



# Mastering TempDB

Danilo Dominici  
ddominici@gmail.com

# Partners



UNIVERSITÀ  
POLITECNICA  
DELLE MARCHE



# Agenda

Che cos'è

Un po' di  
storia...

Monitoring &  
troubleshooting



Che cos'è il TempDB



# Che cos'è il TempDB

- E' un database 😊
- Ne esiste uno solo per ciascuna istanza SQL Server
- La struttura è identica a qualsiasi altro database utente
- Viene ricreato ogni volta che SQL Server viene riavviato
  - Partendo dal database *model*
- Usa il recovery model SIMPLE
  - Unica proprietà che non copia dal database model 😊
- Utilizza un unico filegroup: PRIMARY
- Drop, detach, attach non sono possibili

# Che cosa contiene?

- Tabelle temporanee(# or ##)
- Variabili table
- Risultati nelle funzioni table-valued
- Triggers
- Sorts
- Hash worktables
- Operazioni *online* sugli indici
- Table / Index Spools
- Cursori

TIP: non  
è il  
Tardis 😊





# Come nasce il TempDB ?

Subito dopo aver aperto il database master, viene creato il TempDB

Per prima cosa viene letto il database model

...e copiato (dati e proprietà) nel TempDB

Una volta pronto, viene notificato che il TempDB è pronto e avviata l'apertura dei database utente

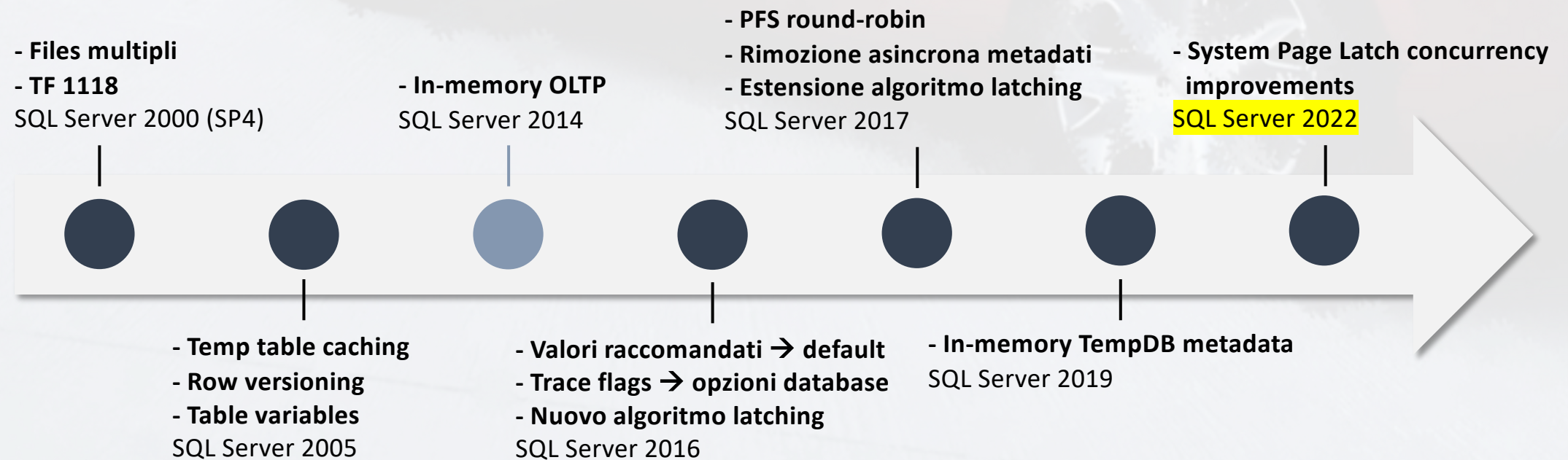
Se per qualsiasi ragione il TempDB non può essere creato, SQL Server viene spento



Un po' di storia...



