high volume data processing techniques

julie koesmarno | @mssqlgirl | mssqlgirl.com | #passwit

LobsterPot Solutions | March 26, 2013

PASSWIT

without driving your dba crazy!

julie koesmarno | @mssqlgirl | mssqlgirl.com

LobsterPot Solutions

PASSWIT

about me (@mssqlgirl)

9+ years experience with SQL Server

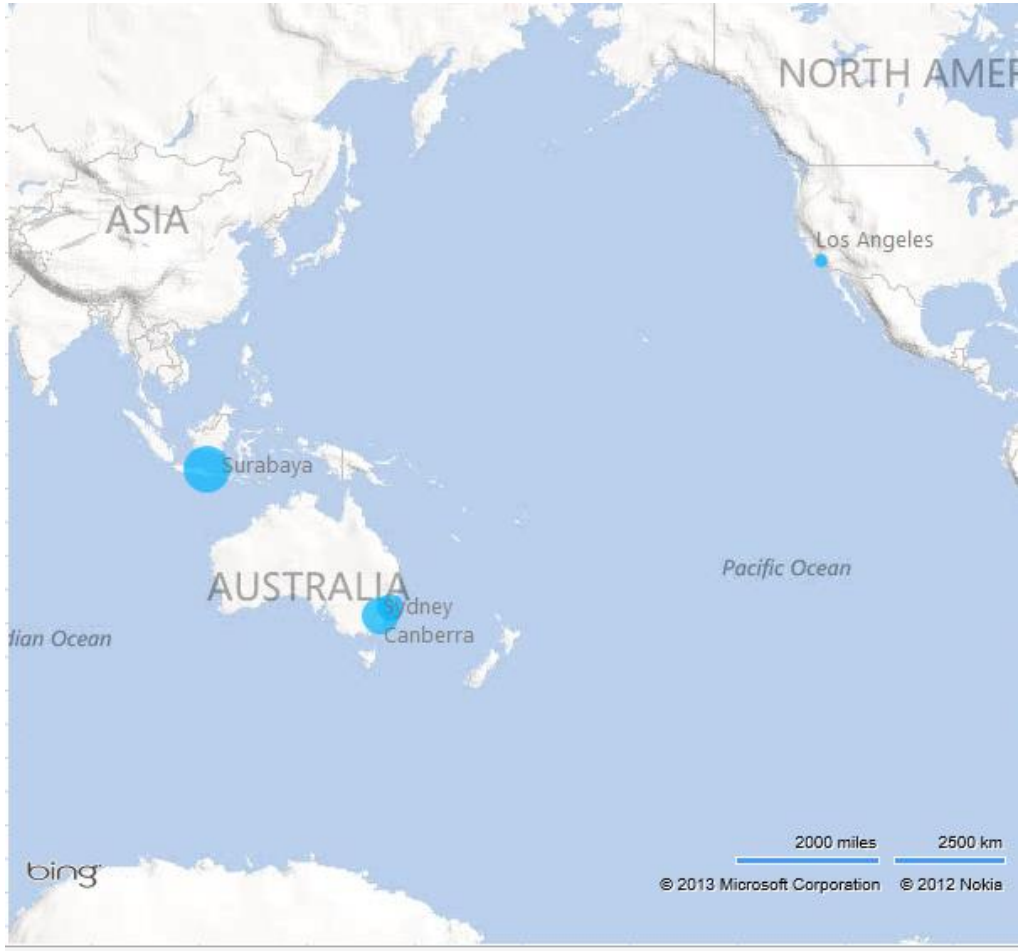
MCSE Data Platform

MCSE Business Intelligence

Current: DW at D&B Credibility Corp in Malibu

Past: BI in SQL Server 2012 using Tabular Model in Sydney

Chapter Leader of SQL MALIBU



agenda

what & why?

batching technique in t-sql

ssis

balanced data distributor

other techniques

caveats

- Every environment is different – requires different configuration
- Test in DEV / UAT
- These are options to other known alternatives such as Partitioning, Index Optimization and many others

