

Mark Broadbent

 #SqlSat675



# Persistence Is Futile- Implementing Delayed Durability in SQL Server

# Sponsor



Bi FACTORY



#SqlSat675 – 18/11/2017



# Organizzatori



GetLatestVersion  
.it

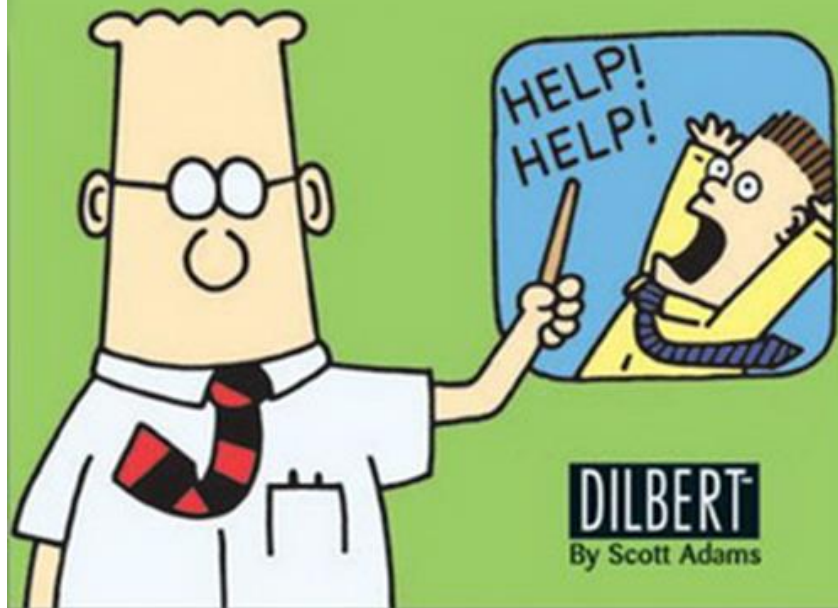


#SqlSat675 – 18/11/2017



# Agenda

## Our Transactional Processing Plan Goes Something Like This...



Transactions & Breaking ACID Properties

1



Log Architecture

2



Delayed Durability Implementation

3



Limitations

4



The best part of the presentation... Gin O'Clock

5

We will also need to discuss and explain In-Memory OLTP in this context!

A  
C  
I  
D



# Transactions can be...

## Auto-Commit

```
INSERT ...  
DELETE ...
```

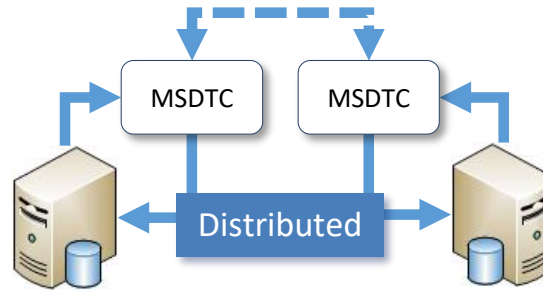
## Explicit

```
BEGIN TRAN  
...  
COMMIT TRAN
```

```
INSERT ...  
DELETE ...  
COMMIT TRAN
```

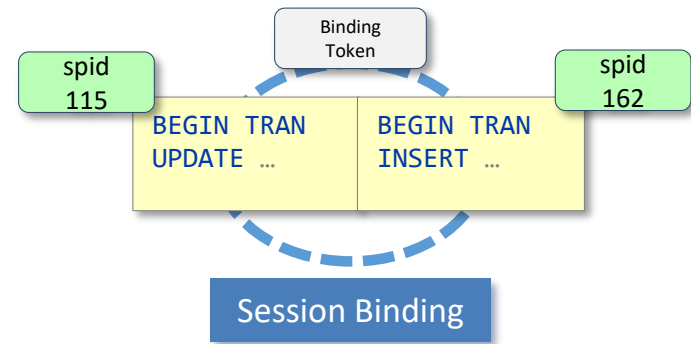
Implicit (evil muthas!)

```
BEGIN DISTRIBUTED TRAN  
...  
COMMIT TRAN
```

