The SQL Server Experts
Discover. Optimise. Mentor.

Designing Quality SQL Server Solutions

Gavin Payne
Principal Architect

gavin@coeo.com
@GavinPayneUK

Gavin Payne

Enterprise and solution IT architect:

Platforms and infrastructures

New and existing environments

Full solution lifecycle focus:

Business technology strategy

Business & technical requirements

Solution design

Solution implementation

IASA
An Association for all IT Architects

Microsoft CERTIFIED
Architect
SQL Server® 2008

Microsoft CERTIFIED
Master
SQL Server® 2008

Agenda

- Defining and measuring solution quality
- Security and Manageability with SQL Server
- Performance and Scalability with SQL Server

Intended Audience

- Anyone who uses any part of SQL Server
- In fact, anyone who creates or manages IT solutions

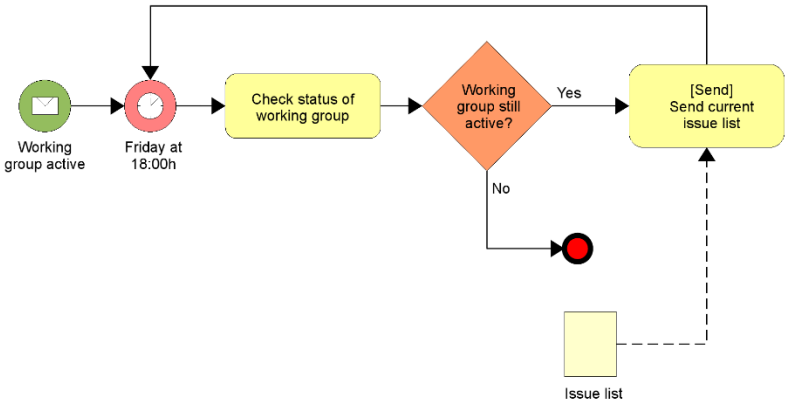
Defining and measuring quality

What is quality?

- Robustness
- Assurance
- Grade of excellence
- Superior
- Suitability

What is solution quality?

- How well a solution functions
  - *Business logic*
  - *Error handling*
  - *Integration*



- How well a solution operates
  - *Manageability*
  - *Scalability*
  - *Security*

What is solution quality?

- How well a solution functions

"Business requirements"

Functional requirements

- How well a solution operates

"Technical requirements"

Non-functional requirements

Non-functional requirements

- Also known as quality attributes
- Judge a system's ***operation*** rather than describe its ***behaviour***
- Cover every aspect of a solution
- Ensure complete lifecycle value:

  Design
  Implementation
  Operation

Pre-defined quality attributes

- Different definitions exist: *architectural, technical, vendor*

Availability	Recoverability
Extensibility	Reliability
Integrity	Scalability
Interoperability	Security
Manageability	Serviceability
Performance	

Source: TOGAF 9.1

How can you use quality attributes?

- Solution design standards
- Solution development

- Acceptance criteria
- Critical success factors

- In-life operations
- Change review processes

Quality	Measurement
Availability	99.9%
Performance	<1 sec per click
Security Auditing	Yes

Quality attributes trade-offs

- Having high quality standards is expensive
- Trade-offs between cost and capability are needed
- Consider the what-if situations

What if we can't support Android?

Security and Manageability

Security

- Protecting a solution from un-authorised access
- What are the potential threats today?

  External hackers
  System administrators
  Un-authorised end users

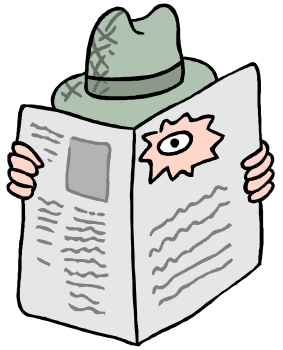
- A good permissions model is insufficient these days

Common Security Issues

- Developers and vendors typically don't know the minimum permissions required

- Users routinely need access to large data sets

  For example PowerPivot



- System administrators can normally see everything

Security with SQL Server

- Database engine is full of security features, that few use

Object permissions model

Encryption – column data, data files

Auditing – access failures, successful access

Separation of duties – Stop DBAs reading data

- Why don't they get implemented very often?

Few can describe their value

The SQL Server Experts
Discover. Optimise. Mentor.

Demo

Delivering Security

Manageability

- Monitoring and controlling a solution
- How does the business monitor the solution's performance?

  It's probably not a Performance Monitor counter
- How easily can you make changes?

  If it's too complex…….

Manageability with SQL Server

- SQL Server has good error handling

  How do you capture or find its error messages?

- How do you capture your own error messages?

  TRY….CATCH….THROW into a logging table

- How do you monitor for unusual activity that completed successfully? *Bad user activity may not cause errors*

Security and Manageability summary

- These can be complimentary quality attributes
- An easy to operate system is a well liked system

- Few people seem to implement much security

  SQL Server has lots of features

  Even if its just to monitor access

Performance and Scalability

Performance

- Performance isn't the same as scalability

- Functionality delivered
- In an acceptable time
- For a single user

- The first quality attribute we consider
- But, how do we measure performance?