# Che cosa è successo negli anni al TempDB?



# SQL Server 2000 Service Pack 4

## **Problema**

Object/Metadata Allocation Contention

## **Soluzione**

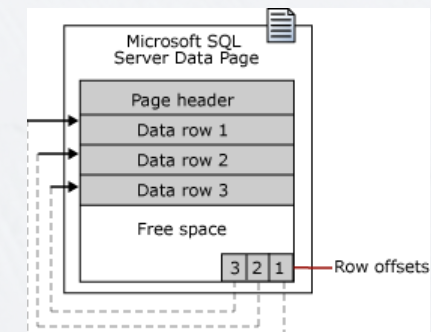
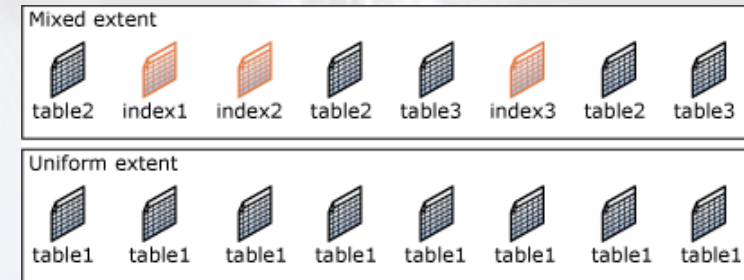
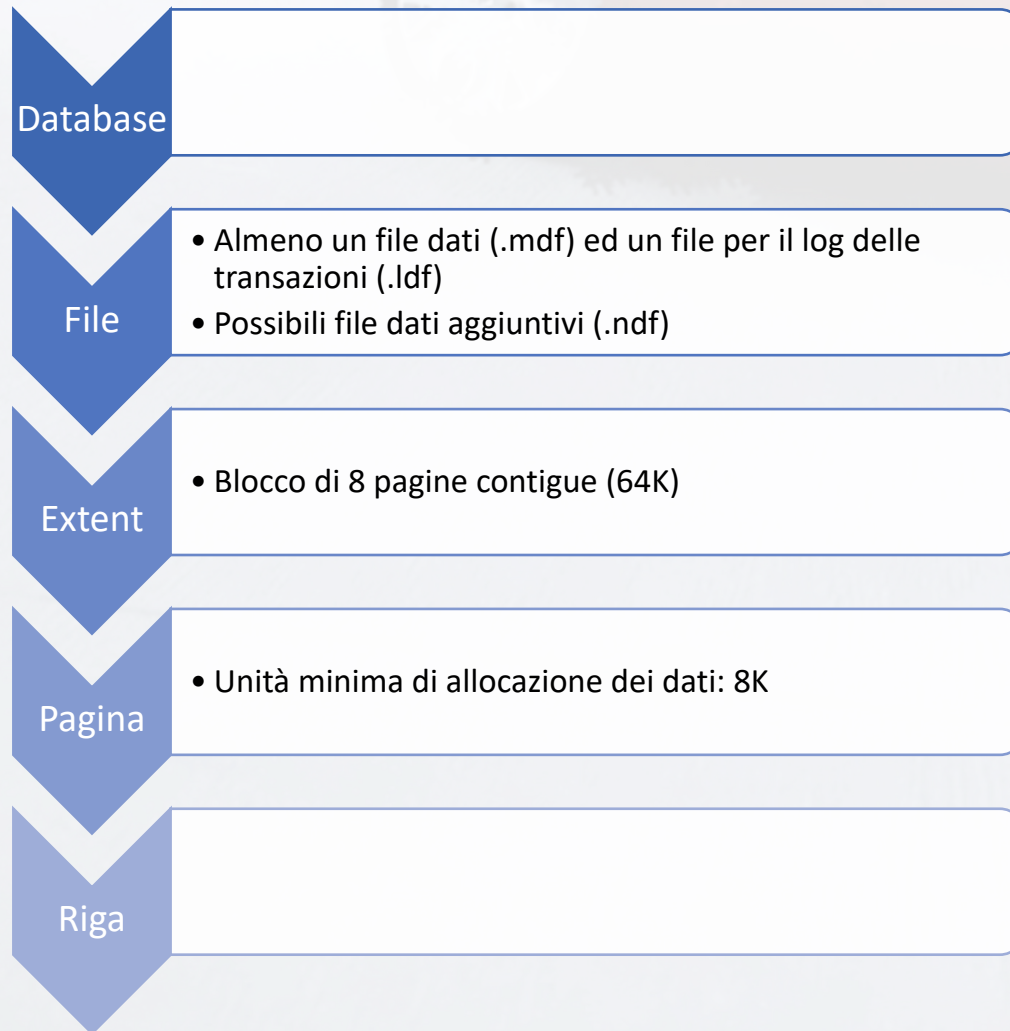
Utilizzo di files multipli, della stessa dimensione