what & why

defining the problem

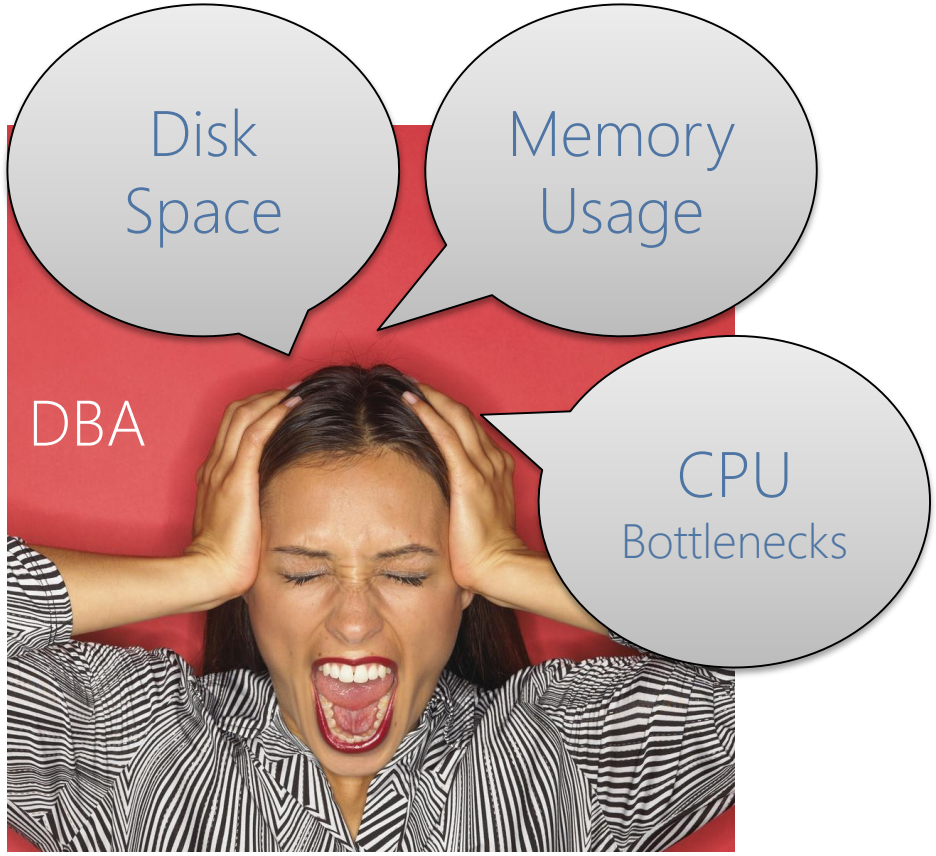
High Volume Data

Size is relative

Business Requirements

- Complex business logic
- Duration
- Data Availability
- Scalability
- Configurable, Rerunable

what drives dbas crazy about it?



why?

why dbas are crazy about it?

Availability / Response Time

Risks Reduction / Avoidance

Governance

details, details, details…

SQL Server environment

- Recovery Model
- Utilisation
- Hardware environment: number of processors, memory, network speed, disk configuration, and many more!

+

What the business wants to achieve

= Great Solution = Happy Users = Happy DBA

useful techniques

The key is to maximise throughput

- Reliable pipe
  - Batch processing using T-SQL
- Fast pipe
  - SSIS, BCP, BULK INSERT
- Divide and conquer
  - Balanced Data Distributor with SSIS
  - ~~Partitioning~~ (not in scope)

t-sql

batching in t-sql

Breaking the problem into smaller pieces

1. Table Lock duration reduced
2. Less rows to rollback
3. Less TempDB consumption
4. Control on error handling
5. Customised visibility on progress

demo

```
21  -- Initialise variables
22  SET @BatchSize = 500000
23  SET @WaitTime = '00:00:00.100' -- delay time after each insert
24
25  -- Get min and max IDs to be processed
26  SELECT
27      @StartID = MIN(i.ISPDailySpeedID),
28      @MaxID = MAX(i.ISPDailySpeedID)
29  FROM Staging.ISPDailySpeed i
30
```

Initialisation

Start of Loop

```
35  -- Run until all IDs are processed
36  WHILE @StartID <= @MaxID
37  BEGIN
38
39      SET @EndID = CASE
40                       WHEN (@StartID + @BatchSize - 1) > @MaxID
41                           THEN @MaxID
42                       ELSE (@StartID + @BatchSize - 1)
43                   END
```

```sql
INSERT INTO Fact.ISPDailySpeed
(
    ISPLocationID,
    CapturedDateID,
    DownloadKbps,
    UploadKbps,
    TotalTests,
    DistanceMiles
)
SELECT
    d.ISPLocationID,
    CAST(CONVERT(VARCHAR(8), i.CapturedDate, 112) AS INT) AS CapturedDateID,
    i.DownloadKbps,
    i.UploadKbps,
    i.TotalTests,
    i.DistanceMiles
FROM Staging.ISPDailySpeed i
    INNER JOIN Dim.ISPLocation d
        ON d.ISPName = i.ISPName
        AND d.City = i.City
        AND d.RegionCode = i.RegionCode
        AND d.CountryCode = i.CountryCode
WHERE i.ISPDailySpeedID BETWEEN @StartID AND @EndID
```

