshoot the reasons behind them. This article builds on the advice from the first article and presents additional troubleshooting techniques, this time based on real-life experiences. Considerations to keep in mind while designing applications and using LotusScript, as well as R5-specific troubleshooting techniques also appear in this article.

Troubleshooting semaphores: Advice and real experiences

After reading the output from the Sem.Timeout command (see the Correlating a semaphore number with a specific process sidebar from this article and the Analysis of a Sem.Timeouts line sidebar from the first article) or after reading Domino server console messages and parsing out the information displayed, you understand areas of the system where the semaphore timeout occurred. Based on that information, you can try some of the following techniques to continue narrowing the focus area around the semaphore timeout.

Unless otherwise noted, these techniques evolved from R4 experiences. Use them in conjunction with the troubleshooting advice provided in part one of this article. For easy reference, we've organized the techniques according to the area of the system where the semaphore timeouts occur (Database, Server, Indexer). You may recognize some of the techniques as they are applicable not only to semaphore timeout issues, but also to general Domino server administration and troubleshooting.

Check disk configuration and database placement (Database)

Check your disk I/O response rate and configuration to improve end-user response time. Use the platform tools to discern the disk activity rate. Disk configuration analysis means looking at the physical disk allocation, logical disk allocation, relationship to disk controllers, and so on. Once you understand your disk configuration, analyze the relationship of the various Domino components, the location of Domino executables, the Domino data files, the paging file, and the transaction log. For more information, see the *Iris Toda* article Optimizing server performance: I/O subsystems and Lotusphere99 presentation ID603, Maximizing DB performance, reliability, availability & scalability.

To reduce database contention independent of making disk configuration changes, try approaches that may reduce database access contention or improve file access time. Some ways to accomplish such improvements include moving the database to a different disk, a different controller, or a less busy disk. For more information on pinpointing specific database issues, see the section on the **Show DBS** command later in this article.

Try clustering (Database)
You may want to pursue a clustering strategy if the database semaphores 0244 and 0245 appear and you notice the following:

- Many active users on the system (from review of **Server.Users** value -- use the Show Stat Server command to display this value).
- Considerable activity opening databases (from review of the **Show DBS** output -- see the section below for more detail).
- Considerable activity opening views (from review of **LOG_UPDATE**).

By clustering servers, you can distribute some user workload. Please refer to the *Iris Today* articles **Optimizing server performance: Domino clusters Part 1** and **Part 2** for solution ideas.

Redistribute user population (Database)
Another tactic to use when you notice a high count of active users on the system is to move users off the system. This tactic applies to application and mail servers only. For reference, see the posted benchmark reports on the **NotesBench** Web site. These reports provide information on the maximum number of users for a given system configuration executing a specified workload.

Once you move the users to other Domino systems, analyze whether you need to move their associated databases. Also, after you move users off the system, review the change in system performance with respect to databases accessed and the activity rate for the different Domino server tasks (internally and externally developed).

Review the new server console command Show DBS -- R5 only (Database)
The Show DBS command, new in R5, displays useful information about the databases currently in use, such as the number of times a database has been opened, whether the database has been modified, and the number of times a user has had to wait for a lock on the database. See the **How we used output from Show DBS** sidebar to review output from this command and learn how we used this output to help us detect bottleneck situations. Also see the topic "Improving database and Domino Directory performance" in **Domino 5 Administration Help** for additional information on this command.

Review output from Show Database command (Database)
Use the output from Show Database when analyzing a specific database. The output from this command reports a variety of information, including the sizes of the different views in the database (in byte count) and what different objects can be found in the database (for example, documents, forms, and views). The following is an excerpt from the output from the Show Database command:

```
> Show Database demo.nsf

Sample Database

Document Type           Live        Deleted
Documents               2349        66
Info                    0           0
Form                    87          0
View                    80          0

View sizes              Bytes
People                  231,208
Server\Connections      131,880
($ServerAccess)         1,531,440
($Users)                1,721,480
```

```
Marks view                    0
```

For example, we learned from our R4 experiences that the $Users view is large and often rebuilt. From such observations, we employed coding changes to improve upon the lower-level Indexing structures, more typically referred to as the B-Trees. Specifically, we changed the B-Trees storage mechanism so that the B-Trees would be rebuilt in the update areas, not necessarily rebuilding everything all of the time. Consequently, we expect the view rebuild issues to appear less frequently. We do understand that as Domino R5 scales more, users will scale more with us, and the new upper boundaries reached will challenge boundary limits.

