Quanto mi costa utilizzare SQL Pool Servless?

Marco Pozzan

Chi sono?

- Consulente e formatore in ambito business intelligence, business analytics e data mining
- Dal 2017 mi occupo della modern data warehouse con prodotti Azure: Synapse, Azure Data Factory, Stream Analytics, Data Lake
- Dal 2002 le attività principali sono legate alla progettazione di data warehouse relazionale e alla progettazione multidimensionale con strumenti Microsoft.
- Docente all'Università di Pordenone nel corso Architetture Big Data e DWH: Tecniche di modellazione del dato
- Community Lead di 1nn0va (www.innovazionefvg.net)
- MCP, MCSA, MCSE, MCT SQL Server
- dal 2014 MVP per SQL Server e relatore in diverse conferenze sul tema.
  - info@marcopozzan.it
  - @marcopozzan.it
  - www.marcopozzan.it
  - http://www.scoop.it/u/marco-pozzan
  - http://paper.li/marcopozzan/1422524394

MVP Microsoft Most Valuable Professional Reconnect

Microsoft CERTIFIED AZURE FUNDAMENTALS

Microsoft CERTIFIED Trainer

Microsoft CERTIFICATION EXAM DP-200

Microsoft CERTIFIED Professional

Microsoft CERTIFIED Solutions Associate SQL Server 2012/2014

Agenda

- Recap su Sql pool Servless
- Come sono calcolati I costi in SQL pool Servless
- Demo
- Risultati
- Conclusioni

Recap Sql pool Servless

- Il pool Serverless SQL è una grande novità quando si tratta di gestire grandi volumi di dati semi-strutturati o non strutturati.
- Il più grande vantaggio del pool SQL Serverless è che puoi interrogare i dati direttamente dai file CSV, parquet o JSON, archiviati nel tuo Azure Data Lake, senza la necessità di trasferire i dati!
- Inoltre, puoi scrivere un semplice T-SQL per recuperare i dati direttamente dai file!
- Ma vediamo come funziona nella realtà in vari casi d'uso e, la cosa più importante, quanto ti costerà ciascuna delle soluzioni!

Che cosa è un file parquet

- Nei file Parquet, i dati vengono compressi in modo più ottimale. Un file parquet consuma circa 1/3 della memoria rispetto a un file CSV che contiene la stessa porzione di dati
- I file Parquet supportano il formato di archiviazione per colonna: le colonne all'interno del file Parquet sono fisicamente separate, il che significa che non è necessario scansionare l'intero file se hai bisogno di dati solo da poche colonne! Al contrario, quando esegui una query su un file CSV, ogni volta che invii la query, eseguirà la scansione dell'intero file, anche se hai bisogno di dati da una singola colonna
- Per chi proviene dal mondo SQL tradizionale, si può pensare a CSV vs Parquet, come database row-store vs database colonnari

Quanto costa il servizio di Sql pool Servless

- l'utilizzo del pool SQL Serverless viene addebitato in base al volume dei dati elaborati (attualmente il prezzo parte da 5 $ / TB di dati elaborati),

- Per questo motivo ci dobbiamo concentrare esclusivamente sull'analisi del volume di dati elaborati e dei costi generati esaminando diversi scenari di utilizzo

Quanto costa il servizio di Sql pool Servless

- l'utilizzo del pool SQL Serverless viene addebitato in base al volume dei dati elaborati (attualmente il prezzo parte da 5 $ / TB di dati elaborati),

- Per questo motivo ci dobbiamo concentrare esclusivamente sull'analisi del volume di dati elaborati (Data Processed) e dei costi generati esaminando diversi scenari di utilizzo

Data Processed (Dati Eleaborati)

I dati elaborati sono la quantità di dati che il sistema memorizza temporaneamente durante l'esecuzione di una query. I dati trattati sono costituiti dalle seguenti quantità:

- Quantità di dati letti dalla memoria. Questo importo include:
  - Dati letti durante la lettura dei dati.
  - Dati letti durante la lettura dei metadati (per formati di file che contengono metadati, come Parquet).

- Quantità di dati nei risultati intermedi. Questi dati vengono trasferiti tra i nodi durante l'esecuzione della query. Include il trasferimento dei dati al tuo endpoint, in un formato non compresso.

Data Processed (Dati Eleaborati)

- Quantità di dati scritti nella memoria. Se si utilizza CETAS per esportare il resultset nella memoria, la quantità di dati scritti viene aggiunta alla quantità di dati elaborati per la parte SELECT di CETAS.