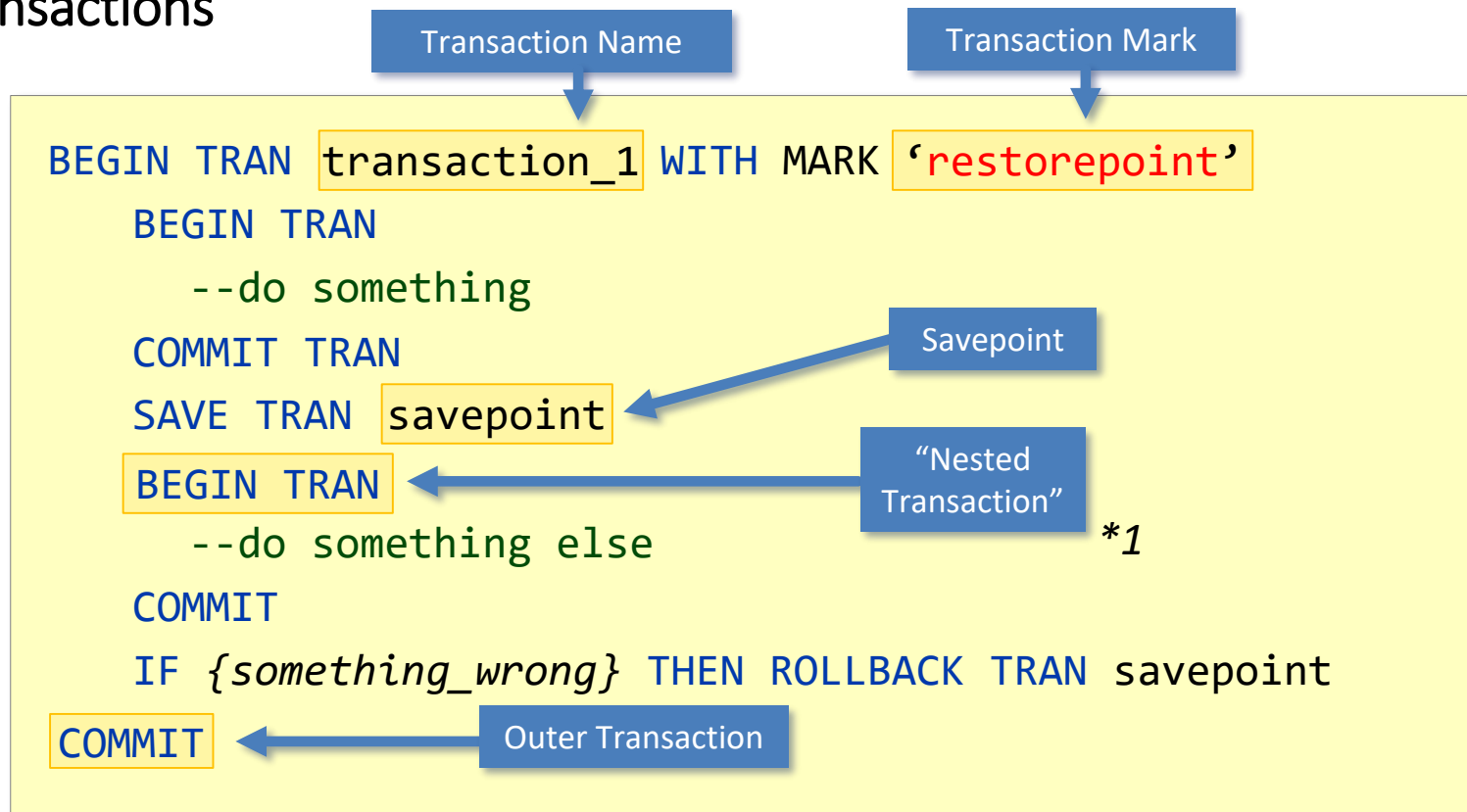


## Batch Scope (MARS)

```
UPDATE ...  
INSERT ...  
DELETE ...  
GO
```



# Transactions



*\*1 Not to be confused with Atomic Blocks (we'll discuss these later!)*

# Demo

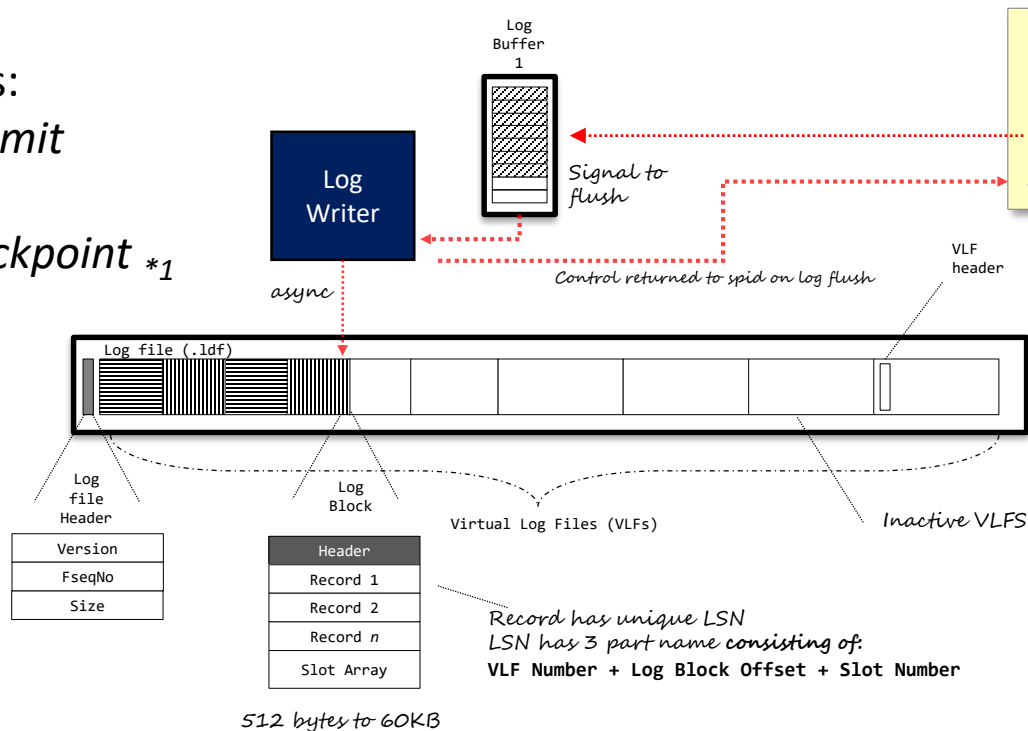
Transactions and broken ACID properties...