Trace flag 1118

Trace flag 1117 (opzionale)



# Ripasso: strutture interne

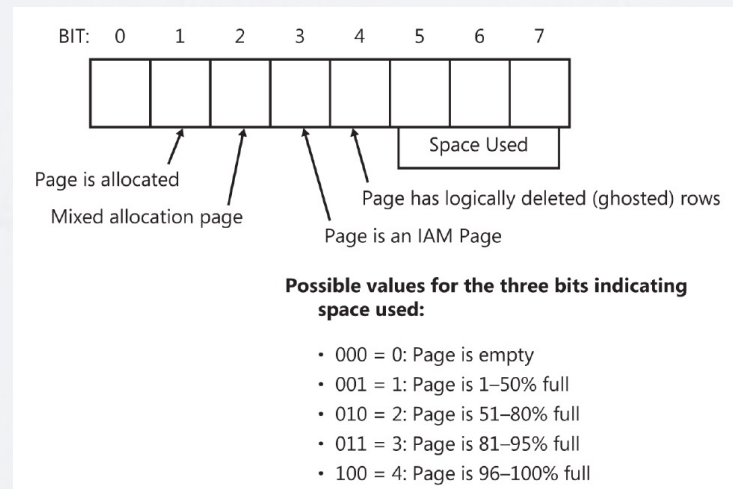


# Pagine di allocazione

## PFS (Page Free Space)

- Indica cosa è memorizzato in ciascuna pagina
- Una ogni 8088 pagine
- Ogni PFS contiene circa 8000 byte

TIP: l'indirizzo di una pagina ha il formato  
DBID:FileID:pagina





# Pagine di allocazione <sup>(2)</sup>

## GAM (Global Allocation Map)

- Registra se un extent è allocato o no
- Una ogni 511230 pagine
- Ogni GAM contiene circa 64000 bits, quindi copre circa 4GB di dati

## SGAM (Shared Global Allocation Map)

- Registra quali extent sono utilizzati come misti
- Una ogni 511230 pagine
- Ogni SGAM contiene circa 64000 bits, quindi copre circa 4GB di dati

Utilizzo dell'extent	GAM bit	SGAM bit
Libero, non in uso	1	0
Uniforme o misto (ma pieno)	0	0
Misto con pagine vuote	0	1

# Creare una nuova tabella temporanea significa:

Leggere la pagina SGAM (2:1:3) per cercare un extent misto con spazio libero (PAGELATCH\_EX)

Leggere la pagina PFS (2:1:1) per cercare una pagina libera all'interno dell'extent (PAGELATCH\_EX)

Creare la tabella temporanea e gli eventuali dati al suo interno

Aggiornare le pagine di allocazione in relazione allo spazio residuo (es. se l'extent in cui ho creato la tabella temporanea è pieno o ha ancora spazio)



