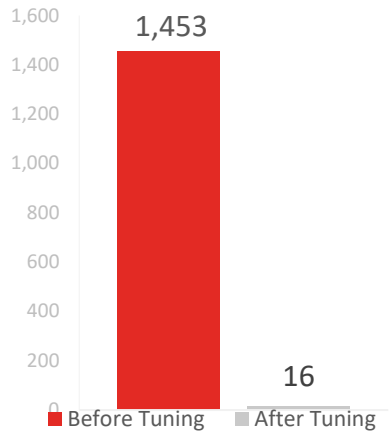
Performance Tuning Report



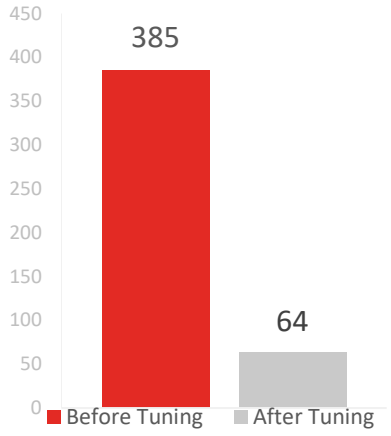
CPU

CPU is **91X** times faster

OR

9,081%

CPU improvement



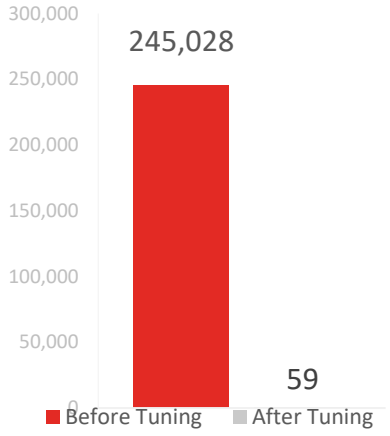
Speed

CPU is **6X** times faster

OR

602%

Speed improvement



Disk (Reads)

Disk is **253X** times faster

OR

415,302%

Disk improvement

Description:

Problem: Select command inside trigger (EHRHist<removed>_AFTER_INSERT_UPDATE) doing an index scan.

Command:

```
SELECT TOP 1 @RecordExists = COUNT(1)
  FROM EHRHist<removed>
 WHERE documentName = @DocumentName
   AND plansOfCare  = @PlansOfCare
   AND encounterID  != @EncounterId;
```

Change: Added an index.

```
CREATE NONCLUSTERED INDEX IX__documentName__encounterID__INC
ON [dbo].[EHRHist<removed>] ([documentName], [encounterID])
INCLUDE ([plansOfCare])
WITH (
       PAD_INDEX = OFF
       , STATISTICS_NORECOMPUTE = OFF
       , SORT_IN_TEMPDB = OFF
       , IGNORE_DUP_KEY = OFF
       , DROP_EXISTING = OFF
       , ONLINE = OFF
       , ALLOW_ROW_LOCKS = ON
       , ALLOW_PAGE_LOCKS = OFF
       , FILLFACTOR = 100
       , DATA_COMPRESSION = PAGE
       )
       ON [INDEXES]
```

Before tuning:

TextData	CPU	Reads	Duration	Writes
Select top 1 count(1) from EHRHist...	1453	245028	426	0
Select top 1 count(1) from EHRHist...	1439	245028	380	0
Select top 1 count(1) from EHRHist...	2296	244995	602	0
Select top 1 count(1) from EHRHist...	1438	244993	385	0

After tuning:

TextData	CPU	Reads	Duration	Writes
Select top 1 count(1) from EHRHist...	0	59	23	0
Select top 1 count(1) from EHRHist...	0	17	62	0
Select top 1 count(1) from EHRHist...	15	17	42	0
Select top 1 count(1) from EHRHist...	16	17	64	0

Technical Background:

Most SQL Servers bottleneck on Disk access (or disk “reads”).

It’s not CPU or RAM – which most customers often suspect first.

And that makes a lot of sense. Here is why.

Inefficient queries scan (or read) lot of data. Data read in is stored in RAM. As more data is read in, “older” data is pushed out from RAM. If there isn’t enough RAM to keep ALL data in memory (which is often not possible), SQL Server has to read from disk – and that is the slowest operation SQL Server can do.

When query can be tuned to read 10 rows vs 10M – less CPU and RAM automatically are necessary. Therefore, tuning for less disk “reads” is often the primary goal.

To the end user nothing is more important than Speed (or Duration of the query) though.

Tuning to reduce CPU/RAM resources are helpful too.

When queries are tuned to need less CPU & RAM, it means that same server now has more capacity. Which means that same server can process double or triple the load. Which means it extends lifespan of the same server. Which means hardware upgrades can be pushed out further into the future.