

Transact SQL Performance Tips

Milos Radivojevic bwin.party 

SQLBits XI, 4th May 2013

twitter #SQLBITS

Milos Radivojevic



MCT, Database Developer, bwin.party Vienna, Austria

- MCT, MCITP, MCTS, MCP
- Software development background
- Relational and DWH/BI Area
- Workshop: SQL Server for Application Developers
- Co-founder: SQL Pass Austria
- Conference Speaker: SQLBits, SQLU Summit, SQL Saturday, SQL Pass Austria

Microsoft
CERTIFIED

Trainer

Microsoft
CERTIFIED
IT Professional



Contact:

Milos.Radivojevic@bwinparty.com

www.bwinparty.com



@MilosSQL

Agenda

- Local Variables
- Functions in the WHERE Clause
- Data Type Conversion
- SQL Server 2012 Tips
- APPLY vs. JOIN
- Database Constraints and Performance
- Other Tips

Query Execution

- Transact SQL is a declarative language
- SQL Server generates an execution plan (cost-based optimization)
- What can affect the query execution?
 - Which tables are involved in the query
 - Are there indexes available on the tables
 - Cardinality estimation by reading table statistics
 - **How the query is written**

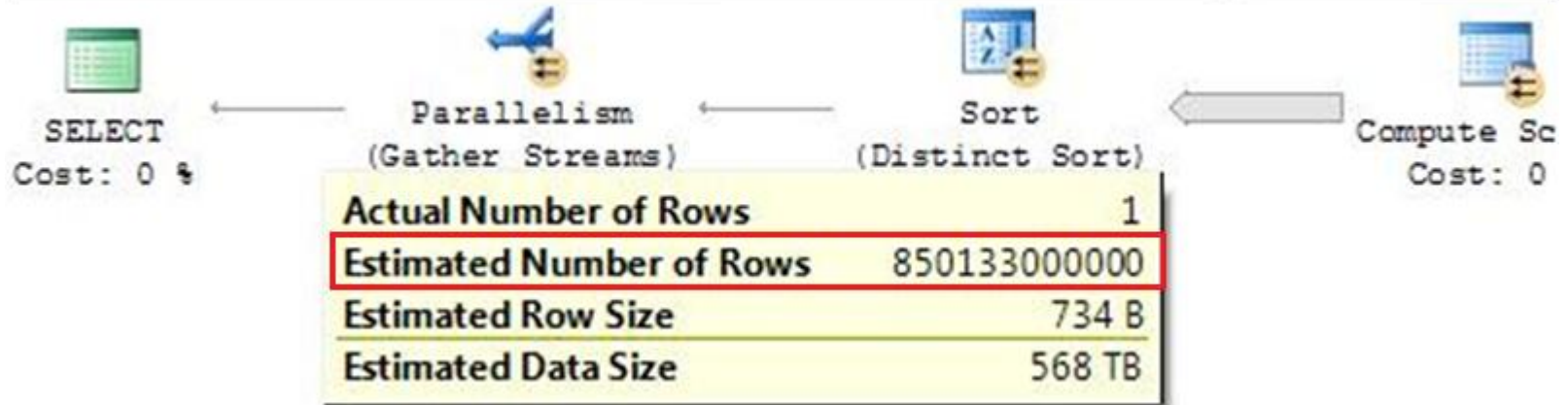
Local Variables

- **Tip: Understand how local variables affect query execution**
 - When local variables are used the SQL Server optimizer cannot generate the optimal execution plan because the variable value is not known at the compile time
- **Tip: Use `OPTION (RECOMPILE)` to get a good execution plan**
 - Do not underestimate discrepancy in the estimation

Local Variables

- **Tip: Do not underestimate discrepancy in the cardinality estimation!**
- **Otherwise you could see something like this:**

Query 2: Query cost (relative to the batch): 100%
select distinct sa.ACCOUNT_ID, sc.NAME, r.ROLLEN_ID, r.ROLLEN