Review output from Show Directory console command (Database)
The output from the Show Directory command represents the entries currently found in the Domino database cache. The entries include a database's full path, name, version number, modified time, and state of transaction logging. We provide sample output from the Show Directory command below (the Modified Time column has been omitted in this example):

```
>Show Directory
[017E:0003-01CF]   DbName                        Version   Logged
[017E:0003-01CF]   f:\notefile\schema50.nsf      V5:41     Yes
[017E:0003-01CF]   f:\notefile\Stats.box         V5:27     No
[017E:0003-01CF]   f:\notefile\mail.box          V5:40     No
[017E:0003-01CF]   f:\notefile\mail2.box         V5:41     Yes
[017E:0003-01CF]   f:\notefile\mail1.box         V5:41     Yes
[017E:0003-01CF]   f:\notefile\wsj.nsf           V5:41     Yes
[017E:0003-01CF]   f:\notefile\wrkinst.nsf       V3:17     No
[017E:0003-01CF]   f:\notefile\wpissues.nsf      V5:41     Yes
[017E:0003-01CF]   f:\notefile\webuser.nsf       V3:17     No
[017E:0003-01CF]   f:\notefile\webadmin.ntf      V4:20     No
[017E:0003-01CF]   f:\notefile\webadmin.nsf      V5:41     Yes
[017E:0003-01CF]   f:\notefile\web.nsf           V5:41     Yes
[017E:0003-01CF]   f:\notefile\VinodTes.nsf      V5:41     Yes
[017E:0003-01CF]   f:\notefile\userreg.ntf       V4:20     No
[017E:0003-01CF]   f:\notefile\usenet.nsf        V5:41     Yes
[017E:0003-01CF]   f:\notefile\usegate.nsf       V5:41     Yes
[017E:0003-01CF]   f:\notefile\unixdisc.nsf      V5:41     Yes
[017E:0003-01CF]   f:\notefile\unames4.ntf       V4:20     No
[017E:0003-01CF]   f:\notefile\uiv5info.nsf      V5:41     Yes
[017E:0003-01CF]   f:\notefile\uiteam.nsf        V5:41     Yes
```

If you observe a good amount of contention on a given database (reviewing information from the **Show DBS** command), and see that the database version is from an earlier Domino release, it is time to update the revision of the database to see if the problem goes away.

Check replication strategy (Database)
You should review the replication strategy specified for your topology, as the Replicator task can impose a load on the Domino server. If you start seeing the database's view collection semaphore (030B) and database's view collection queue semaphore (0309), it may be time to locate the database and view in question. You can get this information (the database and view name) from the **LOG_UPDATE** and **LOG_VIEW_EVENTS** output. Activity on a given view can result from user activity or server task requests -- in particular, Domino Directory lookups slow down the replication activity because view rebuild or replication logging takes too long. If you want to defer the rebuild on the Domino Directory, set the NOTES.INI setting SERVER_NAME_LOOKUP_NOUPDATE=1. This variable allows access to

name lookup views in the Domino Directory during the view index process. It also allows access to otherwise locked views during the view process for authentication and mail routing purposes. See Lotus Customer Support Technote #150232, **What Does the SERVER_NAME_LOOKUP_NOUPDATE=1 Server NOTES.INI Parameter Do?** for more information on the use and benefit of this variable.

You can also review the Domino statistics for replica entries (via the Show Stat Replica command) and see the rate of documents being added, deleted or updated. Reviewing this information in synch with the **Show DBS** command may paint a larger picture. At this point, you may notice the number of waiters growing. There is a relationship here, as the database semaphore is needed to successfully execute the add, delete, and update requests from the Replicator. This means in order to perform the replication successfully, the target database needs to be accessed. If this database has a lot of user/server tasks that want access to it, the replication process cannot complete.

Create multiple Mail.Box files (Database)
If you notice the Domino statistic Mail.Waiting growing or at a high number and the database semaphore messages appearing (0244 or 0245), use **Show Stat Mail** to review the Mail Router statistics. For R5, make sure you're taking advantage of the feature to create multiple mail boxes. Refer to Lotusphere99 presentation ID603, **Maximizing DB performance, reliability, availability & scalability**, or **Domino 5 Administration Help** for information on configuring multiple mail boxes.

Using more than one mail box reduces the time wasted waiting for an update to occur, and interfacing with more than one mail box enables the Router to operate more efficiently. There is a trade-off point though, where supporting multiple mail boxes might not bring additional response time benefits or more efficient use of system resources.