Cosa non viene addebitato (~~Data Processed~~)

- Metadati a livello di server (accessi, ruoli e credenziali a livello di server)
- Database che crei nel tuo endpoint. Questi database contengono solo metadati (utenti, ruoli, schemi, viste, funzioni in linea con valori di tabella, stored procedure, origini dati esterne, formati di file esterni e tabelle esterne)
- Istruzioni DDL, ad eccezione dell'istruzione CREATE STATISTICS perché elabora i dati dalla memoria in base alla percentuale di campione specificata
- Qyery di soli metadati

Come calcolare il volume di dati elaborati

- Esiste una vista di sistema che permette di sapere quanti mega sono stati elaborati dal motore

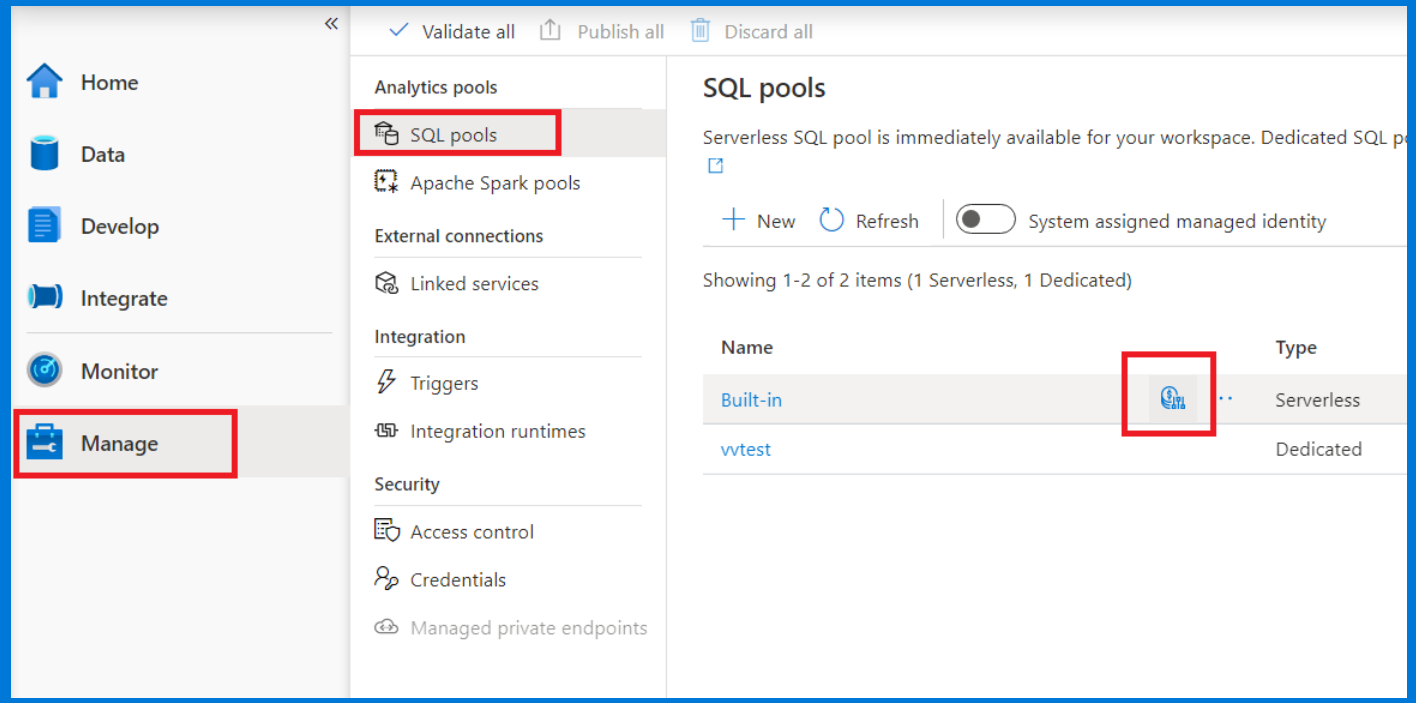
```
SELECT * FROM sys.dm_external_data_processed
```

```
SELECT * FROM sys.dm_external_data_processed
WHERE type = 'daily'
```

- La quantità di dati elaborati viene arrotondata al MB più vicino per query. Ogni query ha un minimo di 10 MB di dati elaborati

Come limitare i costi

- Possiamo impostare il limite di budget giornaliero, settimanale o mensile attraverso:
  - Interfaccia grafica
  - Script T-SQL




```
sp_set_data_processed_limit @type = N'daily', @limit_tb = 1

sp_set_data_processed_limit @type= N'weekly', @limit_tb = 2

sp_set_data_processed_limit @type= N'monthly', @limit_tb = 3334
```

Attenzione all'inferred schema

- L'inferenza dello schema consente di scrivere rapidamente query ed esplorare i dati senza conoscere gli schemi di file.