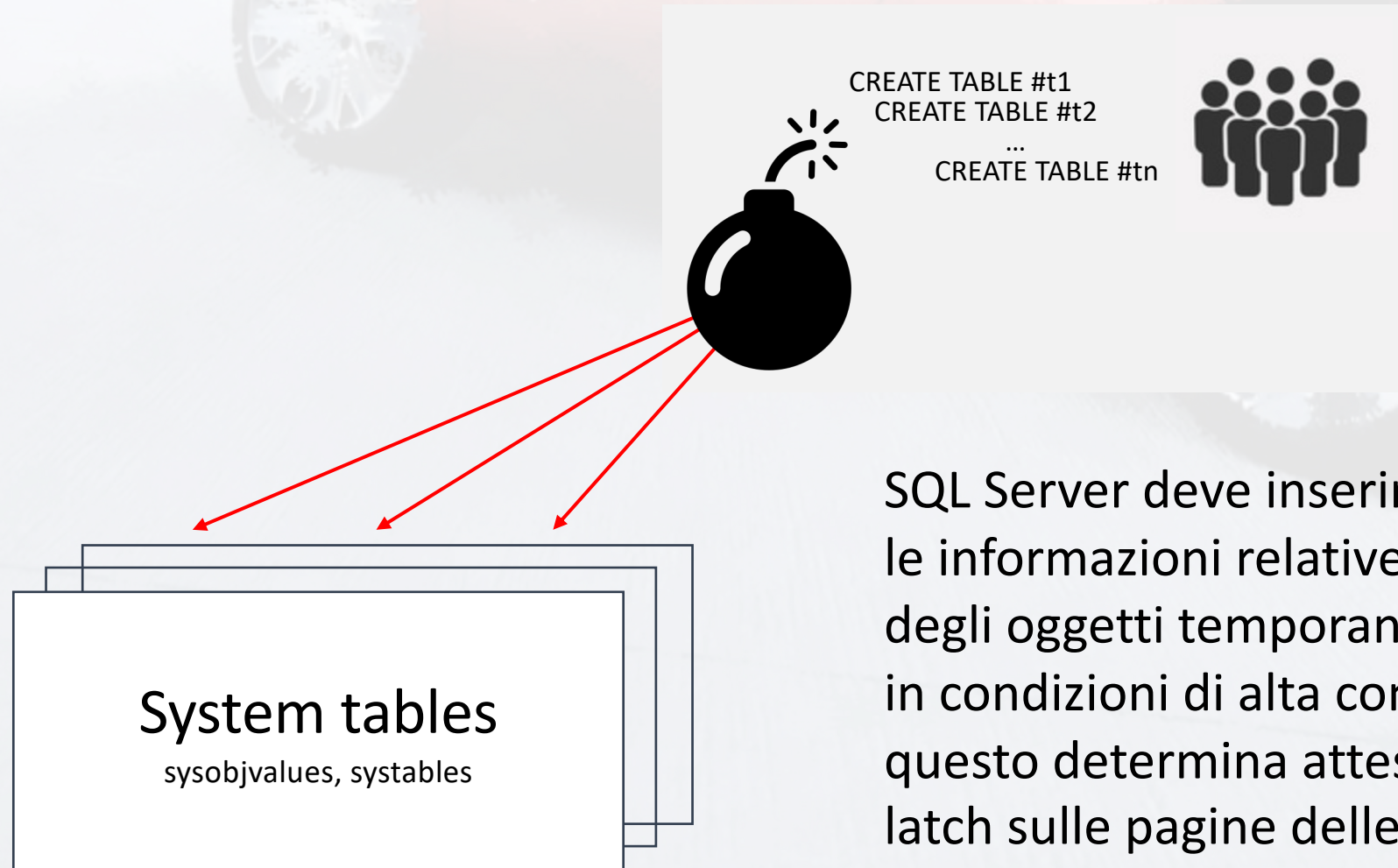
# Object Allocation Contention

SQL Server deve leggere ed aggiornare le pagine di allocazione del database e in condizioni di alta concorrenza si manifesta come tempi di attesa (PAGELATCH\_XX)

tempdb.mdf



# Metadata Allocation Contention



SQL Server deve inserire/modificare le informazioni relative ai metadati degli oggetti temporanei creati e in condizioni di alta concorrenza questo determina attese dovute ai latch sulle pagine delle tabelle di sistema (PAGELATCH\_XX)



# Soluzione: utilizzo di files multipli

La regola è: se il numero di processori logici è pari o inferiore a 8, creare un file per processore logico

