Data Lake Optimization Techniques

- **Data Ingestion considerations**
  - Storage hardware
  - High speed internal network
  - Fast network connection b/w on-premises and cloud
- **Parallel read/write**
  - e.g. Data Factory parallel copies settings
- **Structure your data set**
  - File size vs number of files
    - File size b/w 256 MB to 100 GB
  - Folder and file structure
    - e.g. Dataset\YYYY\MM\DD\datafile_YYYY_MM_DD_HH_MM.tsv
- **Same region**
- **Batch Data**

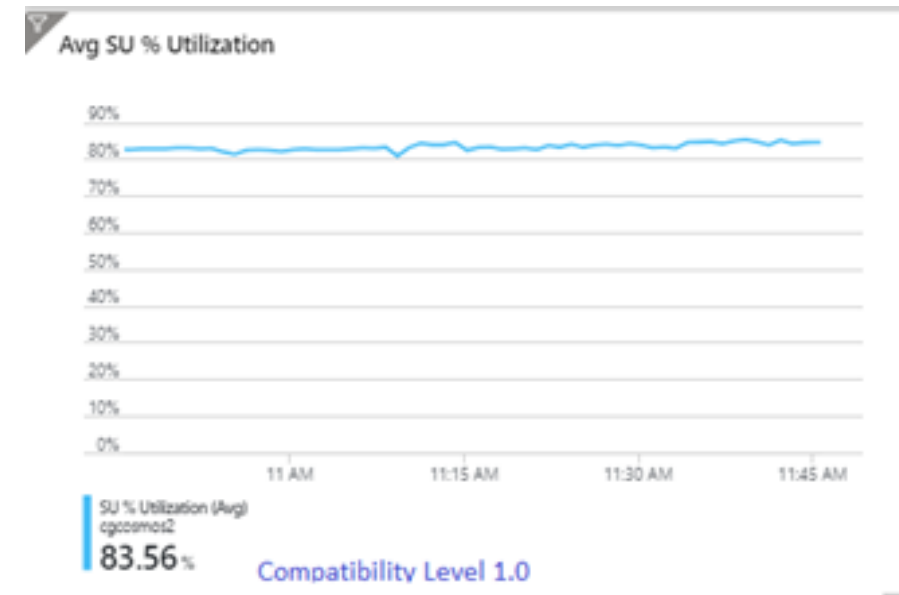# Stream Analytics – Optimization

Stream Analytics – Optimization

Three main component

- Input

- output

- Data processing Query

Stream Analytics – Optimization

Streaming Units (SUs)

- Processing power (CPU and Memory) allocated to your stream analytics job.

- Azure Stream Analytics jobs perform all processing in memory

- If SU% utilization is low and input events get backlogged

- Microsoft recommends setting an alert on 80% SU Utilization metric to prevent resource exhaustion

- The best practice is to start with 6 SUs for queries that don't use PARTITION BY

- Complex query logic could have high SU% utilization even when it is not continuously receiving input events.

Stream Analytics – Optimization

Parallelization

- Partitioning helps to divide data in subsets.

- This would be based on partition key.

- If the data in the Event Hub has a partition key defined, then it is highly recommended to define the partition key in the input of Stream Analytics Job.

- Input are already partitioned, output needs to be partitioned

- Embarrassingly parallel jobs
  - An embarrassingly parallel job is the most scalable scenario in Azure Stream Analytics.
  - It connects one partition of the input to one instance of the query to one partition of the output.
  - The number of input partitions must equal the number of output partitions.

```SQL
SELECT *
INTO output
FROM input
PARTITION BY DeviceID
INTO 10
```

Stream Analytics – Optimization
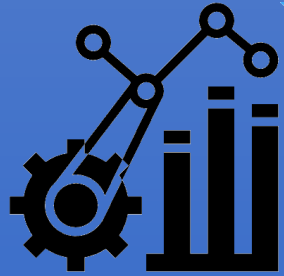
Steps in Query

- You can have multiple step in a query.

- You can start with 6 SUs for queries that don't use PARTITION BY

- You can also add 6 streaming units for each partition in a partitioned step.

- Example:

  - Let's say your input stream is partitioned by value of 10, and you only have one step in query

```SQL
SELECT *
INTO output
FROM input
PARTITION BY DeviceID
INTO 10
```
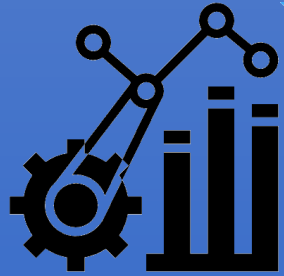
Stream Analytics – Optimization

Event Hubs → INPUT → Streaming analytics job → OUTPUT → Azure Blob Storage

Synapse Analytics Optimization

- **Maintain Statistics**
  - Automatically detect and create statistics on columns
  - AUTO_CREATE_STATISTICS
  - Update statistics of more relevant columns like date (or columns used in joins, where and group by clause)
- **PolyBase**
  - ADF or BCP can be used for small load
  - PolyBase is best choice for large volume of data
  - MPP architecture
  - CTAS or INSERT INTO
- **Hash distribution large tables**
  - Default is Round Robin distribution
  - Small tables joins – Round Robin is fine
 - Big tables joins – use Hash Distribution