# Durable Transactions

Log flushes:

- *On commit*
- *On full*
- *On Checkpoint* \*1



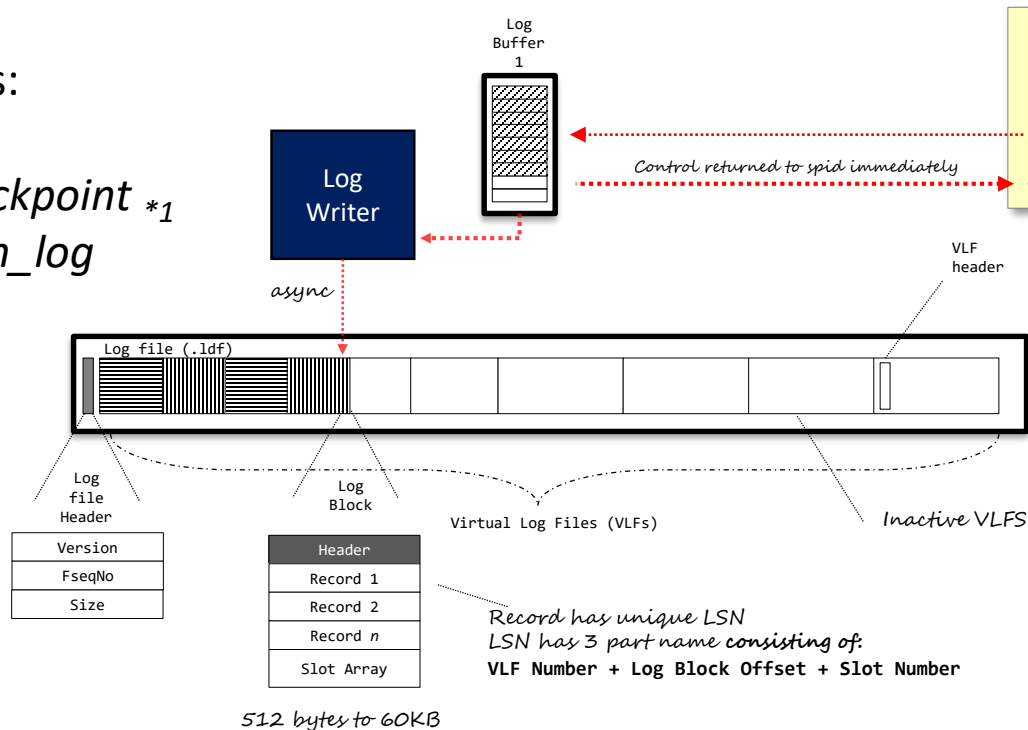
Log growth size  
≤ 64MB then 4 VLFs  
> 64MB AND ≤ 1GB 8 VLFs  
> 1GB 16 VLFs  
Think about how big your log needs  
to be from the beginning