Functions in the WHERE Clause

- **Tip: Avoid functions in the WHERE clause with table columns as arguments!**
 - An index will not be used at all or it will be used inefficiently
- **Tip: Avoid arithmetic operations against table columns in the WHERE clause!**

Data Type Conversion

- **Tip: Choose the right data type!**
- Otherwise:
 - Conversion overhead
 - String Conversion costs can be significant (a “non-Unicode” value will be converted to a “Unicode” value)
 - Estimation problem (for instance with the LIKE Operator)

SQL Server 2012 Tips

- **Tip: Use Window Aggregate Functions!**
 - Great SQL Server 2012 Improvements
 - Running Totals
- **Tip: Use Window Offset Functions!**
 - LEAD, LAG, FIRST_VALUE, LAST_VALUE
 - Current vs. Previous etc.
- **Tip: Use Window Functions Whenever You Can!**
 - In SQL Server 2005, 2008/R2 too!

Database Constraints

- **Tip: Database Constraints help SQL Server to make a better execution plan.**
 - The main purpose is data integrity, but Unique Constraints, Check Constraints and Foreign Keys help SQL Server optimizer to make the right decision about query execution

Other Tips

- **Tip: SELECT Only Required Columns!**
 - Only required information should be requested. It sounds trivial, but there are lot of examples with unnecessary SELECT * or ORDER BY statements
- **Tip: Use ORDER BY only if it was explicitly required!**
 - An overhead can be significant when Sort Operator is involved

Other Tips

- **Tip: Use UNION ALL when you know that sets are not overlapped and when duplicates are allowed!**
- **Tip: EXISTS vs. COUNT(*) perform the same. Use what you are more comfortable with!**
- **The same tip for IN vs. EXISTS!**
 - From the version 2005 perform same

Code Examples

<http://bit.ly/126uWOo>

Coming up...

Speaker	Title	Room
Leonard Lobel	Geospatial Data Types in SQL Server 2012	Theatre
Mark Stacey	Analysis Services for the DBA	Exhibition B
Dennis Lloyd Jr	Thwarting Database Defects	Suite 3
Dmytro Andriychenko	Kerberos for SQL Server and SharePoint the easy way	Suite 1
Justin Langford	Using Windows Azure to provide SQL Server Disaster Recovery	Suite 2
Dave Ballantyne	Estimation, Statistics and “Oh My!”	Suite 4



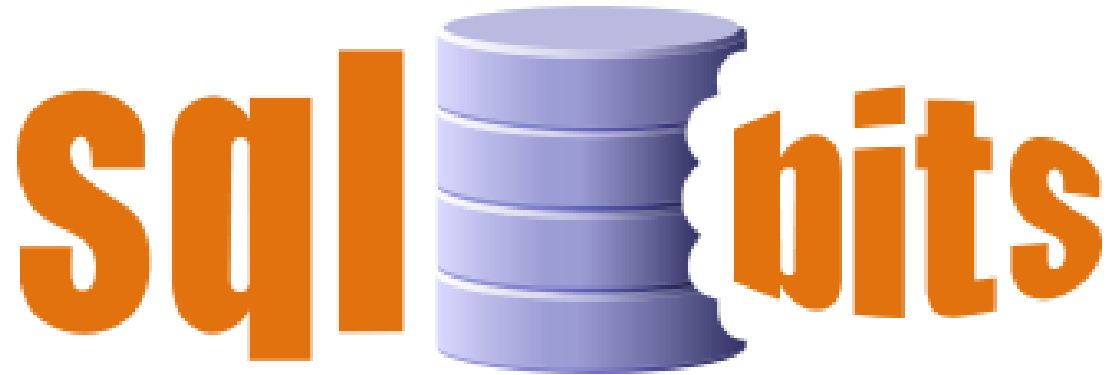


Microsoft®
SQL Server® 2012

Enjoy SQLBits 11

twitter #SQLBITS





Thank You!

