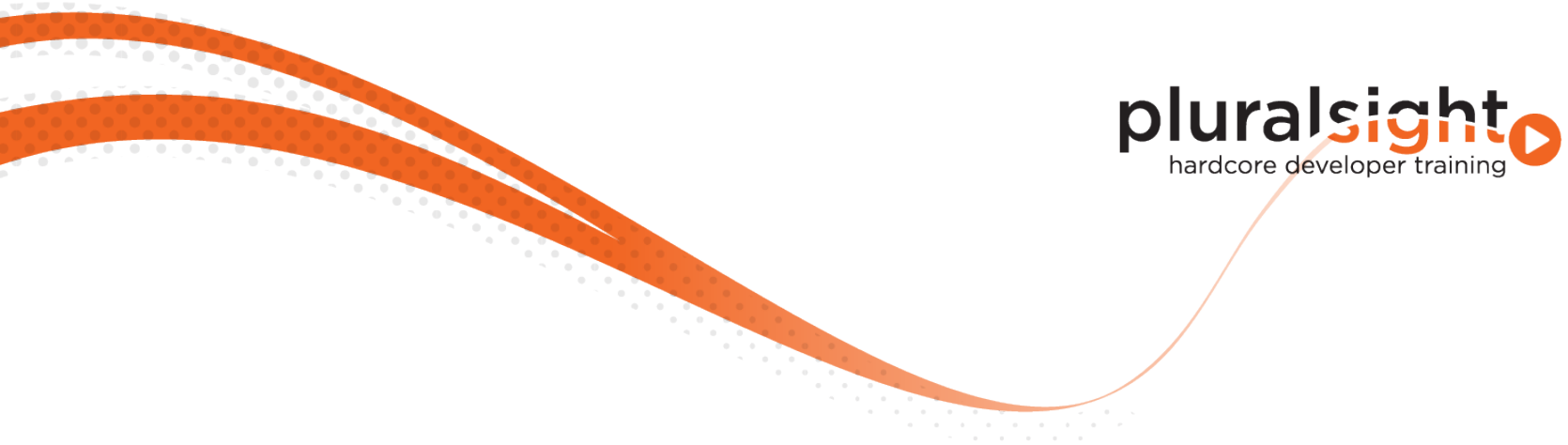
SQL Server: Query Plan Analysis

Module 3: Interpreting Query Execution Plans

Joe Sack

Joe@SQLskills.com



pluralsight
hardcore developer training

Module Introduction

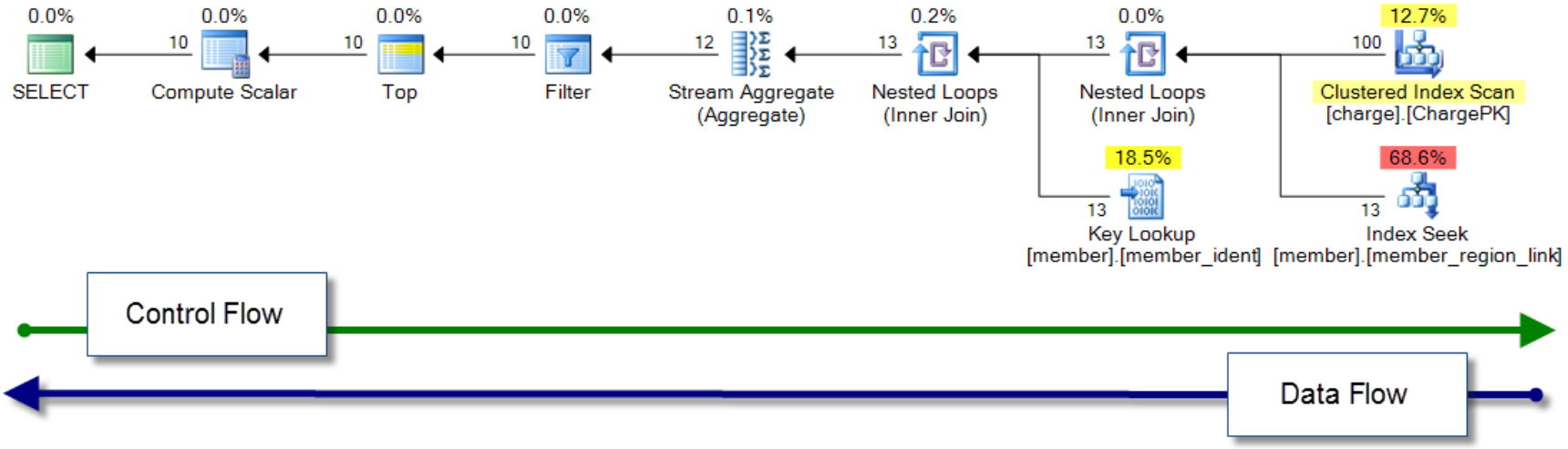
- Once you have a query execution plan, what do you with it?
- This module will cover the fundamental background needed in order to properly interpret query execution plans
- We'll cover the key query execution plan concepts before getting into common operators and noteworthy patterns in later modules

Iterators / Operators

- **The query plan is a tree of operators**
  - Also called iterators
- **SQL Server 2012 has 100+ operators**
  - Logical and physical
- **Think of them as extensible building blocks for a query with each implementing its own functionality**
- **Specific operators are not "good" or "bad"**
  - Some can consume more resources than others or have overhead that you should be aware of
  - Some are more appropriate in given contexts

Query Tree

- Operator reads rows from leaf-level data source OR from child operators and return rows to parent
- Control flow starts at root (left-to-right)
- Data flow starts at leaf level (right-to-left)



Operator Cost (1)

- **Cost used to equate to elapsed time in seconds required to run on a specific Microsoft employee's machine (7.0 days)**
- **So really, "cost" today in the context of query plans is a unit-less measure**
  - Cost <> time <> milliseconds
- **Cost is used for relative comparison across plan operators and between plans**

Operator Cost (2)

- **Operator cost = I/O cost + CPU cost**
- **Cost calculation varies by operator**
  - Some have I/O and CPU costs
  - Some have just CPU cost
- **Sub-tree cost = cost of specific operator + descendants**
- **Total cost found in root operator**
- **Estimated costs remain so in “actual” plans**

Operator Memory (1)

- **Each type of operator requires varying amounts of memory in order to perform the associated operation**
- **Some operators require *more* memory because they cache rows**
  - More rows = more memory required
    - SQL Server performs estimates of the required memory and tries to reserve the memory grant prior to execution
    - This is where cardinality estimation is critical (for more on this subject, see "SQL Server: Troubleshooting Query Plan Quality", http://bit.ly/WRwSpD)

Query Memory (2)

- **One area on the "watch list" are heavy memory consuming operators**
  - Hash Match
  - Sort
- **When available memory is insufficient, high memory requiring queries may wait to execute**
- **Under-estimating memory (due to CE issues) can cause spills to tempdb (I/O)**
- **Over-estimating memory can reduce concurrency!**