You can begin the move to multiple mail boxes by moving from one mail box to two. Then you can increment to three or four mail boxes on a single Domino server if necessary. For information about mail delivery threads, see the **Tell Router Show** command (described below) and review the Lotusphere99 presentation ID601, **Deploying Domino R5 for performance and scalability**.

Adjust NSF_BUFFER_POOL_SIZE (Database)
Once you adjust the NOTES.INI setting **NSF_BUFFER_POOL_SIZE** within the valid range, the current set of users and applications might experience a performance improvement. Keep in mind that adjusting this value is often a short term solution, particularly if you request additional buffer pool space, which requires more overall memory. Thus, less memory is available for additional and future user or application growth. Also, adjusting this value limits the amount of memory available to the operating system's caching requirements.

In Domino R5, we changed the NSF Buffer Pool allocation algorithm to use what the Domino server needed, up to the defaulted or specified amount. The R4 strategy would pre-allocate and reserve the default or specified amount. For example, in R5, if you specify a value of 10,000 for the buffer pool size and only 5,000 is needed, only 5,000 is allocated. In Domino R4, if you specify a value of 10,000 for the buffer pool size and only 5,000 is needed, 10,000 is allocated.

Turn on transaction logging -- R5 only (Database)
New to R5, transaction logging enables you to recover your databases should a system failure occur. Generally, we have observed performance improvements with the recommended use of transaction logging. Why? If the

system experiences semaphore contention issues, the extra performance throughput that transaction logging provides may eliminate or minimize the contention situations. In summary, the sequential nature that transaction logging uses to write out the data translates to a much more efficient method for saving and updating the data that Domino stores. For more information, see the *Iris Today* article **Optimizing server performance: Transaction logging** or **Domino 5 Administration Help**.

Analyze server transactions (Server)
Output from the **Show Trans** command gives you a table of information, including the transaction function types the server executes, the execution frequency (count), and the associated execution times (minimum, maximum, total, average). This information gathers continuously, so if the transaction information hasn't been cleared (via a **Show Trans Reset** command), it should represent a summarized history of transactions the Domino server executed since it started (or since last reset). Refer to **Domino 5 Administration Help** for more information on this command.

From Show Trans output, you can discern a certain system execution profile for your Domino server. You can identify a pattern of transaction types, how often they occur, and which ones take longer to execute. Each transaction has its own unique requirements for CPU, memory, disk utilization and other Domino resources. For example, if the server does a lot of mail processing, more transactions appear in that feature area and less in other areas.

Note that each server executes different types of transactions so you cannot always compare the transaction profiles across different servers, particularly "specialized" servers such as those focused on mail routing or database applications. Some transactions are supposed to take longer to execute, so each transaction needs to be reviewed and analyzed separately.

If you become familiar with Show Trans output, you can determine if excessive time is spent in a general feature area or if a specific transaction is taking a longer than average time to execute. Note the numbers, as there may be a wide range when viewing the "Min" column as compared to the "Max" column, and then factor in the average value. Capturing this data more frequently and noting the widespread data ranges (that is, when the spread occurred) gives you another valuable nugget of information when trying to understand what your Domino server is doing at a given point in time.

Here is sample output from a Show Trans command:

```
Function                     Count   Min    Max   Total  Average
OPEN_DB                        26     10    250   2210     85
OPEN_NOTE                       6     20     60    260     43
UPDATE_NOTE                    26     10    640   1600     61
DB_INFO_GET                     1      0      0      0      0
SEARCH                          3     30    770   1020    340
DB_REPLINFO_GET                23      0     10     70      3
REMOTE_CONSOLE                  7      0     40     60      8
CLOSE_DB                       26      0    140    240      9
CLOSE_COLLECTION                5      0     10     10      2
OPEN_COLLECTION                 5     30   1350   3100    620
READ_ENTRIES                    5      0    310    350     70
NIF_OPEN_NOTE                   1     20     20     20     20
SET_COLLATION                   2      0     10     10      5
READ REPLICATION HISTORY       13      0    240    950     73
WRITE REPLICATION               5      0     10     20      4
HISTORY
GET_MULT_NOTE_INFO_BY_          2     70    190    260    130
UNI
```

In the sample output above, look at the values for the various columns associated with a given server transaction. From these values, you can draw different conclusions. For example, when the maximum value is close to the average value, this implies a fairly constant response time. However, a large range between maximum and average value implies variability in the response time or some internal blocking. It is also interesting to note if the minimum range is close or far from the maximum or average values, which would imply a constant response rate (meaning the system is probably not overutilized). Conversely, a wider response range would imply varying loads on the server, which potentially overutilizes the server.