Core Logic

End of Loop

```sql
    WAITFOR DELAY @WaitTime

    SET @StartID = @EndID + 1

END
```

paging function

ORDER BY … OFFSET … FETCH NEXT …

```
51  SELECT
52      d.ISPLocationID,
53      CAST(CONVERT(VARCHAR(8), i.CapturedDate, 112) AS INT) AS CapturedDateID,
54      i.DownloadKbps,
55      i.UploadKbps,
56      i.TotalTests,
57      i.DistanceMiles
58  FROM Staging.ISPDailySpeed i
59      INNER JOIN Dim.ISPLocation d
60          ON d.ISPName = i.ISPName
61          AND d.City = i.City
62          AND d.RegionCode = i.RegionCode
63          AND d.CountryCode = i.CountryCode
64  ORDER BY i.ISPDailySpeedID
65  OFFSET @StartID ROWS
66  FETCH NEXT @BatchSize ROWS ONLY
```

batching t-sql

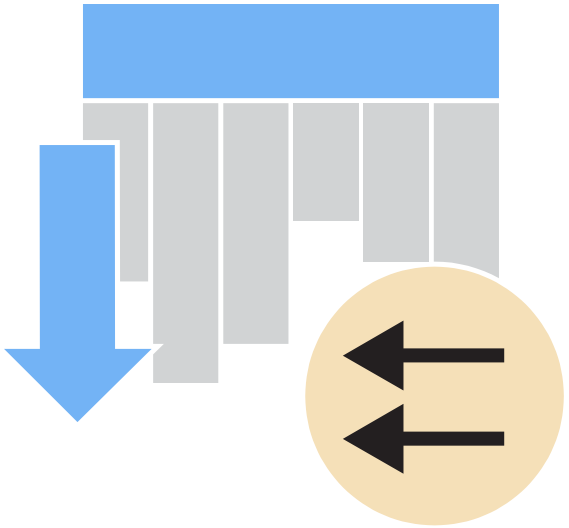
Great for:

1. Transferring data in the same instance
2. Low priority data transfer with long duration
3. Batching scripts can be converted to stored proc – ideal for replication
4. TempDB size is limited
5. Transaction Log is generally small for Simple or Bulk Logged recovery mode

tricky scenarios

- Get a unique list of a number of columns from 100 Million rows
- Joining 100 million rows with another 100 million rows on NVARCHAR(500) columns
- Transaction requirement – i.e. All rows have to be processed and committed as one transaction

columnstore indexes – short intro



Microsoft® SQL Server® 2012

- Group and store data for each column and then join all the columns to complete the whole index
- Enable faster performance for common data warehousing queries such as filtering, aggregating, grouping, and star-join queries.

Source:
http://msdn.microsoft.com/en-us/library/gg492088(v=SQL.110).aspx

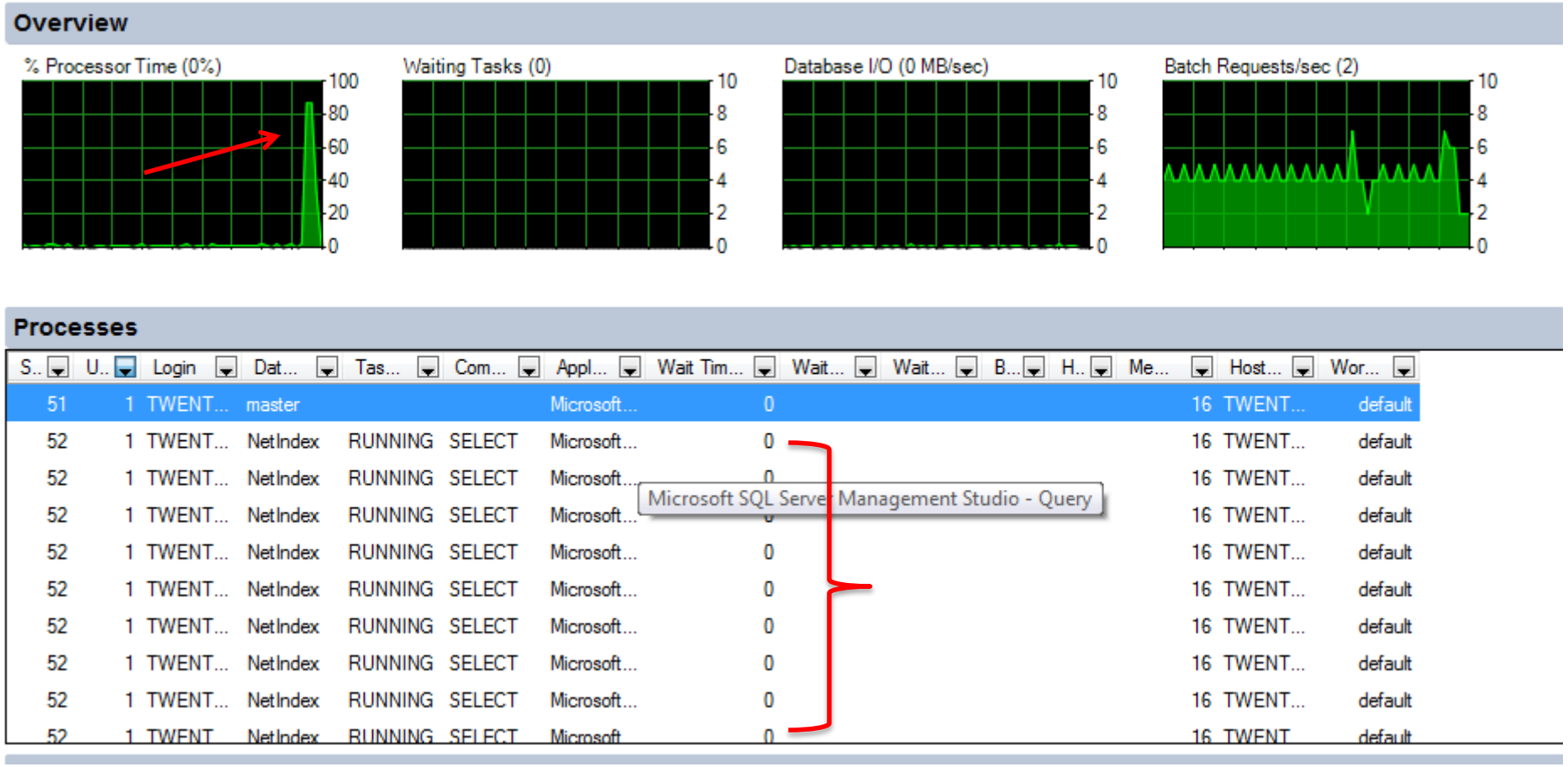
demo

```
13
14  CREATE COLUMNSTORE INDEX IXCS_Fact_ISPDailySpeed_ColumnStore ON Fact.ISPDailySpeed_ColumnStore
15  (
16      ISPLocationID,
17      CapturedDateID,
18      DownloadKbps,
19      UploadKbps
20  );
21
```

