Solid Quality Learning

ETL09

Mastering and Monitoring the Data Flow

SSIS ETL Course Outline

- ETL01: SSIS Overview
- ETL02: Control Flow
- ETL03: Data Flow
- ETL04: Data Extraction and Lineage
- ETL05: Dimension ETL
- ETL06: Fact Table ETL
- ETL07: Configuration, Logging, and Restartability
- ETL08: Execution, Deployment, and Security
- ETL09: Mastering and Monitoring the Data Flow
- ETL10: Case Study: Project REAL

Module Agenda

- Pipeline Buffer Design and <LineageID>
- SSIS Engine
  - Transformation Types
  - Execution Trees, Execution Threads
- Optimization Techniques
- Monitoring and Logging
  - Pipeline Events
  - Performance Monitor Counters

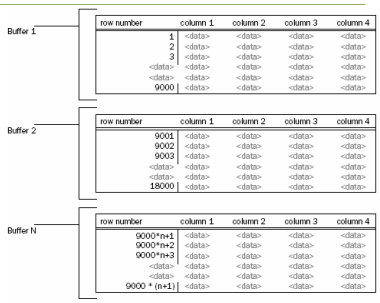
Solid Quality Learning

Pipeline Buffers

- Data extracted into the Data Flow is passed into data buffer groupings
- Transformations logic flows over buffers for optimal performance
- Buffers allow a “streaming” process
- Each Data Flow can have multiple buffer profiles defined for different types of data

Solid Quality Learning

Pipeline Buffers



Solid Quality Learning

Data Flow Architecture

- Buffers based on design time metadata
  - The width of a row determines the size of the buffer
  - Smaller rows = more rows in memory = greater efficiency
- Memory copies are expensive!
  - Pointer magic where possible
  - E.g. Multicast – logical vs. actual

Solid Quality Learning

Column <LineageID>

- Each buffer profile assigns a <LineageID> to each column (not to be confused with ETL Lineage tracking)
- A single column coming in from a source can end up having multiple LineageIDs through its life in the pipeline
- LineageIDs are viewable in the Advanced Input and Output Properties

Solid Quality Learning

ETL09-Demo-A

Buffers and LineageIDs

Solid Quality Learning

Agenda

- Pipeline Buffer Design and <LineageID>
- SSIS Engine
  - Transformation Types
  - Execution Trees, Execution Threads
- Optimization Techniques
- Monitoring and Logging
  - Pipeline Events
  - Performance Monitor Counters

Solid Quality Learning

Component Types

- Streaming (synchronous)
  - Logically works at a row level
  - Buffer Reused
  - ***Examples:*** *Data Convert, Derived Column, Lookup*
- Partially blocking (asynchronous)
  - May logically work at a row level
  - Data copied to new buffers
  - ***Examples:*** *Pivot, Un-pivot, Merge, Merge Join, Union All*
- Blocking (asynchronous)
  - Needs all input buffers before producing any output rows
  - Data copied to new buffers
  - ***Examples:*** *Aggregate, Sort, Row Sampling, Fuzzy Grouping*

Solid Quality Learning

Synchronous and Asynchronous

- Think transformation communication
- Definitions apply to transformation outputs
- Synchronous transformation outputs
  - Same buffers immediately passed onto next transformation
  - No rows added, no rows removed
- Asynchronous transformation outputs
  - Data is copied to new buffer
  - Downstream transformation works independently of upstream asynchronous transformation

Solid Quality Learning

Synchronous and Asynchronous

- A Transform is not limited to a single synchronous output
  - Multicast and Conditional Split have multiple synchronous outputs
- Synchronous outputs preserve the sort order of the input rows
- Identifying Synchronicity
  - See SynchronousInputID property in Advanced Editor | Output property
  - Entry of 0 identifies an asynchronous transformation

Solid Quality Learning

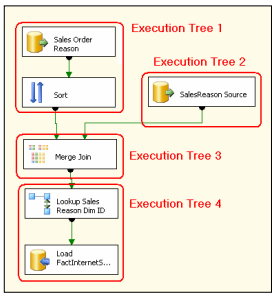
Execution Trees and Threads

- Execution Trees
  - Start from a source or an asynchronous output
  - Ends at a destination or an input that has no sync outputs
  - Different buffer profiles per tree
- Execution Threads
  - Each Source can get a thread
  - Each Execution Tree can get a thread
- Use EngineThreads to control parallelism
  - Value applies to Execution Trees, not Sources

