Transaction Support in Windows NTFS

Surendra Verma
Development Manager
Windows File Systems
Microsoft

Transactional NTFS (TxF)

- Adds transaction support for all NTFS file operations:
  - Full Atomicity, Consistency, Isolation, Durability
  - Allows arbitrary number of file system operations to be treated as an atomic unit
  - Reads, writes, file creations, deletions, renames, security, object-id, named-streams, quota etc.

Semantics - isolation

Committed Read without blocking Reader for Writers

- Transaction 1:
  - File 1
  - File 2 <- New File
  - File 3 -< Deleted File
- Transaction 2:
  - File 1
  - File 3

Transactions don't see changes made by other transactions

Same for Non-Transactions

Contemplating Dirty Reader as an isolation level

Semantics - locking

- File is the unit of locking
- File locked for update for the duration of the transaction
- Other handles in the same transaction allowed to update
- Can be read concurrently (& consistently) by the others

Demo

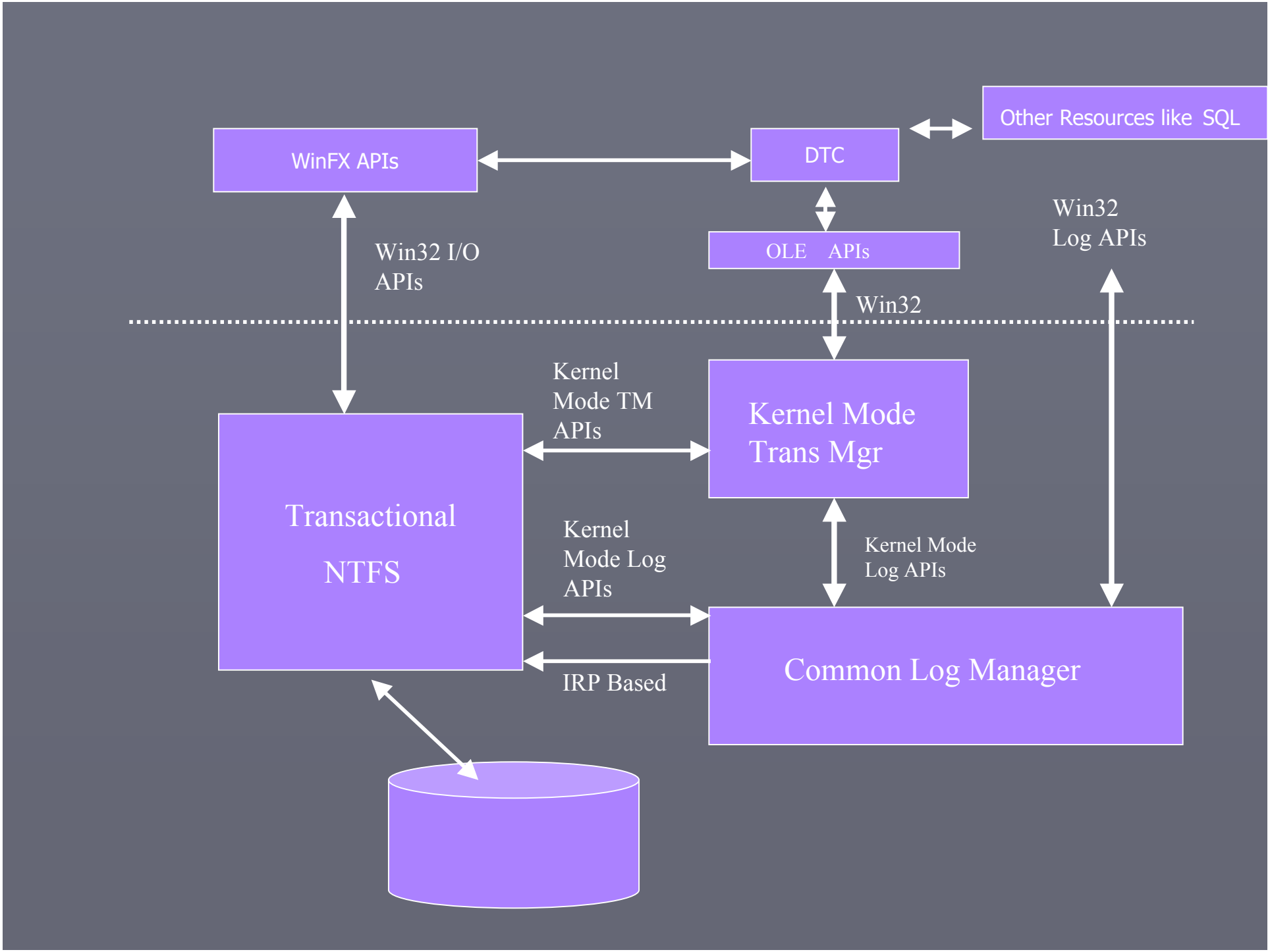
Volumes and RMs

- Each volume comes with a Transactional *Resource Manager* (RM) by default
- Its log is resident on the volume. Recovery automatic. Admin-free.
- Many *secondary* RMs may be created in various places within the volume
- Their logs can be anywhere on the machine
- Their admin/recovery is user-controlled via APIs
- Designed to be embedded in other stores/applications

Logging Modes

- Undo-Only logging: Minimizes logging and supports ACID semantics
- Redo-Undo logging: "redo" is logged as well => log contains *complete* history of changes
- Allows playback from a backup to achieve consistency at a chosen point in time
- Logging Mode can be set for secondary RMs and toggled live

Implementation Details


Other Resources like SQL
WinFX APIs
DTC
Win32 I/O APIs
OLE APIs
Win32 Log APIs
Win32
Kernel Mode TM APIs
Kernel Mode Trans Mgr
Transactional NTFS
Kernel Mode Log APIs
Kernel Mode Log APIs
Common Log Manager
IRP Based

TxF Recovery

TxF builds Two types of content treated differently.

- **Metadata** – Names, attributes, security etc.

upon Ntfs recovery.