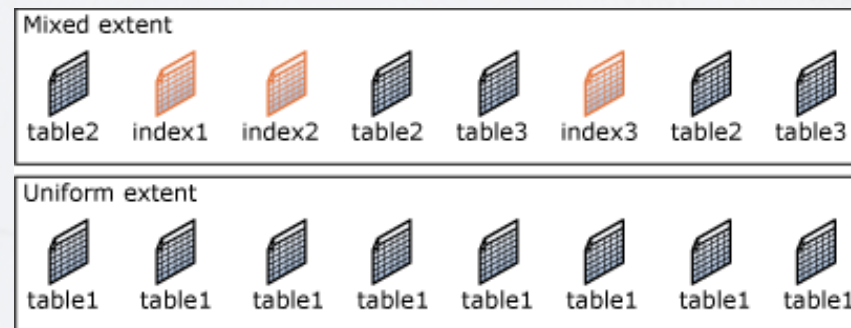
Se il numero di processori logici è superiore a 8, iniziare con 8 file

Monitorare se c'è contesa di accesso alle pagine di allocazione, misurabile tramite i wait types PAGELATCH\_XX sulle pagine PFS, GAM e SGAM del TempDB

# Trace flags 1118

## Trace Flag 1118

- Forza l'allocazione di extent uniformi
- Riduce la contesa di accesso alle pagine SGAM (che tracciano gli extent misti)
- Si applica all'intera istanza come parametro di avvio





# Trace flags 1117

## Trace Flag 1117

- Forza la crescita contemporanea di tutti i files dello stesso filegroup (anche PRIMARY)
- Serve ad assicurarsi che l'algoritmo di bilanciamento dell'allocazione degli oggetti funzioni correttamente
- Si applica all'intera istanza come parametro di avvio
  - Pro e contro: se ho un database utente suddiviso su più files, anche su filegroup diverso da PRIMARY, tutti i files crescono allo stesso modo



# Demo

Object Allocation Contention



# SQL Server 2005

## **Problema**

Metadata Allocation Contention

## **Soluzione**

Caching delle tabelle temporanee

Aggiunti nuovi tipi di oggetti tra quelli memorizzati in precedenza nel TempDB

- Row versioning
- Table variables

# Caching delle tabelle temporanee

Si può sfruttare al meglio quando:

- Non vengono creati constraint *nominali*
- Non si usano statement DDL come CREATE INDEX o CREATE STATISTICS
- Gli oggetti non vengono creati dinamicamente (es. via sp\_executesql)
- Gli oggetti vengono creati all'interno di oggetti quali stored procedure, trigger o funzioni



# SQL Server 2016

Le impostazioni *suggerite* per il TempDB sono ora il default in fase di installazione

# Le impostazioni suggerite sono ora di default

- Durante il setup viene impostato il numero di files per il TempDB sulla base del numero di CPU logiche rilevate
- La dimensione iniziale viene impostata ad 8MB con autogrowth di 64MB e max size di 1GB

The screenshot shows the 'Database Engine Configuration' window with the 'TempDB' tab selected. The 'TempDB Data Files' section is highlighted with a red box, showing the following settings: Number of files: 4, Initial Size (MB): 8, Autogrowth (MB): 64. The 'Data Directories' section shows the path 'C:\Program Files\Microsoft SQL Server\MSSQL13.SQLSERVER2016\TempDB' with an 'Add...' button. The 'TempDB Log File' section shows the path 'C:\Program Files\Microsoft SQL Server\MSSQL13.SQLSERVER2016\TempDB.ldf' with an 'Add...' button. The 'TempDB Log File' section also shows the initial size (8 MB) and autogrowth (64 MB) settings.

**Database Engine Configuration**  
Specify Database Engine authentication security mode, administrators, data directories and TempDB settings.

