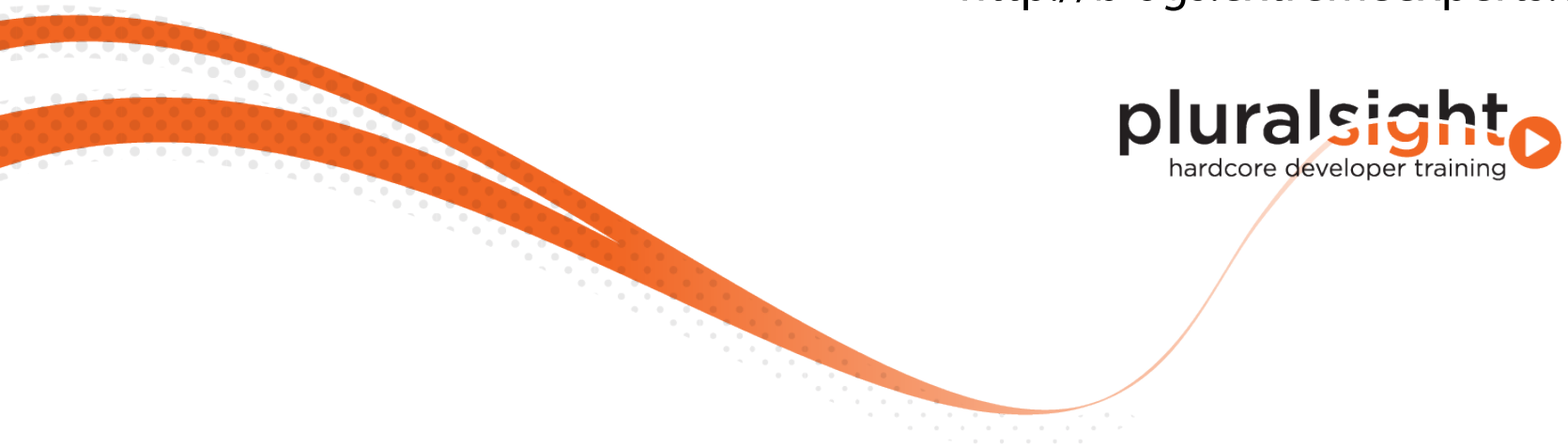
Basics to Execution Plan

Pinal Dave
http://blog.sqlauthority.com

Vinod Kumar M
http://blogs.extremeexperts.com



Execution Plans are like Weather Forecasts

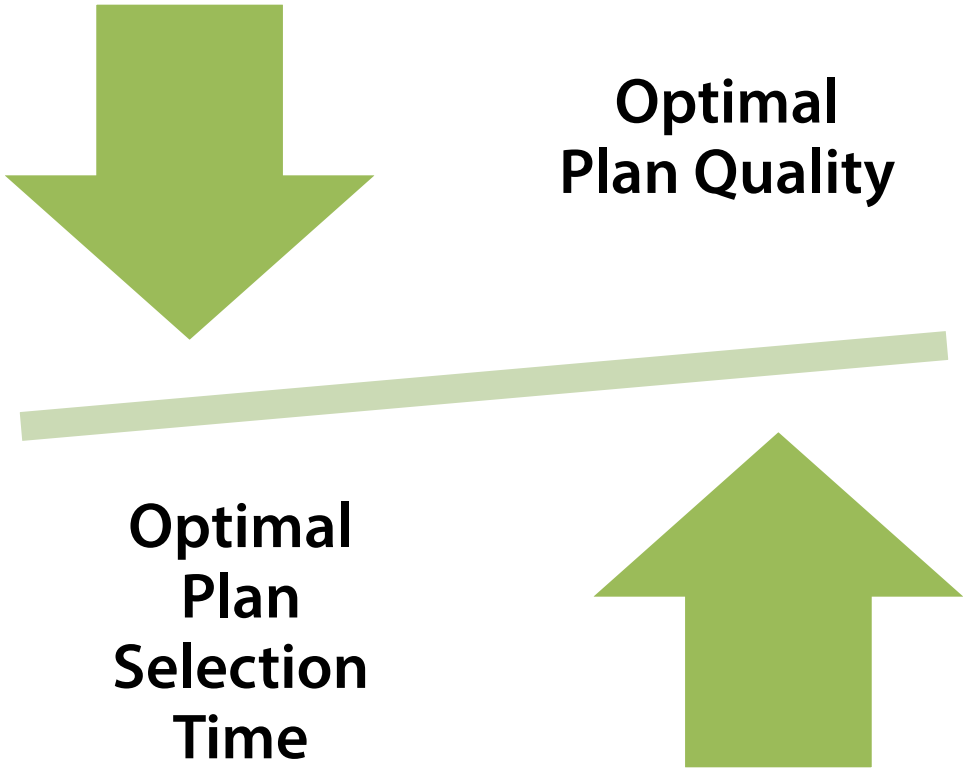
Getting Started

- Query Optimization Fundamentals
- Execution Plan Fundamentals
- Execution Plan Operators
  - Scan
  - Seek
  - Joins

Query Optimization

- Cost Based Rules
  - Generate Multiple Execution Plans
  - Selects Lowest Cost Execution Plans
- Maps Logical Query Operations to Physical Operations
  - Logical (Inner) Join -> Nested Loop Join
  - Sort -> Sort

Finding Optimal Plan



Plan Cost

- Sum of all physical operator in plan
  - CPU
  - IO
  - Memory

Display Options of Execution Plans

- SET STATISTICS TIME ON
- SET STATISTICS IO ON

Types of Execution Plans

- Estimated Execution Plan
- Actual Execution Plan

Display Types of Execution Plans

- Graphic
- Text
- XML

Right Click Options on Operators

- Save Execution Plan As..
- Show Execution Plan XML…
- Zoom In
- Zoom Out
- Zoom Fit
- Properties

Demo

- Query Cost
- Types of Execution Plans
- Execution Plan Properties
- Basic Building Blocks

Execution Plan Operators

- When
- Good
- Bad
- Action Item

80% - 20% Rule

Table Scan

- **When**
  - Table without clustered index is accessed
- **Good or Bad***
  - Can’t decide
- **Action Item**
  - Create Clustered Index

*** 80% - 20% rule**

Clustered Index Scan

- **When**
  - Table with clustered index is accessed
    - Query does not use non clustered index
    - Table does not have non clustered index
- **Good or Bad***
  - Bad unless large data with most columns and rows retrieved
- **Action Item**
  - Evaluate Clustered Index Keys

*** 80% - 20% rule**

Clustered Index Seek

- **When**
  - Table with clustered index is accessed and query locates specific rows in B+ tree
- **Good or Bad***
  - Good
- **Action Item**