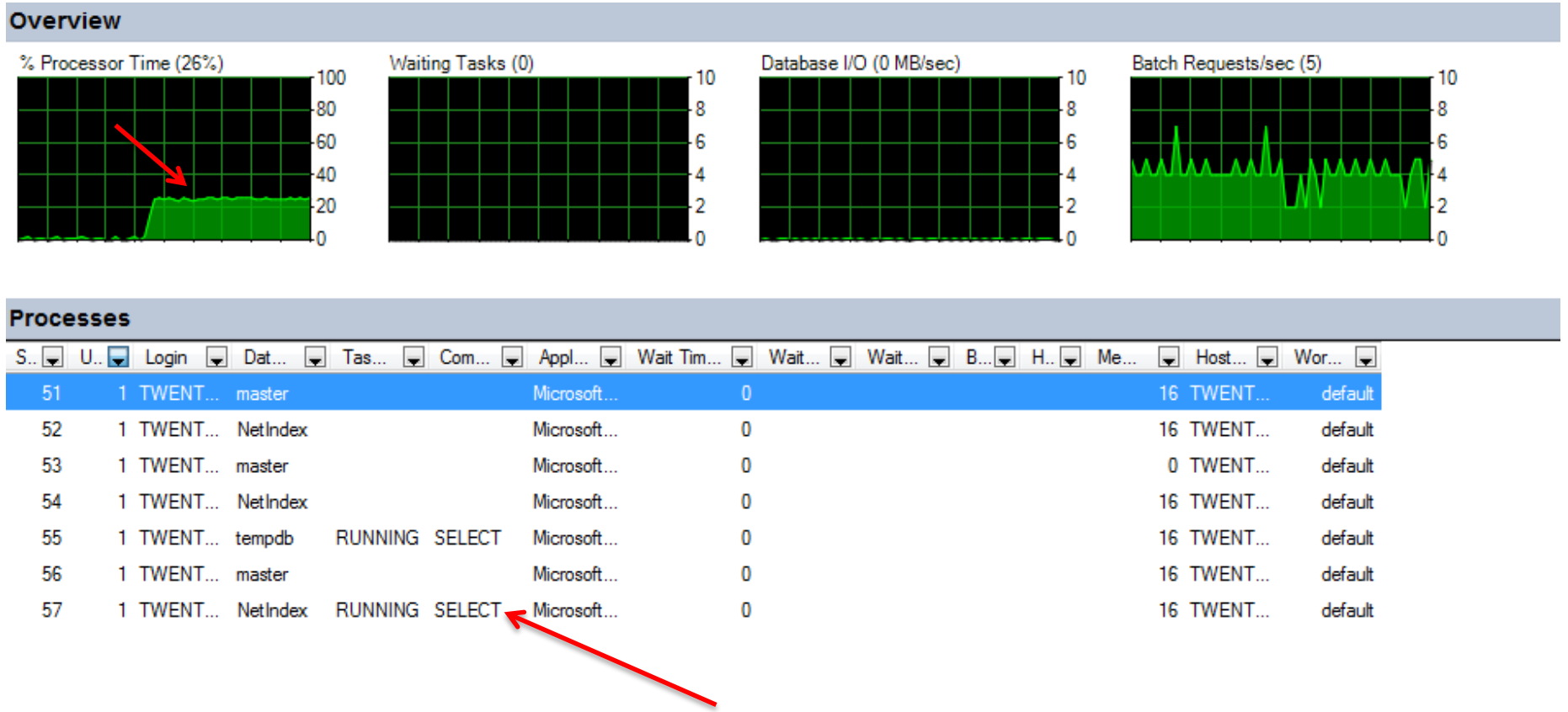
VS

```
6
7   CREATE NONCLUSTERED INDEX IX_Fact_ISPDailySpeed_ISPLocationID_CapturedDateID_Speed ON Fact.ISPDailySpeed
8   (
9       ISPLocationID ASC,
10      CapturedDateID ASC,
11      DownloadKbps ASC,
12      UploadKbps ASC
13  );
14
```