Note that Web-based interactions do not include entries in the Server Transaction table (this includes HTTP, IMAP, LDAP, POP3, and so on).

Review specific server transactions (Server)
For a quick look at some of the server transactions (from Show Trans output), concentrate on the OPEN_COLLECTION, OPEN_DB and OPEN_NOTE transactions. They have associated "CLOSE_" type transactions (CLOSE_COLLECTION, CLOSE_DB and CLOSE_NOTE). These transactions give you insight into the view, database, and individual note level of activity. The START_SERVER transaction includes the authentication process. So if you want to determine if there is a server authentication issue or if you notice that users attaching to the server are taking a longer time to connect to the server, the START_SERVER transaction is the one to monitor.

The transaction information can be cleared via the **Show Trans Reset** command. We recommend you clear the command when trying to isolate a problem situation, assuming it is a reoccurring problem situation. The command clears the whole transaction table list. As transactions are executed after this point, they are added to the list. The numbers associated with the transactions will then more clearly reflect the activities and execution time for a shorter time range.

Check server tasks (Server)
If contention exists on a database on which you do not normally see contention (that is, not a "well known" database), then dive a little deeper and figure out what impact other Domino server tasks have on the database. Many of the tasks have associated console commands, which can help provide additional insight. The following list provides some console commands you can use for further information about the server tasks and their activity.

- Agent Manager -- Use **Tell Amgr Status** to identify the total number and elapsed time of the Agent Manager runs. You can also check out the Agent Manager statistics, under **Show Stat Amgr** for more details.
- Router -- Use **Tell Router Show** for information on how the transfer threads and delivery threads are configured and executing. The output from the command includes the maximum number of threads and the total number of used and inactive threads. For the transfer thread configuration, you also see the maximum number configured to execute concurrently. You receive the same output from the **Tell Router Show** command as the **Tell Router Status** command.
- Mail -- Use **Show Stat Mail** to identify the current mail status. The output from this command gives you a variety of datapoints relating to the mail item delivery rate and the data throughput for the mail items.

You might think viewing output from these commands is equivalent to looking for a needle in a haystack; however, once you have the general feel for how your system behaves, an uncharacteristic response from one or more of the commands can help guide you to the next course of action, which may reduce the semaphore contention issues or internal bottlenecks.

Adjust SERVER_MAX_CONCURRENT_TRANS value (Server)
The default value for the NOTES.INI setting SERVER_MAX_CONCURRENT_TRANS is 20. The recommendation for changes applies to a single partition and to Domino Release 4.x only *where the threadpool architecture enhancement is not available*. Before beginning any adjustments, become familiar with the System Counter Context Switches/sec (found in NT Perfmon). This value is a good one to watch as changes are made to the transaction rate value. As a general rule, you can increase the SERVER_MAX_CONCURRENT_TRANS value in increments of 20. If the Context Switching number starts to go higher, then you've reached or exceeded the limit of where things should be. We recommend that after every adjustment, you wait a week to make sure the Domino server remains in a stable state.

From an end-user perspective, the value is set too high when the Notes clients are timing out and not successfully connecting to the Domino server. From the Domino server console perspective, the value is too high when after issuing the **Show Users** command, you notice that names are not always listed next to the session. The names are not listed because the authentication process does not successfully complete to provide more background information. Another variation of the same issue occurs when as part of the mail delivery process, the $Users view (needed for successful mail delivery) is accessed for granting permission onto the Domino server. Once successfully accessed, then a name is assigned to the session. The unauthenticated session problem just described is an example of an outcome that can result if a Domino server is forced to go beyond reasonable expectations.

We found another example of users unsuccessfully trying to authenticate in R4 production environments in the rebuilding of the $ServerAccess view (needed for authentication). In a large Domino Directory, it takes longer than average to rebuild the view. When there are 20 users concurrently active and the rest of the user population waits on an event, such as requesting an authentication through the server, this situation causes a continuous loop where Notes clients time out. The Notes clients that have timed out try and reattach to the server, which means they need to authenticate. However, the authentication process cannot complete until the $ServerAccess view is rebuilt. And processes other than the Indexer process cannot gain access to the view, so the Notes clients time out again. Thus, the thread count continues to increase, but no additional users gain access to the server.