Synapse Analytics Optimization

- **Do not over partition**
  - Too many partition can slow down query
  - Partition should have more than 1 million rows
  - 60 partition by default
  - So if you manually create 100 partition, behind the scene it is 100*60 = 6000 partitions.
- **Use the smallest possible column size**
  - Important for char or varchar type columns
  - Use varchar instead nvarchar
- **Scaling**
  - Before you perform a heavy data loading or transformation operation
  - During peak business hours
- **Pausing and resuming compute**
  - Storage and compute are seperate
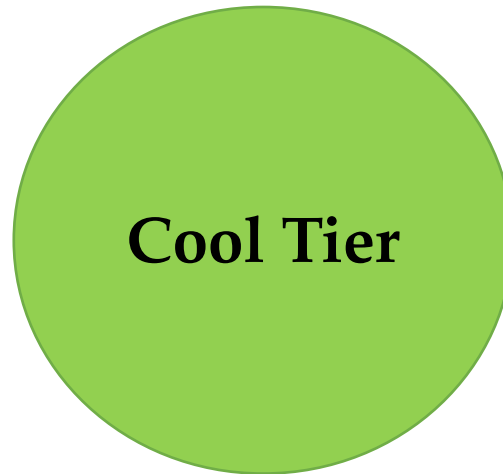 - Transaction cancel

Managing Data Lifecycle

Blob Access Tiers

Hot Tier

Cool Tier

Archive Tier
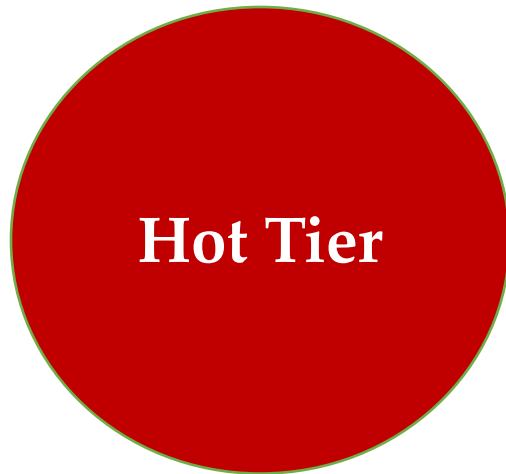
Highest storage cost
Lowest data access cost

Lowest storage cost
Higher data access cost

Lowest storage cost
Highest data retrieval cost
Data is offline

Azure Blob Storage Lifecycle Management

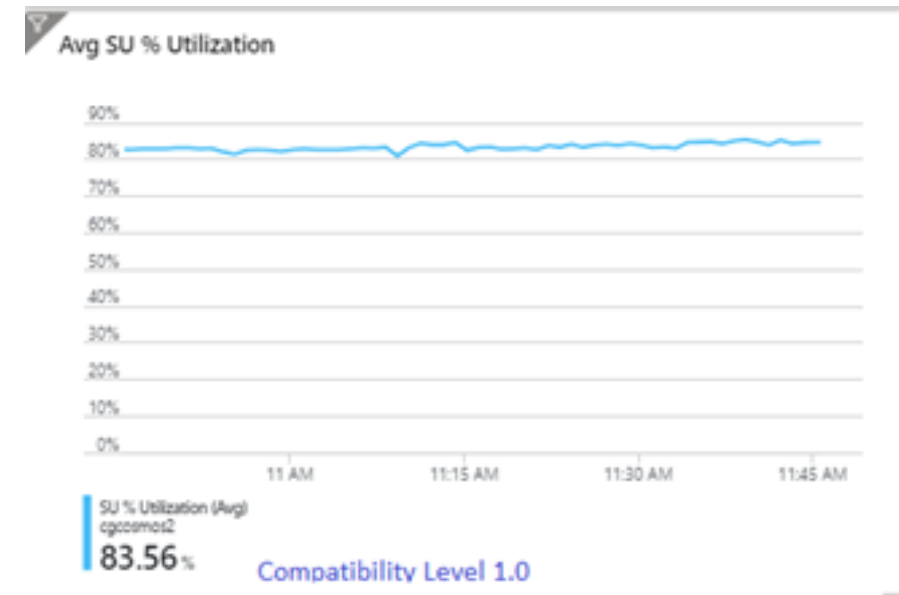# Stream Analytics – Optimization

Stream Analytics – Optimization

Three main component

- Input

- output

- Data processing Query

Stream Analytics – Optimization

Streaming Units (SUs)

- Processing power (CPU and Memory) allocated to your stream analytics job.

- Azure Stream Analytics jobs perform all processing in memory

- If SU% utilization is low and input events get backlogged

- Microsoft recommends setting an alert on 80% SU Utilization metric to prevent resource exhaustion

- The best practice is to start with 6 SUs for queries that don't use PARTITION BY

- Complex query logic could have high SU% utilization even when it is not continuously receiving input events.

Stream Analytics – Optimization

Parallelization

- Partitioning helps to divide data in subsets.

- This would be based on partition key.

- If the data in the Event Hub has a partition key defined, then it is highly recommended to define the partition key in the input of Stream Analytics Job.

- Input are already partitioned, output needs to be partitioned

- Embarrassingly parallel jobs
  - An embarrassingly parallel job is the most scalable scenario in Azure Stream Analytics.
  - It connects one partition of the input to one instance of the query to one partition of the output.
  - The number of input partitions must equal the number of output partitions.

```SQL
SELECT *
INTO output
FROM input
PARTITION BY DeviceID
INTO 10
```

Stream Analytics – Optimization

Steps in Query

- You can have multiple step in a query.

- You can start with 6 SUs for queries that don't use PARTITION BY

- You can also add 6 streaming units for each partition in a partitioned step.

- Example:

  - Let's say your input stream is partitioned by value of 10, and you only have
    one step in query

```SQL
SELECT *
INTO output
FROM input
PARTITION BY DeviceID
INTO 10
```

Stream Analytics – Optimization

Event Hubs → INPUT → Streaming analytics job → OUTPUT → Azure Blob Storage