Applying Common Performance Practices

David Berry
@davidcberry13 buildingbettersoftware.blogspot.com

What We'll Cover

- Using parameterized SQL
- Stored procedures vs. application SQL
- Commit frequency and performance
- Object Relational Mappers and database performance
- N+1 selects problem

Parameterized SQL Advantages

DEV

Understand how an application works

TEST

Log detailed results from a load test

PROD

Track statements that exceed thresholds

Parameterized SQL Advantages

Protects against SQL injection attacks

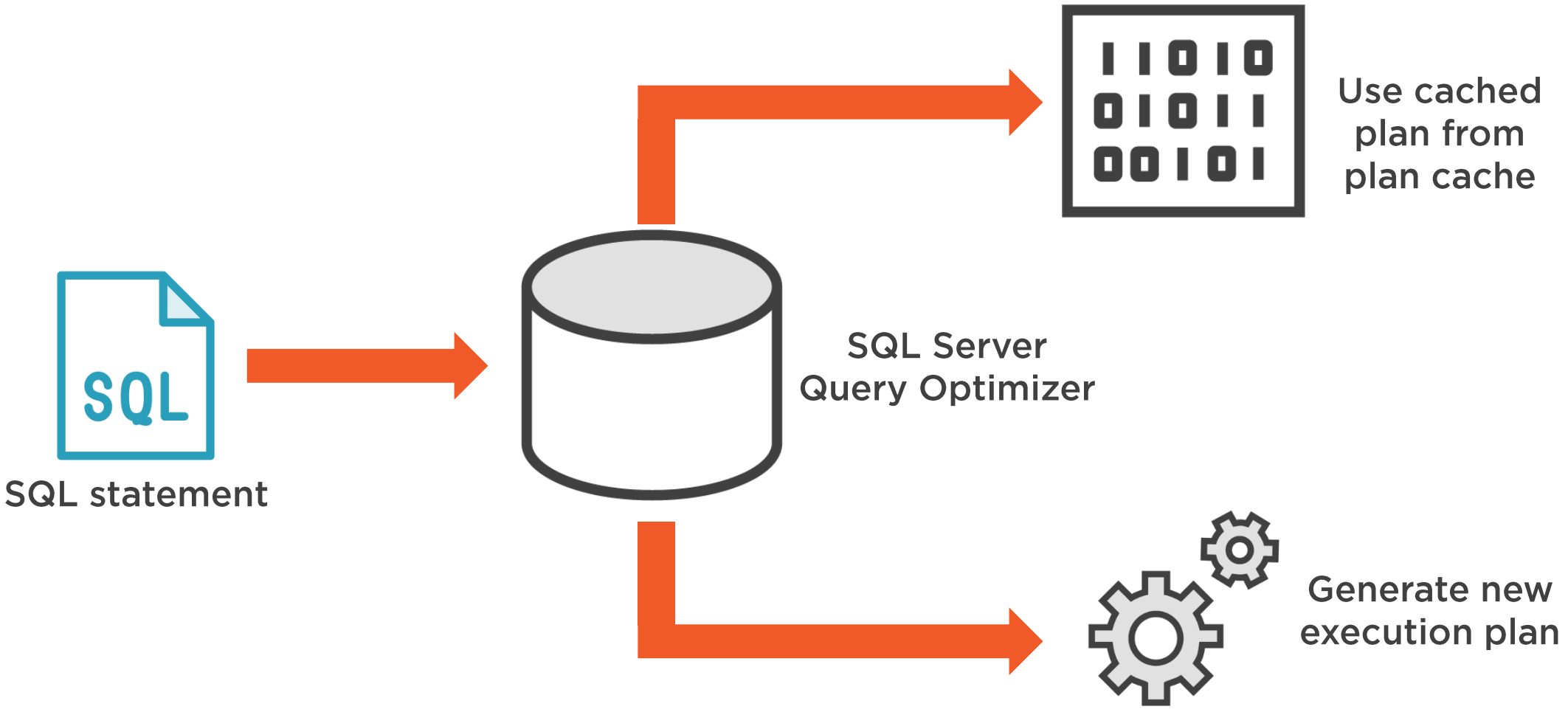
Improves application performance and scalability

Parameterized SQL Summary

ORMs like Entity Framework and Hibernate will automatically parameterize the SQL they generate

Be sure to parameterize SQL in data access code you write

Determining an Execution Plan for a Statement



Are stored procedures faster than including SQL in your application code?

Dynamic vs. Parameterized SQL Performance

	Dynamic SQL	Parameterized SQL
Elapsed Time	8515 ms	963 ms
CPU Time	6827 ms	79 ms
Plan Cache Memory	79 MB	104 KB
Notes	Each SQL statement unique	Plan can be reused for each execution

Similarities Between Stored Procedures and Parameterized SQL

```
CREATE PROCEDURE GetStudentInfoByName(
    @firstName   VARCHAR(40),
    @lastName    VARCHAR(40)
)
AS
    SELECT StudentId, FirstName, LastName, Email, Telephone
        FROM Students
        WHERE FirstName = @firstName
            AND LastName = @lastName;
GO
```

SQL statement with parameters

Stored Procedure Performance Comparison

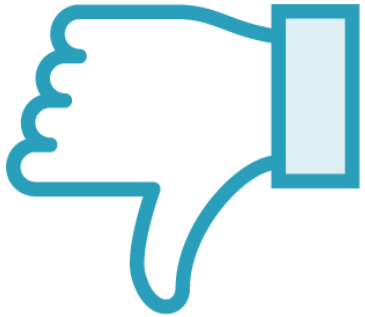
	Dynamic SQL	Parameterized SQL	Stored Procedure
Elapsed Time	8515 ms	963 ms	979 ms
CPU Time	6827 ms	79 ms	151 ms
Plan Cache Memory	79 MB	104 KB	104 KB
Notes	Each SQL statement unique	Plan can be reused for each execution	Similar performance to parameterized SQL

SQL Server drivers have auto-commit turned on by default

Commit as often as your business transaction dictates

Object Relational Mappers

Increases developer productivity



More difficult to tell what is happening in the database

ORM code will be converted into SQL statements

Tracing can be used to get SQL text and analyze further

N+1 Selects Problem

Records are being loaded individually because they are being lazy loaded

Eager loading is a better choice in this case

Watch for N+1 selects problem any time you are loading child objects

Use tracing to understand exactly what is happening

Module Summary

Parameterized SQL

Stored procedures and parameterized SQL

Commit frequency and performance

ORMs create SQL

N+1 selects problem

Thank you

You now know the fundamental tools and concepts of SQL Server performance

Get some hands on experience with an application and database you own