Check the Domino Directory and associated view rebuild time (Indexer)
If the Domino server takes a long time to update a view, a few different semaphore issues might start to appear. Typically we have observed that the database (0244), view collection (030B) and/or session table semaphores (0A0B) start appearing. Another symptom, which we describe in greater detail in the first part of this article, is seeing unauthenticated users appearing in the list output from the **Show Users** console command (the command output displays session numbers instead of user names). Again, this type of behavior occurs when the updates take a long time and the users cannot successfully authenticate to establish a session with the Domino server.

The Domino Directory typically has many changes in the form of updates, additions, and deletions. The number of changes also has an impact upon the server's replication activity. The Domino Directory typically keeps information recent and up-to-date, because authentication process uses this information and many user and server initiated tasks access this central file. So you need to decide the delicate balance that works for your system -- find a place between the Domino Directory updating almost to the minute for view rebuilds and document content and the Domino Directory batching up some of the changes and thus not necessarily updating to the minute.

If you want to defer the rebuild on the Domino Directory, set the NOTES.INI setting SERVER_NAME_LOOKUP_NOUPDATE=1. This variable allows access to name lookup views in the Domino Directory during the view index process. It also allows access to otherwise locked views during the view process for authentication and mail routing purposes. See Lotus Customer Support Technote #150232, **What Does the SERVER_NAME_LOOKUP_NOUPDATE=1 Server NOTES.INI Parameter Do?** for more information on the use and benefit of this variable.

Analyze long view updates (Indexer)
We frequently spend time analyzing the view rebuild time. You can also be more proactive on this path. To perform an initial level of troubleshooting, think about the following related points:

- Is it a large view?
- Are there lots of views?
- Are there a lot of updates taking place? A good way to tell if there are "a lot" of updates taking place is to start noticing if a view is constantly being updated or if users complain that they have to wait for a view to be available to them.

For example, if the updates are coming from the Replicator and the Replicator logging options are enabled (output is directed to the Domino Server console and log, see **Domino 5 Administration Help** for more information on enabling), then the Replicator displays the number of notes added/deleted/updated in each database. This information is helpful in terms of reflecting a level of activity, but it doesn't reveal what would trigger a view update. That is, it doesn’t reveal which notes are associated with a given view.

Internally, there is a semaphore protecting each view called the Collection semaphore. If a view is large, with mostly read-type activities performed on the notes (such as minimal new notes being added or existing notes being modified) and unread marks are disabled (which would cause updates to the view behind the scenes) the view update and display of a large view should perform well. Otherwise, you may want to investigate if splitting up the view into two or more parts is a viable option for your application.

In general, these are the important points to consider when addressing a view update concern on your Domino server:

- Are you properly set up to take advantage of the R5 enhancements for Optimized Rebuild? For information, see Lotusphere99 presentation ID603, **Maximizing DB performance, reliability, availability & scalability.**
- Is your Buffer Pool specification initialized to a value, and is it too small?
- Are you taking full advantage of understanding the output from the specifications LOG_VIEW_EVENTS and LOG_UPDATE?

Analyze whether the timeout is a real problem
Finally, as we mentioned in the first article, not all semaphore timeout messages indicate a serious problem. Using information from this and the first article on semaphores, you have techniques to help you learn what issues exist behind the timeout and can, thus, discern if the timeout is a temporary situation or a real problem.

This is an example of a timeout caused by a temporary situation: As part of the evaluation effort to put together this article, we artificially created a semaphore timeout situation by executing the **Show Stat** command frequently on a busy server. As a result, we incurred the semaphore timeout of 0116. The long Show Stat output (especially using the debug version of the code) may block another process as it executes. This type of situation is not serious and should only occur once for each Show Stat console command issued. At the beginning of this article we recommended becoming familiar with the output from the command, but note that we supplied specific parameters to view certain specific subsets of the Show Stat output. Alternatively, if your platform supports it, you can capture the same Domino server statistics as the results from the Show Stat command by looking at Stats & Events database. This database captures the same data automatically on a scheduled interval (performed by the Collector task) and is helpful when performing historical trend analysis.

Considerations for Domino application developers

To reduce or minimize semaphore contention issues, Domino application developers should keep the following techniques in mind when designing applications. The techniques described here also apply to externally developed applications, so you can use them as a checklist for reviewing individually-developed applications and Domino server tasks.

Don’t lock a semaphore for a long period of time

After securing the lock, remain aware of the amount of code executed, the type of operations performed, the function calls made, and the amount of looping until releasing the lock. Analyze all of these things in terms of scalability of operations.