Product Key  
License Terms  
Global Rules  
Product Updates  
Install Setup Files  
Install Rules  
Setup Role  
Feature Selection  
Feature Rules  
Instance Configuration  
Server Configuration  
**Database Engine Configuration**  
Feature Configuration Rules  
Ready to Install  
Installation Progress  
Complete

Server Configuration Data Directories **TempDB** FILESTREAM

TempDB Data Files

Number of files: 4  
Initial Size (MB): 8 Total Initial Size (MB): 32  
Autogrowth (MB): 64 Total Autogrowth (MB): 256

Data Directories: C:\Program Files\Microsoft SQL Server\MSSQL13.SQLSERVER2016\TempDB Add... Remove

TempDB Log File

Initial Size (MB): 8  
Autogrowth (MB): 64

Log directory: C:\Program Files\Microsoft SQL Server\MSSQL13.SQLSERVER2016\TempDB.ldf ...



# Modifica nella gestione dei trace flags 1117/8

I trace flag 1117 e 1118 sono ora attivi per default per il TempDB e possono essere modificati sui singoli database mediante le opzioni: MIXED\_PAGE\_ALLOCATION e AUTOGROW\_\*

```
ALTER DATABASE <dbname>
```

```
SET MIXED_PAGE_ALLOCATION { ON | OFF }
```

```
ALTER DATABASE <dbname>
```

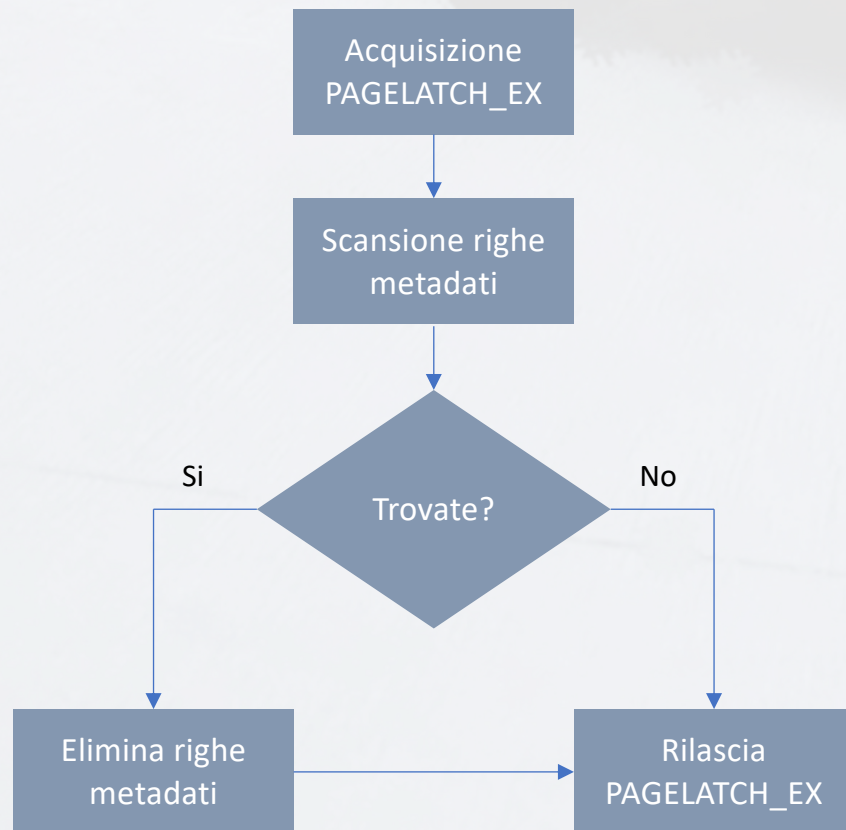
```
MODIFY FILEGROUP <filegroup> {
```

```
    AUTOGROW_ALL_FILES | AUTOGROW_SINGLE_FILE
```