- Il costo di questa comodità è che i tipi di dati dedotti possono essere maggiori dei tipi di dati effettivi.

- Ciò accade quando non sono disponibili informazioni sufficienti nei file di origine per assicurarsi che venga utilizzato il tipo di dati appropriato. Ad esempio, i file Parquet non contengono metadati sulla lunghezza massima delle colonne di caratteri. Quindi il pool SQL serverless lo deduce come varchar (8000).

Attenzione all'inferred schema

- È possibile utilizzare sp_describe_first_results_set per controllare i tipi di dati risultanti della query.

```
EXEC sp_describe_first_result_set N'SELECT * FROM
    OPENROWSET(
        BULK N''https://adlsafterhour.dfs.core.windows.net/synapse/DataSat/yellow_tripdata_2020-01.parquet'',
        FORMAT = ''PARQUET''
        ) as test'
```

Is_hidden	Column_ordinal	Name	Is_nullable	System_type_id	System_type_n...	Max_length	Precision	Scale
False	1	VendorID	True	127	bigint	8	19	0
False	2	tpep_pickup_datetime	True	42	datetime2(0)	6	19	0
False	3	tpep_dropoff_datetime	True	42	datetime2(0)	6	19	0
False	4	passenger_count	True	127	bigint	8	19	0
False	5	trip_distance	True	62	float	8	53	0
False	6	RatecodeID	True	127	bigint	8	19	0
False	7	store_and_fwd_flag	True	167	varchar(8000)	8000	0	0

Attenzione all'inferred schema

I tipi di dati utilizzati nella query influiscono sulle prestazioni. Puoi ottenere prestazioni migliori se segui queste linee guida:

- Utilizzare la dimensione dei dati più piccola che soddisfa il valore più grande possibile.
  - Se la lunghezza massima del valore del carattere è 30, utilizza un tipo di dati carattere di lunghezza 30.
  - Se tutti i valori delle colonne di caratteri hanno dimensioni fisse, utilizzare char o varchar. Altrimenti, usa nchar o nvarchar.
  - Se il valore della colonna intero ha un massimo di 500, utilizza smallint perché è il tipo di dati più piccolo che può contenere questo valore.
- Se possibile, usa varchar e char invece di nvarchar e nchar.
- Se possibile, utilizza tipi di dati basati su numeri interi. Le operazioni SORT, JOIN e GROUP BY vengono completate più velocemente sui numeri interi che sui dati carattere.

Utilizzo del CETEAS

CETAS è una delle funzionalità più importanti disponibili nel pool SQL serverless.

CETAS è un'operazione parallela che crea metadati di tabelle esterne ed esporta i risultati della query SELECT in un set di file nell'account di archiviazione.

È possibile utilizzare CETAS per memorizzare parti di query utilizzate di frequente, come tabelle di riferimento unite, in un nuovo set di file. È quindi possibile far riferimento a questa singola tabella esterna invece di ripetere join comuni in più query.

Poiché CETAS genera file Parquet, le statistiche verranno create automaticamente quando la prima query ha come destinazione questa tabella esterna, con conseguente miglioramento delle prestazioni per le query successive che mirano alla tabella generata con CETAS.

Utilizzo del CETEAS

Sintassi del comando

```
CREATE EXTERNAL TABLE [ [database_name  . [ schema_name ] . ] | schema_name . ] table_name
   WITH (
      LOCATION = 'path_to_folder',
      DATA_SOURCE = external_data_source_name,
      FILE_FORMAT = external_file_format_name
)
   AS <select_statement>
[;]

<select_statement> ::=
   [ WITH <common_table_expression> [ ,...n ] ]
   SELECT <select_criteria>
```

```
CREATE EXTERNAL TABLE trip_2020_csv_ext
WITH (
    LOCATION = 'synapse/DataSat/trip_2020_csv_ext',
    DATA_SOURCE = DS_Taxi,
    FILE_FORMAT = exCsv
)
AS
SELECT *
from dbo.trip2020_parquet
```