\*1 Data pages that need to be written to disk will require  
Buffers containing earlier LSNs to be flushed to disk

# Delayed Durable Transactions

Log flushes:

- *On full*
- *On Checkpoint* \*<sub>1</sub>
- *sp\_flush\_log*

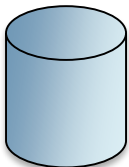


Log growth size  
≤ 64MB then 4 VLFs  
> 64MB AND ≤ 1GB 8 VLFs  
> 1GB 16 VLFs  
Think about how big your log needs to be from the beginning

\*1 Data pages that need to be written to disk will require Buffers containing earlier LSNs to be flushed to disk

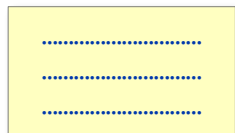
# Delayed Durability Hierachy

## DATABASE



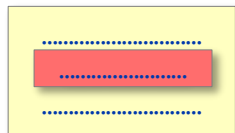
- DISABLED
- ALLOWED
- FORCED

## TRANSACTION (On-disk or IM tables)



- OFF
- ON

## ATOMIC BLOCK (IM tables via Native Compiled Stored Procs)



- OFF
- ON

## Except

- If CDC is enabled
- Is a cross database transaction
- Is a distributed transaction

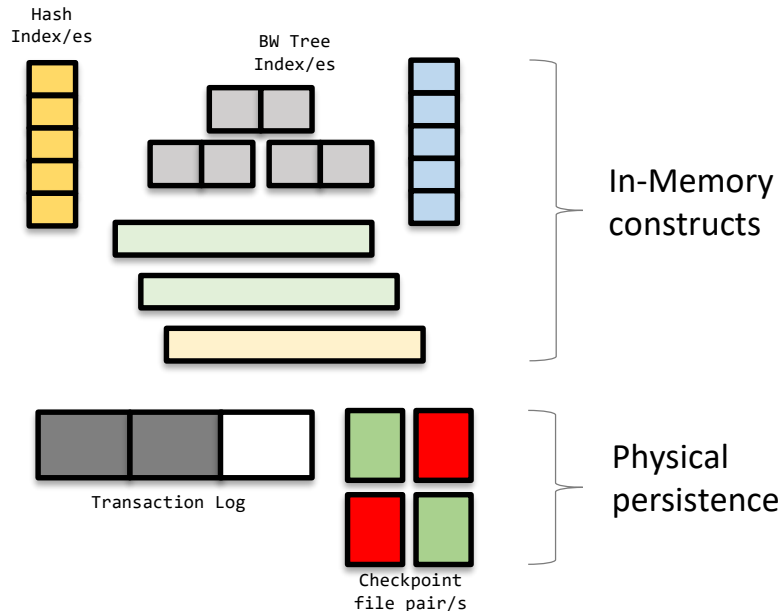
...but delayed durability not  
guaranteed regardless!