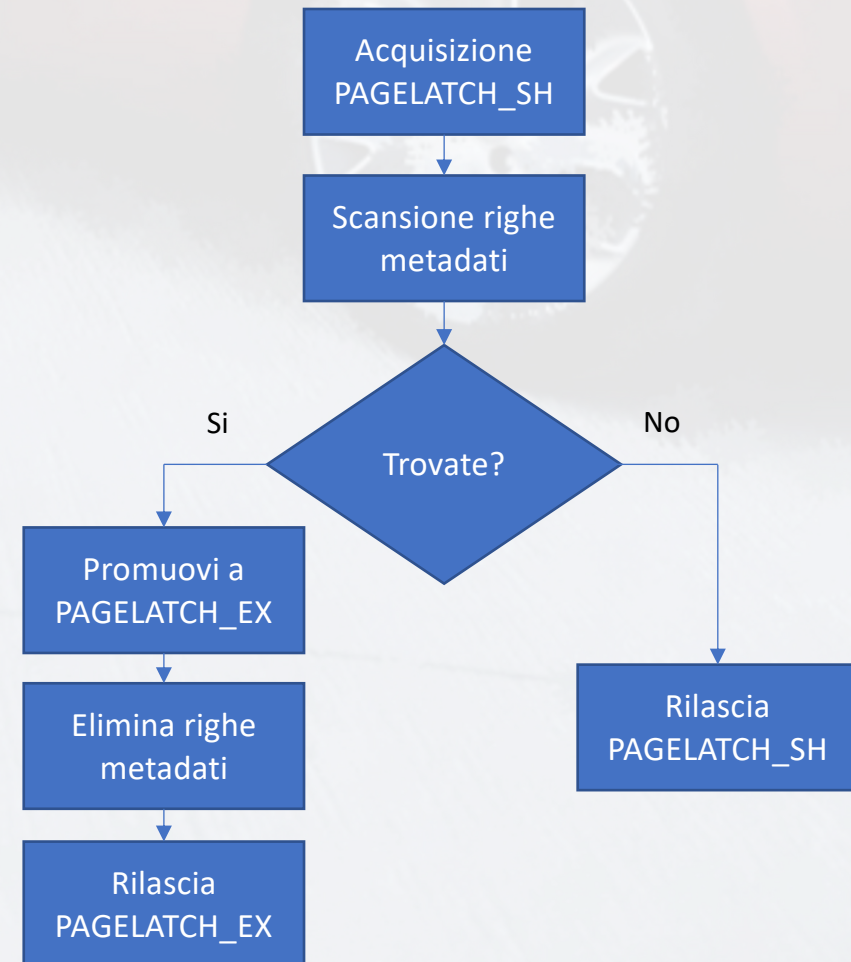
```
}
```

# Ottimizzazione dell'algoritmo di latching

## Vecchio algoritmo



## Nuovo algoritmo





# SQL Server 2017

- Il setup di SQL Server può specificare una dimensione iniziale per i files del TempDB di 256GB, con un warning per dimensioni superiori a 1GB quando IFI è disabilitato
- Migliorato l'algoritmo round-robin di allocazione degli oggetti nelle pagine PFS
- Estesa l'applicazione dell'algoritmo di latching
- Eliminazione asincrona dei metadati

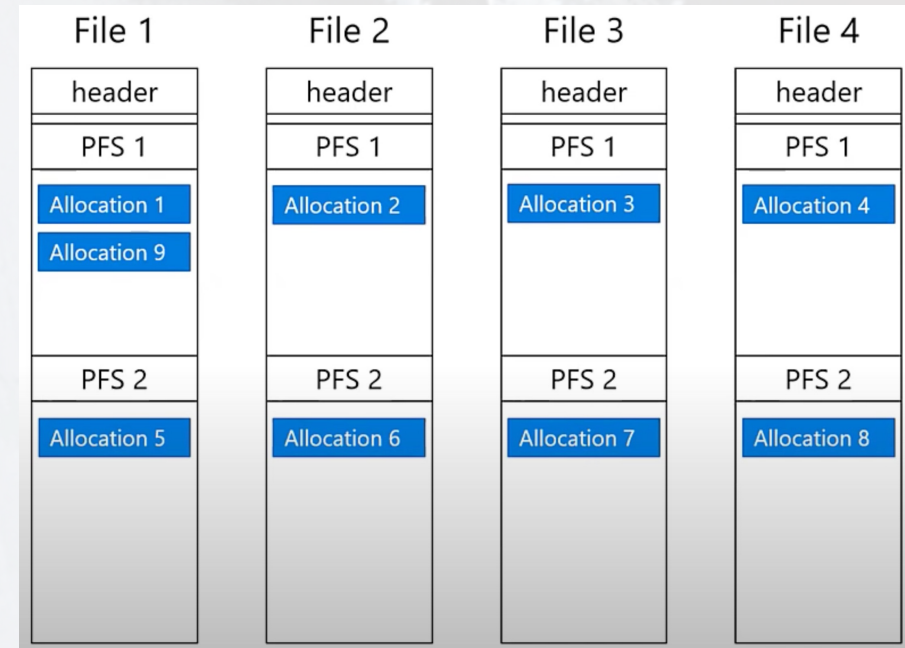
# Ottimizzato l'algoritmo di allocazione in PFS