  - Evaluate possibility of non-clustered index

*** 80% - 20% rule**

Non-Clustered Index Scan

- **When**
  - Columns part of non-clustered index accessed in query
- **Good or Bad**
  - Bad unless large data with most columns and rows retrieved
- **Action Item**
  - Create more refined non-clustered index

*** 80% - 20% rule**

Non-Clustered Index Seek

- **When**
  - Columns part of non-clustered index accessed in query and rows located in B+ tree
- **Good or Bad***
  - Good
- **Action Item**
  - Further evaluate other operators

*** 80% - 20% rule**

Lookups

- **When**
  - Query Optimizer uses non-clustered index to search few column data and base table for other columns data
- **Good or Bad***
  - Bad
- **Action Item**
  - Included Index or Covered Index

*** 80% - 20% rule**

Seek or Scan

Nested Loops Join

- **When**
  - A nested loops join is particularly effective if the outer input is small and the inner input is pre-indexed and large
- **Good or Bad***
  - Good in small transactions
- **Action Item**
  - Optimizer knows best

*** 80% - 20% rule**

Merge Join

- **When**
  - A merge loops join is particularly effective when both the inputs are sorted on the merge column
- **Good or Bad***
  - Good when query has large result set
  - Bad when inputs are not sorted
- **Action Item**
  - Optimizer knows best

*** 80% - 20% rule**

Hash Join

- **When**
  - A hash join is particularly effective with large data requires many types of set-matching operations
- **Good or Bad***
  - Good when query requires set-matching operations
- **Action Item**
  - Optimizer knows best

*** 80% - 20% rule**

Optimal Join

Optimizer's Choice

Summary

- **Query Optimization Fundamentals**
- **Execution Plan Fundamentals**
- **Execution Plan Operators**
  - Scan
  - Seek
  - Joins

Remember: SQL Server Optimizer usually opts for most efficient execution plan.

Remember: 80%-20% Rule. There are always special cases.