Solid Quality Learning

Execution Trees and Threads

- Each component within an execution tree applies work on the same set of buffers
- Data in a new execution tree requires existing buffer data to be copied into new buffers



Solid Quality Learning

ETL09-Demo-B

Identifying Data Flow Transformation Types and Execution Trees

Solid Quality Learning

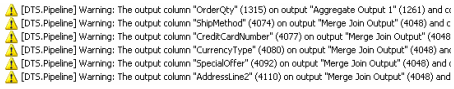
Agenda

- Pipeline Buffer Design and <LineageID>
- SSIS Engine
  - Transformation Types
  - Execution Trees, Execution Threads
- Optimization Techniques
- Monitoring and Logging
  - Pipeline Events
  - Performance Monitor Counters

Solid Quality Learning

Data Flow Optimization

- Increase Engine Threads
- Breakup large Execution Trees with a Union All transformation to allow a more process threads to handle operations
- Remove columns in pipeline not used downstream (avoid pipeline warnings)



Solid Quality Learning

Data Flow Optimization

- Limit Rowbased operations
- Limit Blocking Transforms (presort if possible)
- Perform data correlation in the Data Flow
- Handle staging requirements with a Multicast transformation
- Use strategic staging to optimize pipeline
  - Limit lookup cache
  - Filter source queries

Solid Quality Learning

When to Stage Data

- Restartability requirements
- Process window times and precedence
- Intense downstream transformations causes source back pressure (slows extraction)
- Data Flow optimization
- Eases complexity

Solid Quality Learning

Agenda

- Pipeline Buffer Design and <LineageID>
- SSIS Engine
  - Transformation Types
  - Execution Trees, Execution Threads
- Optimization Techniques
- Monitoring and Logging
  - Pipeline Events
  - Performance Monitor Counters

Solid Quality Learning

Monitoring the Data Flow

- Pipeline logging events
  - Pipeline Execution Trees
  - Pipeline Execution Plan
- Performance Monitor counters
  - Object = SQLServer: SSIS Pipeline
    - Blob counters
    - Buffer counters
    - Row counts

Solid Quality Learning

Pipeline Logging Events

- Pipeline Execution Trees
  - Defines each execution tree input/output, base 0

```
begin execution tree 0
   output "OLE DB Source Output" (582)
   input "Merge Join Right Input" (686)
end execution tree 0
```

- Pipeline Execution Plan
  - Assigns engine threads to sources and execution trees

```
SourceThread1
  CreatePrimeBuffer of type 4 for output ID 749.
      SetBufferListener: "WorkThread1" for input ID 976
      CallPrimeOutput on component "Prior Weeks" (740)
   End Output Work List
```


Solid Quality Learning

Pipeline PerfMon Counters

- BLOB Columns
  - BLOB columns require large memory overhead
  - BLOBs come from Source Adapters AND the Import Column transformation
- BLOB Counters
  - BLOB bytes read
  - BLOB bytes written
  - BLOB files in use – based on the Import and Export Column transformation (read and write to files)

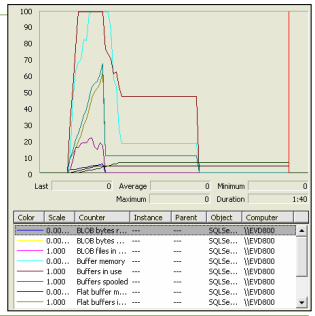
Solid Quality Learning

Pipeline PerfMon Counters

- Buffer types
  - Flat Buffers: primary buffers used in the pipeline
  - Private Buffers: used by individual transformations to perform operations (Sort, Aggregate, Lookup cache)
- Buffer Counters (by Total, Flat, or Private)
  - Buffer memory: Amount of memory used by buffers
  - Buffers in use: number of buffers
  - Buffers spooled: # of buffers spooled to disk

Pipeline PerfMon Counters

- **Rows Read:** aggregate of the rows read since the SSIS service was started
- **Rows Written:** aggregate of the rows written since the SSIS service was started



Solid Quality Learning

ETL09-Demo-C

Using the Log View
Setting up a PerfMon Session

Solid Quality Learning

Summary

Questions?