Avoid using a single semaphore that protects a large amount of data

When a semaphore protects a large amount of data, assuming one or more tasks uses it on a write basis, the semaphore becomes a bottleneck within the system. The writer process(es) lines up to access the data structure, and is held back until the semaphore is made available. We recommend a better strategy: to remain aware of the data requirements from a program design viewpoint and have a semaphore protect smaller amounts of data.

Avoid designs that lead to a deadly embrace situation

Use a layered approach when multiple locks are needed. The layered approach method requires that locking starts in a consistent and ordered approach. So, for example, if semaphores were numbered 1 to 10, each task would start by locking the lowest number first, working its way up to the highest numbered.

Have an organized approach towards registering semaphore codes

This approach works well for the larger scale programming efforts, when multiple developers have to incorporate lock/unlock type processing. Having an organized methodology towards recording and reporting semaphores provides a cornerstone for a more maintainable system in the long term.

Use different types of semaphore classes

Have different kinds of semaphore classes: prioritization, specialization based on activity (for example, FIFO ordering, taking advantage of multiple readers) as the program increases in size. To avoid deadly embrace situations, introduce some sort of weighting system associated with a semaphore -- based on the specific semaphore -- to ensure execution of a path in a given sequence or at a higher priority. When there are multiple tasks waiting to gain access to data structures, there are multiple techniques that can be used. One technique is to allow the reader processe(es) to take precedence over the writer process(es). Another tactic is to take advantage of what a specific operating system may have to offer. This tactic requires that you review what the base OS kernel provides or advises in terms of designing the use of semaphores within your program logic.

Build in deadlock detection logic

As part of our debug code, we built in some deadlock detection logic -- for example, logic that detects and reports layering violations. (Note that this overhead processing is not enabled for the production code.) This logic represents an advanced coding technique that is required as the number of developers and the sophistication of the program developed increases. The additional coding detection logic provides payback in the long term, as it more quickly identifies problem situations.

LotusScript: How does it fit in?

LotusScript is designed to be thread-safe, which means it uses semaphores to protect data structures so that each loaded instance of LotusScript is protected. We make the following recommendations for designing to avoid semaphore issues when using LotusScript.

Use special care in use of Lock and Unlock functions

For R5, LotusScript enables you to develop code making use of Lock and Unlock functions. When using these functions, apply special care and planning. For example, if you are not releasing the lock (equivalent to holding a semaphore), you will create your own blockage problem. LotusScript is "helpful" in that when you exit your LotusScript program, the completion logic cleans up any outstanding locks. We use the word helpful carefully in that you should always try to design for minimal cleanup processing outside of your own program. This completion logic is invoked when an end-user presses Ctrl-Break and when normal termination processing occurs. On the flip side, the LotusScript logic does not detect programming anomalies such as an infinite loop so you must take extra steps to ensure that does not occur.

See the section above, “Considerations for Domino application developers,” for more information.

Plan agent processing with the HTTP server

When designing and developing an agent that executes on a large scale production system, review the processing involved when a request initiates an agent that executes on an HTTP server. You need to ensure that the ensuing supporting software and execution is as expected and needed. More specifically, when an agent is triggered, an instance of LotusScript is loaded. If there are many requests issued, resulting in many agent executions (which also translates to the separate LotusScript executions), contention for the same resources occurs. You can control this contention partially through the HTTP Active Thread specification, which you specify in the Domino Directory. So when raising the number of active HTTP threads in your Server Configuration document, you may want to also factor in the implication that as the Active Thread count is raised, the processing support and resource cost for supporting the additional threads also increases. You need to make sure you are achieving your goals of raising system throughput without penalizing all the other types of processing performed. See the *Iris Today* article **Optimizing server performance: HTTP Threads settings** for more information.

General strategies practiced in R5

Based on what we learned from semaphore issues in R4 environments, we employ the following strategies to minimize semaphore issues in R5 environments.

Use a “hash table” approach

Within internal Domino development, we made a concentrated effort to convert internal list structures to an indexed table (or “hash table”). A hash table includes the different types of "queues" of information. As Domino continues to scale, it takes longer to traverse a longer list (which is not a practical approach after a given point in scaling), so the hash table approach eliminates the need to perform an exhaustive search through information.

Practice deploying the latest pre-releases of code

We deploy the latest pre-release of versions of the Domino server software as soon as possible to multiple configurations and user environments within Iris/Lotus/IBM. We consider this debugging effort a high priority so that we can investigate all crashes and serious system conditions. Our rule to investigate and pursue every system issue helps in minimizing the number of scenarios that have potential semaphore contention issues.

