How Intelligent Query Processing Improves T-SQL Performance

Andy Yun
Principal Solutions Engineer

SentryOne

Andy Yun

Principal Solutions Engineer

- SQL Server DBA & DB Developer
- Chicago Suburban User Group Chapter Leader
- Chicago SQL Association – Director-at-Large
- Working with SQL Server since 2001
- Speaking since Early 2014
- Microsoft MVP (2017-2018)

@SQLBek - ayun@sentryone.com

https://blogs.sentryone.com/andyyun/

https://www.github.com/sqlbek/

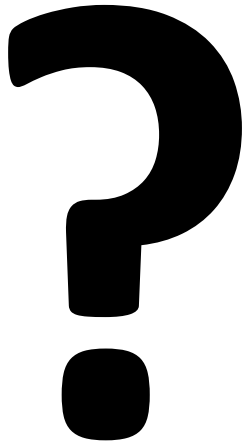
SentryOne™

Is Your T-SQL Codebase…

- Something you inherited?
- Written by a vendor?
- Still mission-critical but no longer in active development?
- Older than my socks?

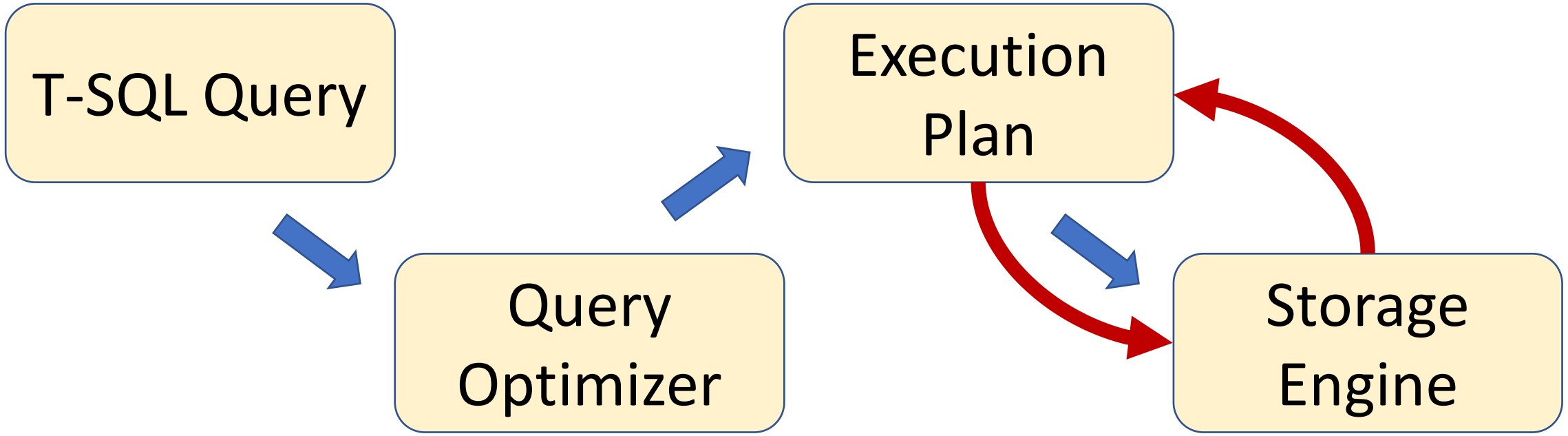
Wouldn’t It Be Nice If…

What Version Are You Running On



History of Query Processing

AQP - Rewriting History



2019 - Intelligent Query Processing

- Table Variable Deferred Compliation
- Inlining Scalar User Defined Functions
- Batch Mode for Rowstore
  - Adaptive Joins *
  - Memory Grant Feedback *

** Introduced in 2017*

Table Variable Deferred Compliation

SentryOne™

Why I've Always Disliked Table Variables

- Variable vs a Table
- Variable + Table = Table Variable
- Query Optimizer = What's in here?

DEMO:
Table Variable Deferred Compliation

SentryOne™

Inlined Scalar Valued Functions

SentryOne™

Why I've Always Disliked Scalar UDFs?

```
SELECT
    ColumnOne,
    ColumnTwo,
    udf_DRY_TS
FROM dbo.Scala
```

D.R.Y.

VS

R.B.A.R.

hree)

Horribly

What

- Tran
- Has

A scalar T-SQL UDF can be inline if all of the following conditions are true:

- The UDF is written using the following constructs:
  - `DECLARE`, `SET`: Variable declaration and assignments.
  - `SELECT`: SQL query with single/multiple variable assignments[1].
  - `IF`/`ELSE`: Branching with arbitrary levels of nesting.
  - `RETURN`: Single or multiple return statements.
  - `UDF`: Nested/recursive function calls[2].
  - Others: Relational operations such as `EXISTS`, `ISNULL`.
- The UDF does not invoke any intrinsic function that is either time-dependent (such as `GETDATE()`) or has side effects[3] (such as `NEWSEQUENTIALID()`).
- The UDF uses the `EXECUTE AS CALLER` clause (the default behavior if the `EXECUTE AS` clause is not specified).
- The UDF does not reference table variables or table-valued parameters.
- The query invoking a scalar UDF does not reference a scalar UDF call in its `GROUP BY` clause.
- The query invoking a scalar UDF in its select list with `DISTINCT` clause does not have `ORDER BY` clause.
- The UDF is not used in `ORDER BY` clause.
- The UDF is not natively compiled (interop is supported).
- The UDF is not used in a computed column or a check constraint definition.
- The UDF does not reference user-defined types.
- There are no signatures added to the UDF.
- The UDF is not a partition function.
- The UDF does not contain references to Common Table Expressions (CTEs)

entryOne™

DEMO:
Inlining Scalar User Defined Functions

SentryOne™

Batch Mode For Rowstore

SentryOne™

What is Batch Mode?

- T-SQL is Set Based…
- Or is it?
- Used to only be available for Columnstore Indexes
- Now available for Rowstore in SQL Server 2019

What are Adaptive Joins?

- Nested Loop Joins vs Hash Joins
- Only available for Queries that leverage Batch Mode

What is Memory Grant Feedback?

- Need buffer pool workspace to work
- Do we always guess the right amount we need?

DEMO:
Batch Mode, Adaptive Joins, & Memory Grant Feedback

SentryOne™

Parting Thoughts

Be Cautious, Especially of Parameter Sniffing/Caching Issues…

But Be Not Afraid!

SQL Server is all about trade-offs

Utilize Database Scoped Configurations

Thank you!

https://github.com/SQLBek

Andy Yun | @SQLBek
ayun@sentryone.com | SQLBek@gmail.com
http://blogs.sentryone.com/andyyun/
http://sqlbek.wordpress.com

Want To Learn More About SentryOne?
Book a demo with me!
http://www.sentryone.com/BookAndy/

SentryOne