- **Data** –

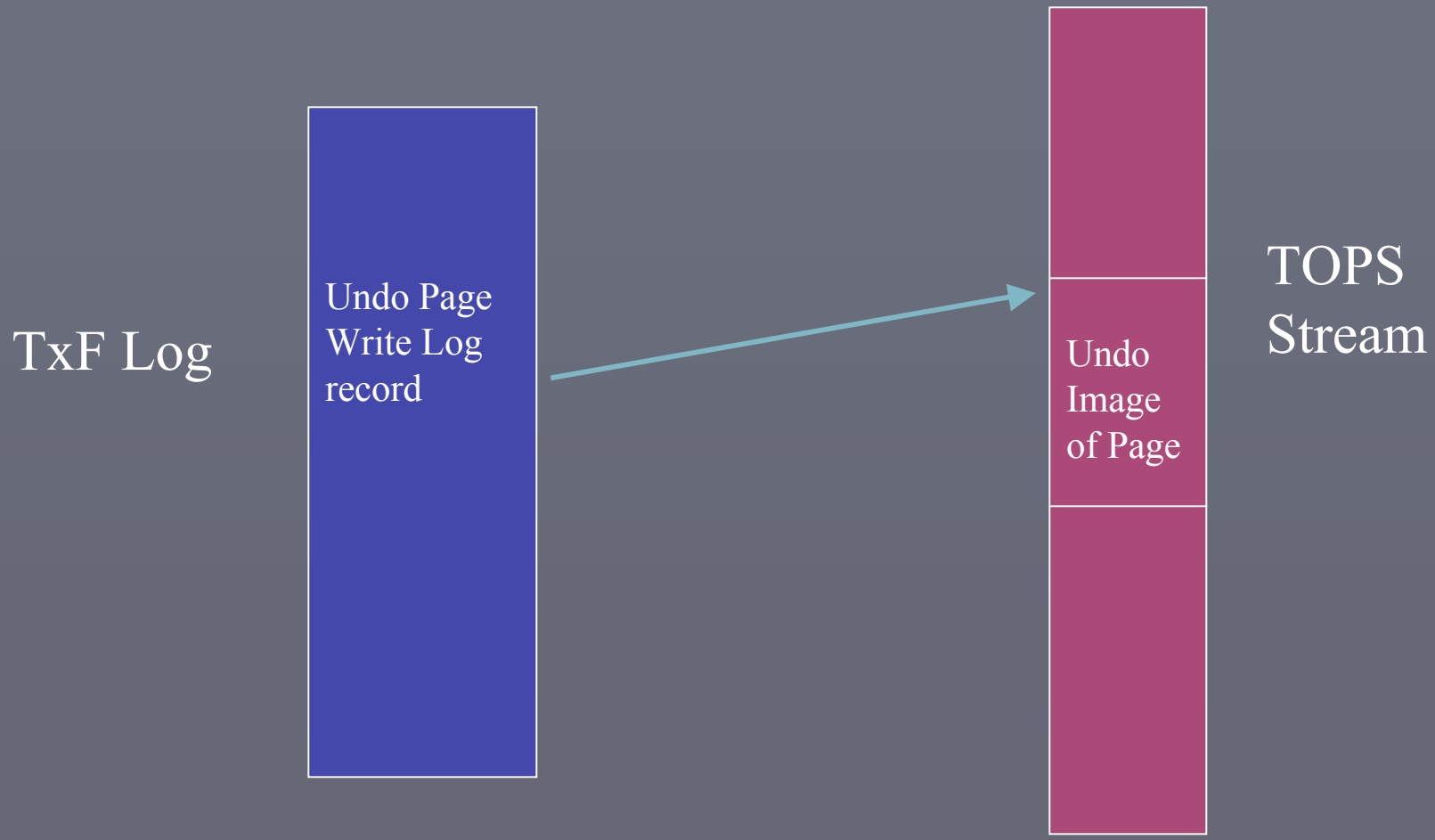
Built from scratch.

- Both use Write Ahead Logging Technique.

Namespace Isolation

- Main-memory balanced binary trees used.

TxF Data Recovery



TxF Data Recovery

- For undo only logging mode – files flushed on commit.
- TOPS stream pages and log written independently of each other.
- A page changed multiple times in a transaction gets logged only once.

Questions

Transaction Management (KTM)

- Coordinates commit/abort processing between the various actors:
  - Resource Managers (RMs) (eg, SQL, TxF)
  - Applications
- Persistently maintains outcome of transactions using the common-log
- Lives in the kernel with Win32 and kernel interfaces

Why Common Logging

- Group log writes from multiple clients into a single physical Disk I/O
- Single logical view log for tightly coupled but distinct resources
- Ease of configuration, archival, and media recovery, and administration
- Single paradigm for high-bandwidth logging on Windows

Common Logging

- Multiple clients sharing a single logical log stream
- Each client has exclusive use of a virtual log stream
- Common log manager multiplexes multiple client streams to single logical log stream
- Multiplexing separated from I/O

Example: two file updates

- Program writes to file1, then to file2
- System/application crashes
- Are the updates on disk?
- Both on disk? Some? None?
- Do others see updates as they occur?
- What if the system showed the previous (consistent) state until app ready to expose them?
- Same issues if only *one* file is involved!

Scenario: Update a web-site

- Hide temporary inconsistencies
- System handles data recovery on app failure or system crash
- System guarantees that updates survive crash once committed
- On high-end, archive transaction logs for shipping or media recovery

Scenario: Remote file Copy

- Reliable copy of file over the network
- Cheap, low-level message transfer coordinated with other transaction work.
- Pass data between branch office and central office (financial institutions, retail)
- Frequently mentioned scenario by our customers

Document Management

- Files in the file-system, file-attributes in a relational database
- Transaction maintains consistency between the two
- Makes it possible to integrate administration utilities between the two stores