Review output from console commands

As mentioned earlier in this article, we review output from the following commands:

- Show DBS
- Show Database
- Show Directory

Conclusion

Armed with information about why semaphore timeouts occur, what techniques help deduce their specific causes, and what steps help minimize their appearance, you are now prepared to analyze your system should semaphore issues arise.

In this article, you learned some troubleshooting techniques built upon information gathered from the first article. You also know the considerations we keep in mind while coding to keep semaphore timeouts to a minimum. And when it comes to using LotusScript and R5, you have some practical advice that can help keep the system running smoothly.

In this and the first article on semaphores, we hope to have eased some of the confusion over their role in your system -- and enabled you to be proactive in the job of troubleshooting timeout issues.

ABOUT CAROL

Carol Zimmet started working at Iris in 1994. She is responsible for evaluating performance and performance tool development on the server team. Carol continues to search for the one-step solution to everyone's performance problems. Carol enjoys bicycling with her kids in the trailer and playing racquetball. She has a longing to return to stained glass!

ABOUT LAURA

Laura worked as a user assistance writer for Lotus until she had her daughter, Kate, in January 1999. Now she spends a good amount of time with some sort of baby food mush on her clothing, daydreaming about the days when "sleeping late" involved an hour later than 7:00 am. Some of Laura's favorite things include her daughter (of course), her husband, her two dogs, and taking long hikes with all of them. Other favorite things include frozen cappuccinos, cheese pizza, and margaritas (plain, on the rocks, and without salt) from Sierras restaurant in Sudbury.

What do you think about this article?

Register Here!

About this Site | **Feedback**
Lotus Home | **IBM Home** | **Iris Home**

[back to" Optimizing server performance: Semaphores (Part 2)"]

Correlating a semaphore number with a specific process (sidebar)

Note: Part one of this article include theAnalysis of a Sem.Timeouts line sidebar. This sidebar builds on that information, providing additional detail.

To view Sem.Timeouts lines, type**Show Stat Sem.Timeouts** at the Domino server consoleIf the system has experienced a semaphore timeout, a message similar to the following appears:

```
Sem.Timeouts=430D:58  030B:28
```

where **430D:58** translates to [semaphore number]:[count of occurrences]

The line indicates that two semaphore timeout types occurred. The first timeout is related to the Indexer, Name Lookup function, and occurred 58 times. The second timeout relates to a database that has large collections to be rebuilt (typically the Domino Directory) and occurred 28 times.

To associate the semaphore number with a particular area or function, see the following table. We also provide you with information about our expectations of these semaphore notifications occurring in R5. Domino developers and Lotus Customer Support provided input as to which semaphores have been the ones under the most contention (that is, they suggested the entries and prioritized the list below). We base our expectations on what we have seen to date on our R5 production servers and on Domino servers deployed by our customers. Quite a few administrators who have migrated to R5 on their servers have told us that their systems are not experiencing any semaphore timeouts.

Semaphore	Description	Occurs in R5?
0244 & 0245	Relates to database and database queue. May appear when you have a large or very active database. Typical indicates when one task is trying to open a database, and another task is using it exclusively.	Rarely
0309 & 030B	Relates to view rebuilding (collection and collection queue). May appear when the database (such as Domino Directory) has very large collections to be rebuilt and the other tasks are locked out.	Rarely
410F	Relates to OS file system (File Descriptor Queue). May appear with many file accesses.	Rarely
430D	Relates to Indexer, Name Lookup function (collection and collection queue). May occur with many name lookup requests from different threads (end user or server task initiated). **Note:** The Collection Group semaphores are only used by the NameLookup internal process and protect the user's list of NameLookup data structures.	Rarely
0A12 & 0A13	Relates to network-level activity (buffered log package and log commit). May appear with a large burst of activity, when reporting of server's activity occurred at	Rarely

	rate faster than server log could update to stay in synch	
4117	Relates to OS system, tracking most recently freed memory. May appear with many requests for memory from different threads.	Rarely
0A0B	Relates to network and session tables. May appear when there are many users connected. Use the **Show DBS** command to reveal the database in question. The output reveals if a problem is from contention on the Domino Directory, which could preven clients from authenticating in a reasonable amount of time. See the Show DBS section in the main article for more information.	This semaphore may occur in R5. It remains a semaphore for a few different situations. Contact Lotus Customer Support if you encounter this semaphore.
4113	Relates to OS file system, handle table movement.	Does not occur in versions R4 and later.
0255	Relates to database (NSF), B-Tree access. May appear when a view is being rebuilt (whose underpinnings are a B-Tree) and there is a heavy demand for the view.	Rarely in R5 (as we optimized the underlying data structures for scalability) but if it occurs, contact Lotus Customer Support.
0924	Relates to server (Database Server Queue). May appeai when there is much database activity concurrently for a given server. Typically seen on application servers	Rarely