# Demo

On disk delayed durability

# In-Memory OLTP Overview

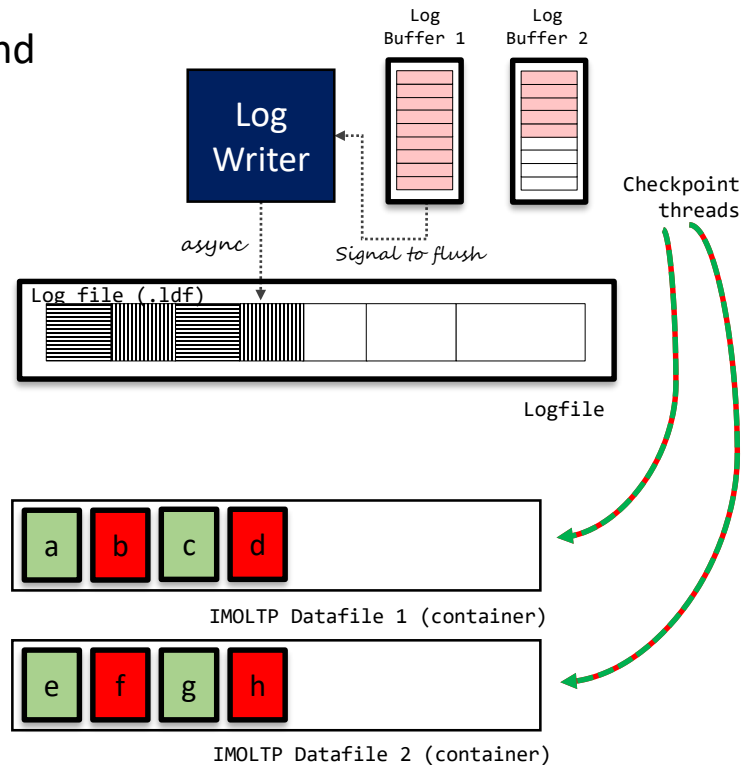
- New concurrency model (as of SQL Server 2014)
- Provides lockless and latchless operation
- All row and index data in memory (optimised for memory data structures)
- All versioning In-Memory
- Physical persistence (SCHEMA\_AND\_DATA)



# In-Memory OLTP Logging and Improvements

- Persistence (and recovery) through Transaction Log and checkpoint file pair/s
  - ...but log file is the primary source!
- Log Improvements
  - Log block “compression”
  - No undo record generation
  - Indexes not persisted, rebuilt on start-up
    - ...so NO index maintenance logging

...but on disk structures become an even bigger potential bottleneck



# Demo

In-Memory delayed durability

# Log Waits and Log Performance Monitoring

- LOGBUFFER wait – time taken to create log record in log buffer
- WRITELOG wait – time taken for log buffer to be flushed to logfile
- Use sys.dm\_io\_virtual\_file\_stats to look at file IO, sizes and stalls
- Performance Monitors e.g.
  - Log flushes per second counter of Databases counter set
  - Transactions per second counter of Databases counter set



# Limitations and Special Cases

- Ignored when
  - CDC enabled tables
  - Cross-database or distributed transactions
- Not supported in Transaction Replication
- Data is delayed (temporarily missing) in
  - Sync Read-only replicas
  - High Safety Database Mirrors
  - Log backups and Log shipping targets

**Warning!**  
Data loss is very possible in these scenarios!

*\*1 Basically crash recovery in standalone and automatic failovers in HADR configurations can and possibly will result in data loss!*

# Summary

- Delayed Durability attempts to make log flushes more efficient
- A compromise between Improving performance over durability
- Can force durability on every database transaction, specific transactions or atomic block via Natively Compiled Stored Procedures.
- In-Memory OLTP could remove all other bottlenecks – so Delayed Durability \*could\* remove log buffer waits
  - However IMOLTP already provides some great logging improvements
- You can lose data, even under highly available environments!

# Thank you for listening!

Email: [mark.broadbent@sqlcambs.org.uk](mailto:mark.broadbent@sqlcambs.org.uk)

Twitter: [retracement](#)

Blog: <http://tenbulls.co.uk>

Slideshare: <http://www.slideshare.net/retracement>

Demo: [https://github.com/retracement/Persistence is Futile](https://github.com/retracement/Persistence_is_Futile)