columnstore index



non clustered index



columnstore vs non clustered index

	Columnstore Index	Non Clustered Index
Total Size in MB	146	510
Creation Time in Seconds	98	19
Query Time in Seconds	3	38

Columnstore index < 33% of the size, > 10x faster* than equivalent Non Clustered index

* Results may vary depending on environment

columnstore indexes – restrictions

Selective Data Types	Read Only	No Clustered Indexes No PK/FK/UQ
No Sparse Columns	 See BOL	No Include Keyword
No Replication	No Page / Row Compression	No Views

ssis

ssis - data flow task

Minimally Logged Inserts

1. Transaction Log keeps track of extent allocations and metadata changes only
2. If no sorting required, it won't use TempDB –
   i.e. no additional read/write

In Memory pipeline

Fastest when inserting data as Heap

Bulk Upload Lock – multiple concurrent threads

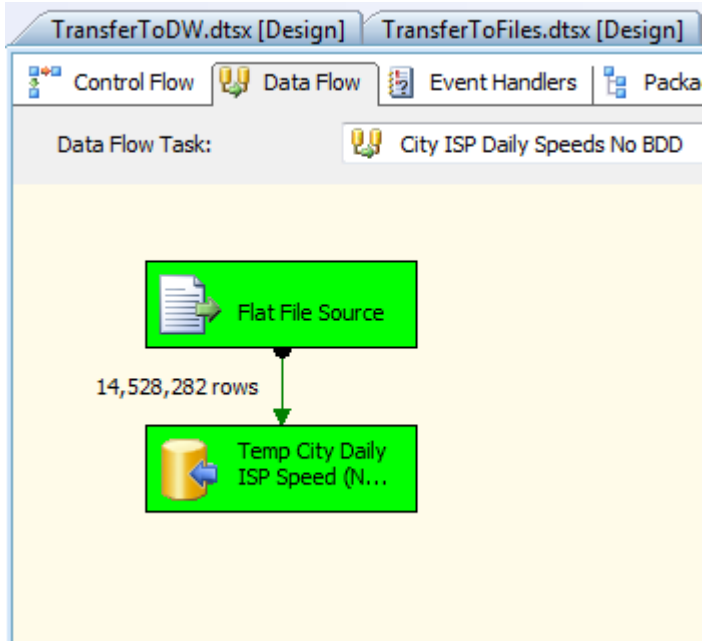
Heavy transformation offloaded to isolated SSIS server

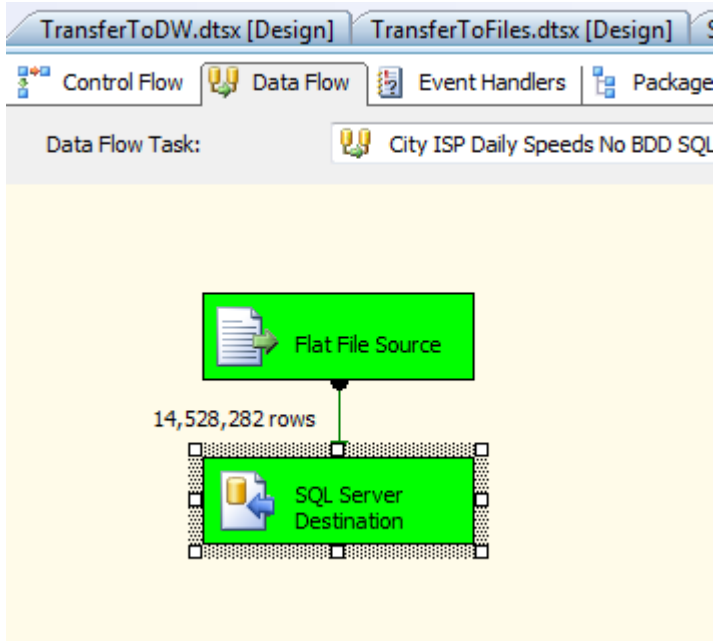
minimally logged inserts

Requirements:

1. Simple or Bulk Logged recovery model on Target database
2. Empty target table
3. Non clustered indexes are disabled
4. Data arriving in the same order of Clustered Index (if any)
5. Table Lock
6. SQL Server Destination (faster) or OLE DB Destination

demo



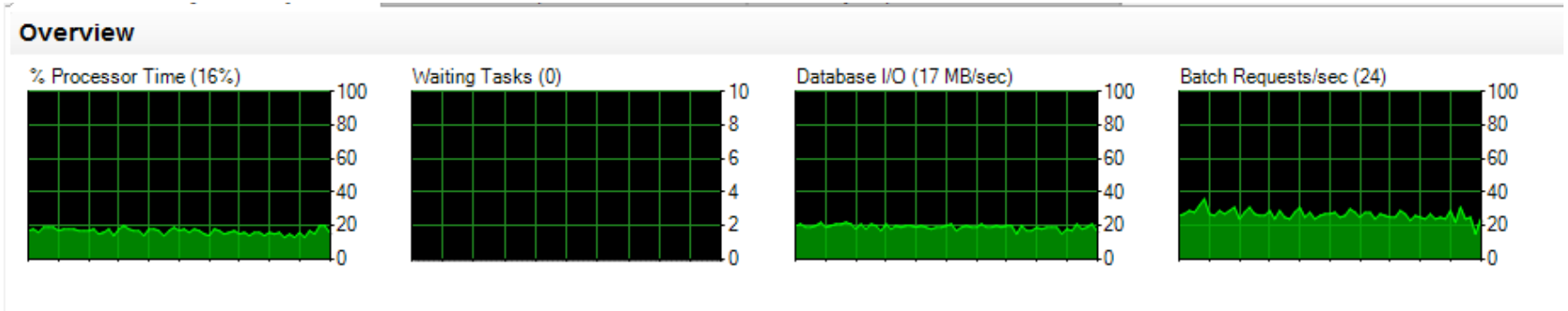


00:03:58
OLEDB

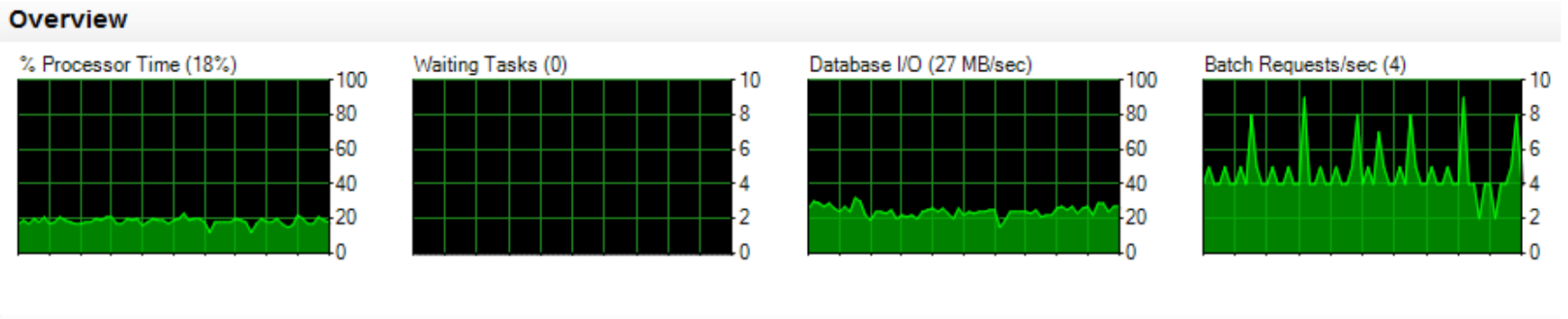
00:02:11
SQL Server Destination

demo

OLE DB Destination – flat table no index, no PK



SQL Server Destination – flat table no index, no PK



all about transaction log

existing data	clustered index	non clustered index	data page updates	index page updates
No	No	No	Minimally Logged	n/a
No	No	Yes	Minimally Logged	Minimally Logged
No	Yes	Doesn't matter	Minimally Logged	Minimally Logged
Yes	No	No	Minimally Logged	n/a
Yes	No	Yes	Minimally Logged	Fully Logged
Yes	Yes	Doesn't matter	Fully Logged	Fully Logged

Source: http://www.mssqltips.com/sqlservertip/1185/minimally-logging-bulk-load-inserts-into-sql-server/

ssis – data flow task configurations

SSIS – controlling transaction log and tempdb growth:

- Rows per batch
- Maximum Insert Commit Size

Limiting Memory/CPU usage:

- Resource Governor
- SQL Server Max Memory size

ssis – tuning guide

SSIS memory settings

- DefaultBufferSize
- DefaultBufferMaxRows
- EngineThreads

SSIS Operational and Tuning Guide for SQL Server 2012:
http://technet.microsoft.com/en-us/library/jj873729.aspx

ssis data flow task

Great for:

1. Transferring data from different instances/formats
2. Quick data transfer with short duration and minimal logging*
3. Lots of tuning options (also lots to learn!)
4. Can offload transformation on a separate server

design consideration

Insert

- Disable Indexes in Destination table before and rebuild after
- Disable Foreign Keys in Destination table before and rebuild after