Nella vecchia versione l'algoritmo (round-robin) sceglieva per ogni nuova allocazione un file diverso

Con questa ottimizzazione le nuove allocazioni vengono fatte su tutte le pagine PFS presenti

Implementato in:

- SQL Server 2017 CU7
- SQL Server 2017 SP1 CU9
- SQL Server 2016 SP2 CU1
- SQL Server 2014 SP3





# SQL Server 2019

## **Problema**

Metadata Allocation Contention

## **Soluzione**

Memory-optimized TempDB metadata

Miglioramento nella gestione della cache

Aggiornamento concorrente delle pagine PFS



# Demo

Memory-optimized TempDB metadata



# SQL Server 2022

## **Problema**

Metadata Allocation Contention

## **Soluzione**

Aggiornamento concorrente delle pagine  
GAM/SGAM



# Monitoring & troubleshooting



# Monitoring & troubleshooting

Gli strumenti inclusi in SQL Server per monitorare il TempDB sono:

- Dynamic Management Views (DMV)
- Extended Events
- SQL Profiler / SQL Trace
  - Deprecati, meglio usare gli Extended Events
- Performance Monitor di Windows

# Monitoring & troubleshooting <sup>(2)</sup>

## Performance Counters

- Access Methods: Worktables Created/sec
  - Indica query spools, LOB variables e cursori
- Access Methods: Workfiles Created/sec
  - Indica il numero di operazioni di hashing
  - Memorizza i risultati temporanei nelle operazioni di join/aggregati
- General Statistics: Temp Tables Creation Rate
  - Indica il numero di tabelle temporanee create
- General Statistics: Temp Tables Destruction Rate
  - Indica il numero di tabelle temporanee in attesa di essere eliminate dal thread di pulizia



# Monitoring & troubleshooting (3 )

## DMV

- `sys.dm_io_virtual_file_stats`
  - Problemi di performance
- `sys.dm_db_file_space_usage`
  - Consente di identificare il numero di pagine usate per tipologia (unallocated, version\_store, user e internal)
- `sys.dm_db_session_space_usage`
  - Utilizzo del TempDB per le sessioni attive
- `sys.dm_db_task_space_usage`
  - Utilizzo del TempDB per i task in esecuzione

# Ricapitolando: che cosa dobbiamo fare?

## Configurazione

- Files multipli di dimensioni identiche
- Il minore tra 1 file per core o 8 ed incrementare se necessario
- TF 1117+1118 se SQL Server 2014 o precedenti, altrimenti abbiamo già il default come serve
- In alternativa, pre-dimensionare il TempDB al 80% del disco e monitorare lo spazio
- Disco dedicato, SSD, RAID10
- Abilitare Instant File Initialization

## Versione

- Aggiornare all'ultima Cumulative Update disponibile
- Considerare l'upgrade a SQL Server 2019, dove possibile

## Sviluppo

- Non modificare le tabelle temporanee dopo la creazione
- Non troncare le tabelle temporanee
- Utilizzare la sintassi inline per la creazione degli indici (2014+)
- Evitare dove possibile l'uso di tabelle temporanee in elaborazioni batch