Demo

- Analizzeremo i costi in 9 diversi scenari di utilizzo di SQL pool Servless
  1. Caricamento Importato in Power BI con csv/parquet (view/external table)
  2. Caricamento DQ in Power BI con csv / parquet (view/external table)
  3. Caricamento DQ in Power BI con csv / parquet (view/external table)
  4. Caricamento DQ in Power BI con csv / parquet e aggregazioni
  5. Caricamento DQ in Power BI con csv / parquet e aggregazioni in memoria

Risultati Demo

Caso	*Costi $ con csv*	*Costi $ con csv ext*	*Costi $ con parquet*	*Costi $ con parquet ext*	*Note*
Test 1: modello importato su CSV in Power BI (2 visuals)	~0.007 per refresh	~0.001 per refresh	~0.002 per refresh	~0.0008 per refresh	Numero di visual e numero di utenti non fanno aumentare I costi
Test 2: modello DQ su CSV in Power BI (2 visuals - 1 utente)	~0.011 per 2 visuals/1 user	~0.002 per 2 visuals/1 user	~0.001 per 2 visuals/1 user	~0.000075 per 2 visuals/1 user	Numero di visual e numero di utenti fanno aumentare I costi
Test 3: modello DQ su CSV in Power BI (2 visuals - 1 utente) filtrando con lo slicer	~0.011 per 2 visuals/1 user	~0.002 per 2 visuals/1 user	~0.001 per 2 visuals/1 user	~0.000075 per 2 visuals/1 user	Numero di visual e numero di utenti fanno aumentare I costi
Test 4: modello DQ su CSV in Power BI (2 visuals - 1 utente) aggregando	~0.011 per 2 visuals/1 user	~0.002 per 2 visuals/1 user	~0.001 per 2 visuals/1 user	~0.000075 per 2 visuals/1 user	Numero di visual e numero di utenti fanno aumentare i costi
Test 5: modello DQ su CSV in Power BI (2 visuals - 1 utente) aggregando in memory	~0.0062 per 2 visuals/1 user	~0.00095 per refresh	~0.00053 per refresh	~0.00003 per refresh	Numero di visual e numero di utenti non fanno aumentare i costi

Conclusioni

- Quando possibile, usa i file Parquet invece di CSV
- Quando possibile, importa i dati in Power BI: ciò significa che pagherai solo quando lo snapshot dei dati verrà aggiornato, non per ogni singola query all'interno del report
- Se hai a che fare con file Parquet, quando possibile, crea dati pre-aggregati (viste) nel pool SQL Serverless in Synapse
- indipendentemente da ciò che esegui all'interno del pool SQL Serverless sopra i file CSV, saranno completamente scansionati al livello più basso del processo di preparazione dei dati!

Conclusioni

- Poiché il pool SQL Serverless non supporta ancora la cache del resultset (il team di Microsoft ci sta lavorando), tieni presente che ogni volta che esegui la query (anche se stai restituendo lo stesso set di risultati), verrà generata una query e dovrai pagarla!

- Se le analisi richiedono un numero elevato di query su un grande dataset (così grande che la modalità di importazione non è un'opzione), forse dovresti prendere in considerazione l'archiviazione dei dati nel pool SQL dedicato, poiché pagherai costi di archiviazione fissi, invece dei dati costi di elaborazione ogni volta che si interrogano i dati.

Conclusioni

- non è solo la quantità di dati scansionati che fa il totale dei dati elaborati, ma anche la quantità di dati in streaming a un client. Più i dati sono grandi più il costo è elevato Quindi, tieni presente che la pre-aggregazione dei dati all'interno di un pool SQL Serverless può farti risparmiare la quantità di dati in streaming

- Attenersi alle migliori pratiche generali quando si utilizza il pool SQL Serverless in Synapse Analytics https://docs.microsoft.com/en-us/azure/synapse-analytics/sql/best-practices-sql-on-demand

Conclusioni

- Attenersi alle migliori pratiche generali quando si utilizza il pool SQL Serverless in Synapse Analytics https://docs.microsoft.com/en-us/azure/synapse-analytics/sql/best-practices-sql-on-demand

«A mio parere, anche se Synapse ha ancora molta strada da fare con il pool SQL Serverless, non c'è dubbio che si stia muovendo nella giusta direzione»

«il costante miglioramento del prodotto e l'aggiunta regolare di nuove funzionalità, può davvero essere un unico punto di accesso per i carichi di lavoro sui dati»

Grazieee