Update & Delete

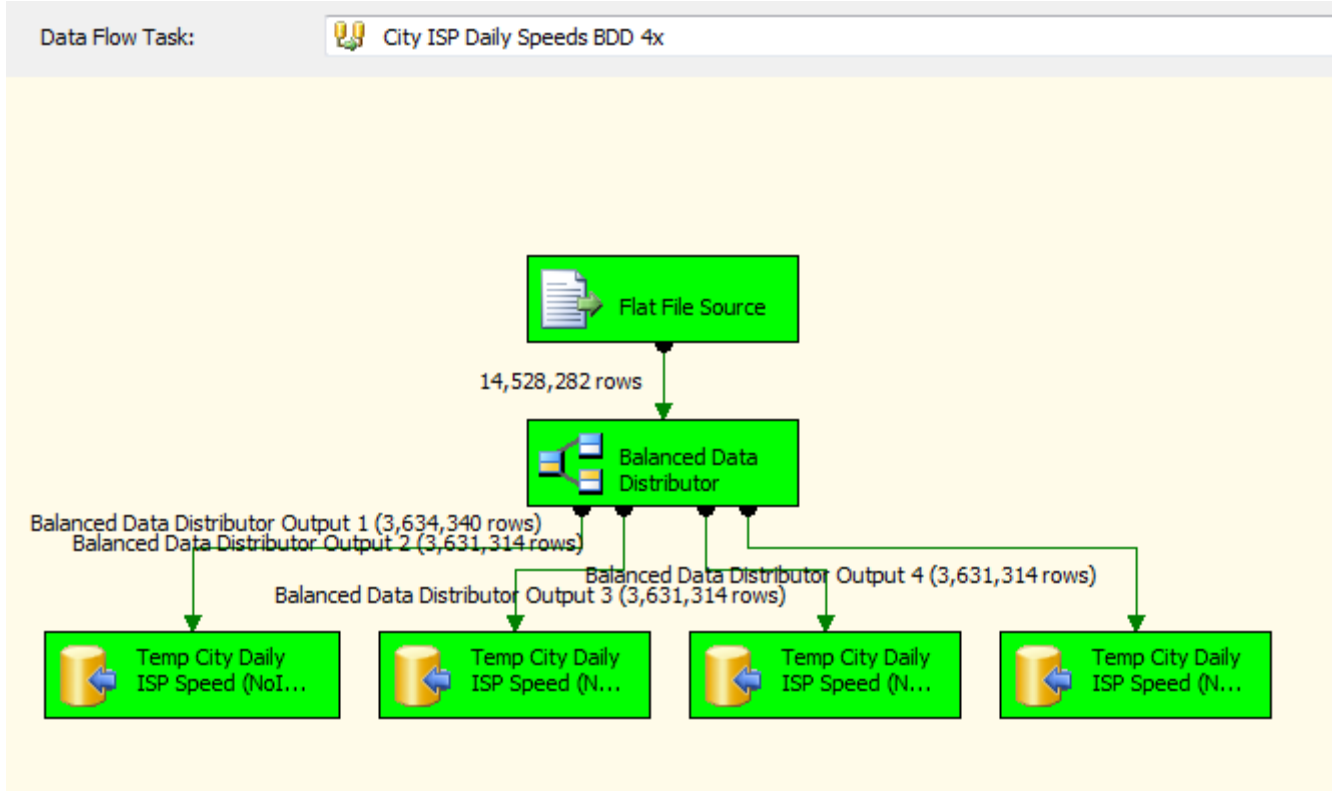
- Reload Destination table instead of update / delete

balanced data distributor

balanced data distributor

- Parallelism in SSIS Data Flow
- Equal proportion of “pipes” to the same destination
- SQL Server 2008 & SQL Server 2012
- Runs on Windows 7, Windows 8 and Windows Server 2008 only

