Opportunistic locking

This article is about opportunistic locking, a little known protocol of the LAN Manager of MS Windows NT/2000 and XP designed to increase performance of network application - in theory! However this protocol imposes a couple of problems for Xbase^{++}/Clipper and even MS Access or Excel applications running in a shared network environment.

In this article I would like to explain the basic idea of oplocks, which problems they implicate for network applications and, last but not least, various strategies how to increase performance and reliability of your network application.

What is opportunistic locking

Opportunistic locks (oplocks) are a characteristic of the LAN Manager networking protocol implemented in the 32-Bit Windows family of operating system environments. Basically, oplocks are guarantees made by a server for a shared logical volume to its clients. These guarantees inform the Client that a file's content will not be allowed to be changed by the server, or if some change is imminent, the client will be notified before the change is allowed to proceed.

So much about the theory, in practice the LAN Manager introduces with oplocks an additional protocol on top of the well known standard file-system level locking protocol. This is done for performance reasons. Only with oplocks the server and its clients are allowed to cache data (read ahead) or defer writes to a file (write behind). In other words, oplocks are introduced to synchronize the data in an environment of multiple workstations, each of which having their own local cache, sharing a file via a file-server. Remember, any workstation has its own cache to increase performance when it comes to file I/O. Imagine a situation where 2 workstations read data (into their local cache) and a single workstation writes data back. As a matter of fact, one workstations cache is out-of-date and holds "old" data. With oplocks the local cache of workstation 2 gets informed about the fact of modification by other workstations and therefore invalidates its local cache.

The problem with oplocks is that they:

- are designed for file-sharing with medium concurrency
- introduce a additional level of overhead
- block clients from performing their operations immediately.

In practice the oplocks introduced by Microsoft are good enough when it comes to sharing files like word-documents or excel spreadsheets in a networking environment. But they fail when it comes to heavy concurrency in environments with file-based databases such as Xbase^{++}, Visual FoxPro and even MS-Access or VB applications with the Jet-Engine. There are other strategies available, such as the ones implemented with the Andrew's file-system which deal with the problem of distributed caches in a network, but for whatever reason Microsoft has decided to go their own route.

The reason for the problems the oplock approach has with file based database systems lies in their specific characteristic how they deal with the file-system. File based database systems:

- impose a heavy concurrency and high probability of data changes anywhere in the file.
- handle concurrency issues on their own at the semantical layer of the database. Not on the file-based level. Remember they lock records, ensure that updates get not lost using sophisticated implicit and explicit locking protocols.
- implement specialized local optimization strategies to handle caching of data and invalidation of cached data. All based on the record or index-page level.

This is where the universal oplock solution with Microsoft LAN Manager and file-based databases conflicts and can serious hamper them in terms of performance and reliability. Even deadlocks produced by the LAN Manager become possible with high concurrency due to its restricted handling of so called worker threads in the context of oplocks. With Windows for Workgroups, IBM LanManager and Novell Netware oplocks problems do not exist because these LAN operating-systems resolve the problems differently or simply have not implemented a optimization strategy such as the one introduced by the oplocks under Microsoft LAN Manager.

Which applications may be affected!

In general database application but also office applications when online collaboration is enabled are affected by the oplock problem when using Windows NT/2000 and XP as the file server. Since Windows NT 4.0 Service Pack 4, 5, 6, or 6a and Windows 2000 a default oplock configuration which is critical to all types of applications which run in a environment of heavy concurrency was introduced.

Not affected:

- Windows NT 4.0 with Service Packs prior of #4
- Windows ME/98/95

Affected:

- Windows NT 4.0 with Service Pack 4 or higher
- Windows 2000
- Windows XP

At best your network performance is sluggish, at worst you encounter problems with an open file shared by the LAN Manager such as:

- The file is in use
- The file is already open
- Unable to write to drive U:
- The network drive U: is no longer available
- Data loss Data corruption - file size inconsistency

How Microsoft reacts to this issue?

You can find various MS Knowledgebase articles about this issue. MS has officially confirmed that the default oplock configuration is a problem for "DOS" style database applications. However, understanding the Microsoft terminology is critical here:

The term "older MS-DOS-based database applications" refers to all file-based database applications. Because they perform a so called heavy concurrency pattern. MS has no strategic interest in resolving this problem, specifically in the context of the MS-Access and Jet-Database Engine users community. Because now they can talk them into using MS-SQL Server instead of resolving the performance and reliability problems. The following MS Knowledgebase articles are of interest to the entire subject:

Q219022 Improving Performance of MS-DOS Database Applications Because Xbase^{++} database engines in the context of Clipper/Comix and SIX interoperability behave against the file system such as a MS-DOS database application this article is valid for DBF/NTX/CDX drivers et al.

Q129202 Explanation of Opportunistic Locking on Windows NT This article describes additional configuration parameters regarding oplocks. The article is valid for NT/2000 and XP.

Q163525 Delay When Saving Word File to Windows NT 4.0 Server Even MS Word and other Office applications may run into performance problems with the oplock solution by Microsoft.

Q126026 Server Optimization in RFCB Caching If you have encountered problems when opening files located on a server. This article describes how to detect if the problem is related to oplocks and the CachedOpenLimit configuration parameter.

Q249799 Slow Network Performance with Service Pack 4, 5, 6, or 6a Valid for Xbase^{++}/Clipper file-based database applications.

Q163525 Delay When Saving Word File to Windows NT 4.0 Server FYI: Office 2000 has problems with oplocks too! You are not alone!

What to do?

The simplest way to resolve problems related to oplocks is to disable them on behalf of the LAN Manager server.

This is done by manipulating the following registry key:

```
\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters
EnableOplocks   REG_DWORD   0 or 1
Default: 1 (true)
```

Specifies whether the server allows clients to use oplocks on files. Oplocks are a significant performance enhancement, but have the potential to cause lost cached data on some networks, particularly wide-area networks.

By default, the registry entry is 1 (oplock enabled), as a rule of thumb you should set this key to 0 (disable oplock) when sharing files with Xbase^{++}/Clipper/FoxPro and MS-Access. We have not encountered any performance drawbacks in real-world scenarios after having disabled oplocks. However problems with Xbase^{++} file-based database applications simple went away after re-configuration of the server.

What else todo?

Also the LAN Manager server has a build in strategy to optimize file open/close patterns, this feature is intended and designed to optimize batch processing where files get opened/closed often. The drawback with this strategy is that sometime your table or index files is closed on behalf the client but not on the server side. Which in turn may cause a conflict when opening the file again. This configuration setting is called the CachedOpenLimit. To be on the safe side you should also disable this feature using the following registry setting:

```
\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters
CachedOpenLimit  REG_DWORD   0 to disable or > 0
Default: 5 (open files per connection)
```

Summary

Opportunistic locks where introduced by Microsoft to increase performance of network applications when it comes to file sharing. The strategy used significantly decreases performance of file-based database applications. The oplock can easily and safely be disabled via a registry-key setting change. After disabling of oplocks your database application should show increased performance and higher reliability. Other applications typically do not encounter performance penalties. The only potential drawback of disabling oplocks is eventually a higher bandwidth usage when it comes to standard file usage patterns such as opening/closing/reading/writing office documents.