For more information, see the following:

- Lotusphere99 presentation ID601,**Deploying Domino R5 for performance and scalability**
- Lotusphere99 presentation ID603,**Maximizing database performance, reliability, availability & scalability**
- Lotus Customer Support Technote #112710,**Semaphores and Semaphore Timeouts**

[back to "Optimizing server performance: Semaphores (Part 2)"]

How we used output from Show DBS (sidebar)

We converted a subset of the output from the **Show DBS** command from one of our R5 Domino servers into a table. Some of the columns of information helped detect bottleneck situations and helped us make optimization decisions before we made the final R5 release. So to understand the output in greater detail, understand that the database entries listed reflect which databases are requested for access; this list is not a superset or a subset of the entries in the Database Cache (see the output from the **Show Directory** command for the Database Cache entry list).

Name	Refs	Mod	FDs	LockWaits	Avg Waits	#Waiters	MaxWaiters
mail\cnoyes.nsf	3	N	1	0	0	0	0
Alice!!mail\hankins.nsf	2	N	0	0	0	0	0
mail\bhankins.nsf	4	Y	3	0	0	0	2
mail\rstephen.nsf	3	N	1	0	0	0	0
kerry\50Status.nsf	3	N	1	0	0	0	1
mail\mvincenz.nsf	3	N	1	0	0	0	0
StrHockey.nsf	3	Y	1	0	0	0	0
mail\rmaffa.nsf	7	N	1	0	0	0	1
clubusy.nsf	1	N	1	0	0	0	5
lotnames.nsf	73	N	20	0	0	0	4
notnames.nsf	79	N	17	0	0	0	1
names.nsf	128	N	17	0	0	0	3

Based on the table above, read the following way we used output from the Show DBS command to help detect semaphore issues:

- Monitor the columns LockWaits and Avg Waits for an indication of contention issues. In the example given above, there are no issues to report. But another command execution for another server may have shown numbers greater than zero for Avg Waits, which then provides a good opportunity to look at which file is the culprit. There is a good chance that file might be the Domino Directory (names.nsf), the Mail.Box file, or a unique database. If, for example, the Mail.Box has an Avg Waits time in the warning area, you may want to increase the number of Mail.Box files configured on your Domino server.
- If the Avg Waits time is in the warning area for a unique database (where "unique" really refers to a database that supports your applications), it is time to explore what application or agent is interacting with it and review the current procedure. A value in the "warning area" is really system-dependent when the value is above 0. Falling back on the original suggestion of getting a feel for your system, if you see a large change from the standard value reported, it is time to take proactive steps. So reviewing the output of the Show DBS command on a regular basis -- especially as it gives you a feel for activity on core databases -- is a good practice to follow.

- Consider the Refs count column, which represents the number of references to the database. You should consider this information within the big picture of understanding how your Domino server performs. Review the Refs count column in conjunction with the #Waiters column. Augment that information with the knowledge gathered about typical system behavior. For example, on a Domino mail server, you should expect the names.nsf file to have a high reference count. If #Waiters count is also growing, this is the point that you should determine that there is an issue that needs further investigation. So for another approach on making use of the available information, those databases that have a higher reference count are databases under contention. Their disk placement as well as view update rate needs to be considered individually.
- Review the #Waiters and MaxWaiters columns to see if there is a backlog of requests to gain access to the database, which can lead to a semaphore contention issue. If that information is steadily or actively growing, it could be an indication of a mounting or immediate problem (respectively).

For a practical observation, we used Show DBS output as part of our development efforts in fine tuning Domino Release 5. In particular, we reviewed Show DBS output to monitor the performance implications of multiple Mail.Box files. While running benchmark tests, we evaluated the impact of increasing the number of available Mail.Box files, going from 1 to 4, and saw the average wait time decrease. The table output given above illustrates an example of a "well-behaving" production system (that is, we did not observe any real problems).

For more information on the Show DBS command, see the topic "Improving database and Domino Directory performance" in **Domino 5 Administration Help**.

Register Here!

About this Site | Feedback
Lotus Home | IBM Home | Iris Home