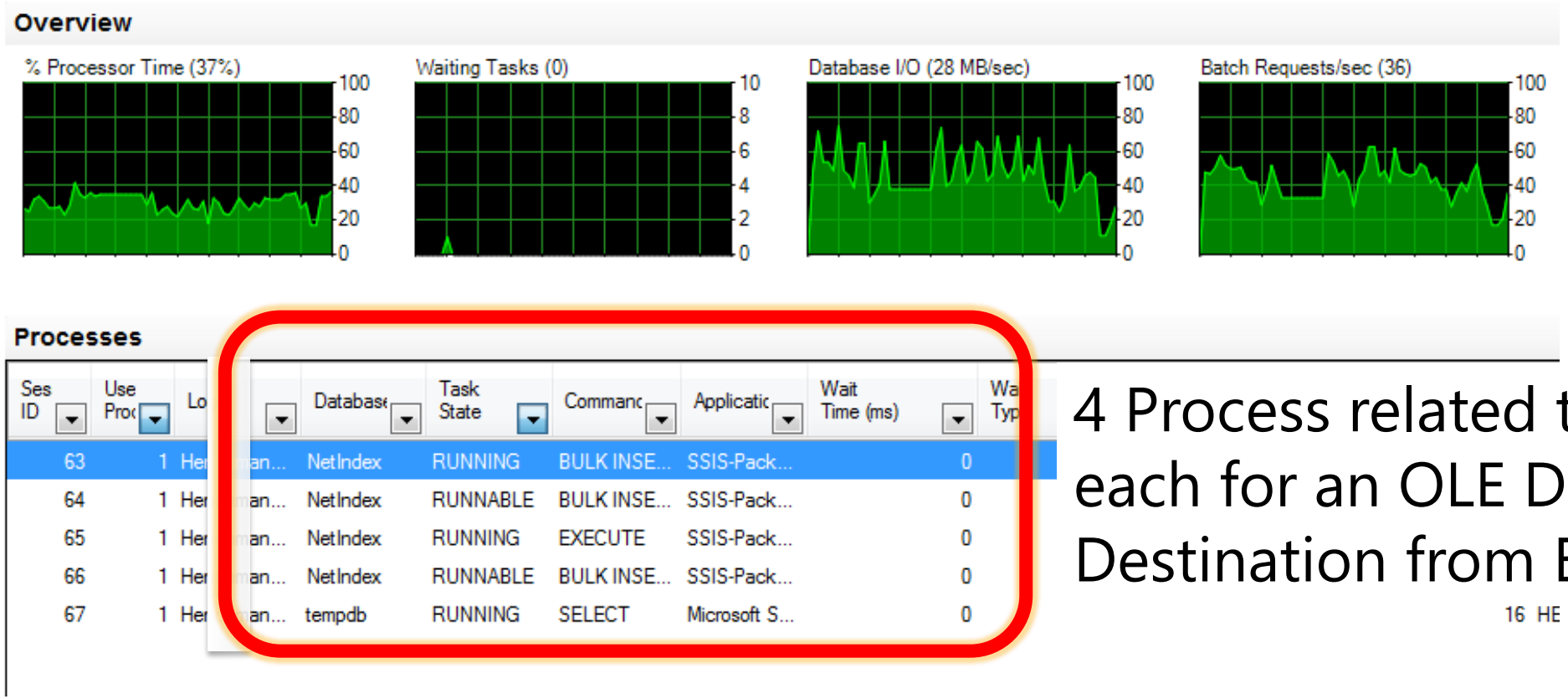
demo



00:01:54 to import 14,528,282 rows into an empty table

Using Balanced Data Distributor with 4 OLE DB Destinations

demo



4 Process related to SSIS – each for an OLE DB Destination from BDD

balanced data distributor

Great for

1. Inserting lots of records
2. Taking advantage of multi-processor and multi-core servers
3. Follow similar rules to SSIS Data Flow Task
4. Data transfer without specific order
5. Bottleneck is in transformation or destination

other techniques

other techniques

Inserting data files into database

- bulk insert
- bcp

Performance Tuning / Index optimization

Simplify design

wrap up

wrap up

	Microsoft SQL Server 2008 R2	Microsoft SQL Server 2012
TSQL	Yes	Yes + Columnstore Index
SSIS Data Flow Task	Yes	Yes
Balanced Data Distributor	Yes	Yes

remember!

Always TEST first

The Right Techniques for the Right Situation

Great Solution = Happy Users = Happy DBA

q & a

Contact details:

Email signal@mssqlgirl.com

Blog http://www.mssqlgirl.com

Twitter @mssqlgirl

LinkedIn http://www.linkedin.com/in/juliekoesmarno

further reading

SQL Server 2012

ORDER BY Clause (OFFSET .. FETCH NEXT ...)

http://msdn.microsoft.com/en-us/library/ms188385%28v=SQL.110%29.aspx

Paging Function Performance in SQL Server 2012

http://www.mssqlgirl.com/paging-function-performance-in-sql-server-2012.html

Columnstore Indexes

http://msdn.microsoft.com/en-us/library/gg492088(v=SQL.110).aspx

http://social.technet.microsoft.com/wiki/contents/articles/3540.sql-server-columnstore-index-faq.aspx

SSIS

The Data Loading Performance Guide

http://msdn.microsoft.com/en-us/library/dd425070(v=sql.100).aspx

Minimally Logging Bulk Load Inserts into SQL Server

http://www.mssqltips.com/sqlservertip/1185/minimally-logging-bulk-load-inserts-into-sql-server/

BDD

The "Balanced Data Distributor" SSIS

http://blogs.msdn.com/b/sqlperf/archive/2011/05/25/the-balanced-data-distributor-for-ssis.aspx

Download Center – BDD

http://www.microsoft.com/download/en/details.aspx?id=4123

thank you
for attending this session