Performance with SQL Server

- Query design and tuning
  - *Table indexing and statistics*
  - *Amount of data being processed*
  - *Balance of processing between app and data tiers*
- Data design
  - *Duplication of data*
  - *Ease of finding data*

Scalability

- Maintaining acceptable performance as utilisation increases

  How many concurrent users can we support?

  How do data volumes affect performance?

- Scalability on day 1 should be perfect

  What about day 180 and day 5000?

- Scalability requires more than load testing

The SQL Server Experts
Discover. Optimise. Mentor.

Scalability Testing Overview

Scalability Testing

- Test objective

What type of clustered index allows the fastest inserts?

Regular clustered

GUID

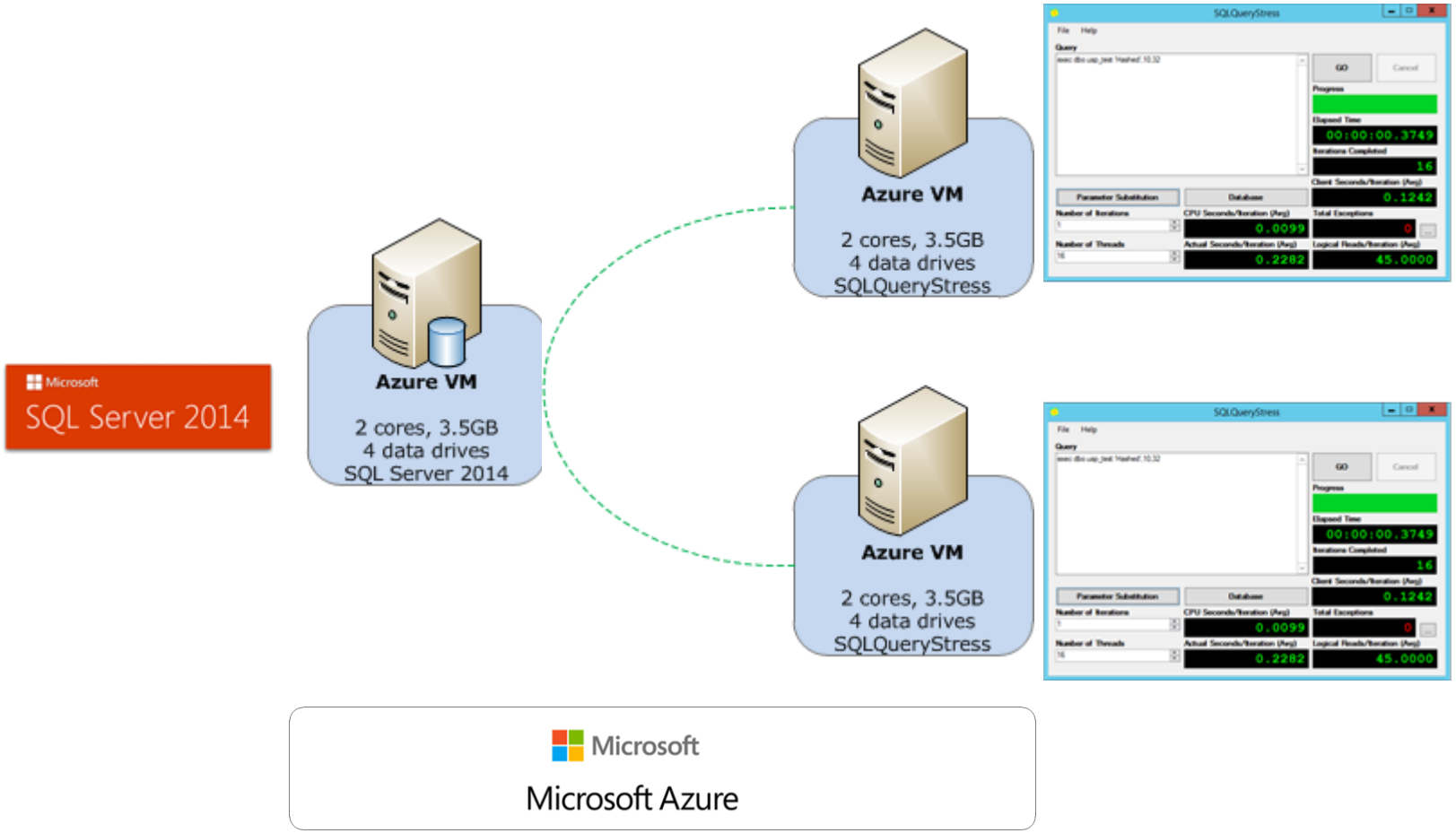
GUID with 50% fill-factor

Clustered ColumnStore index

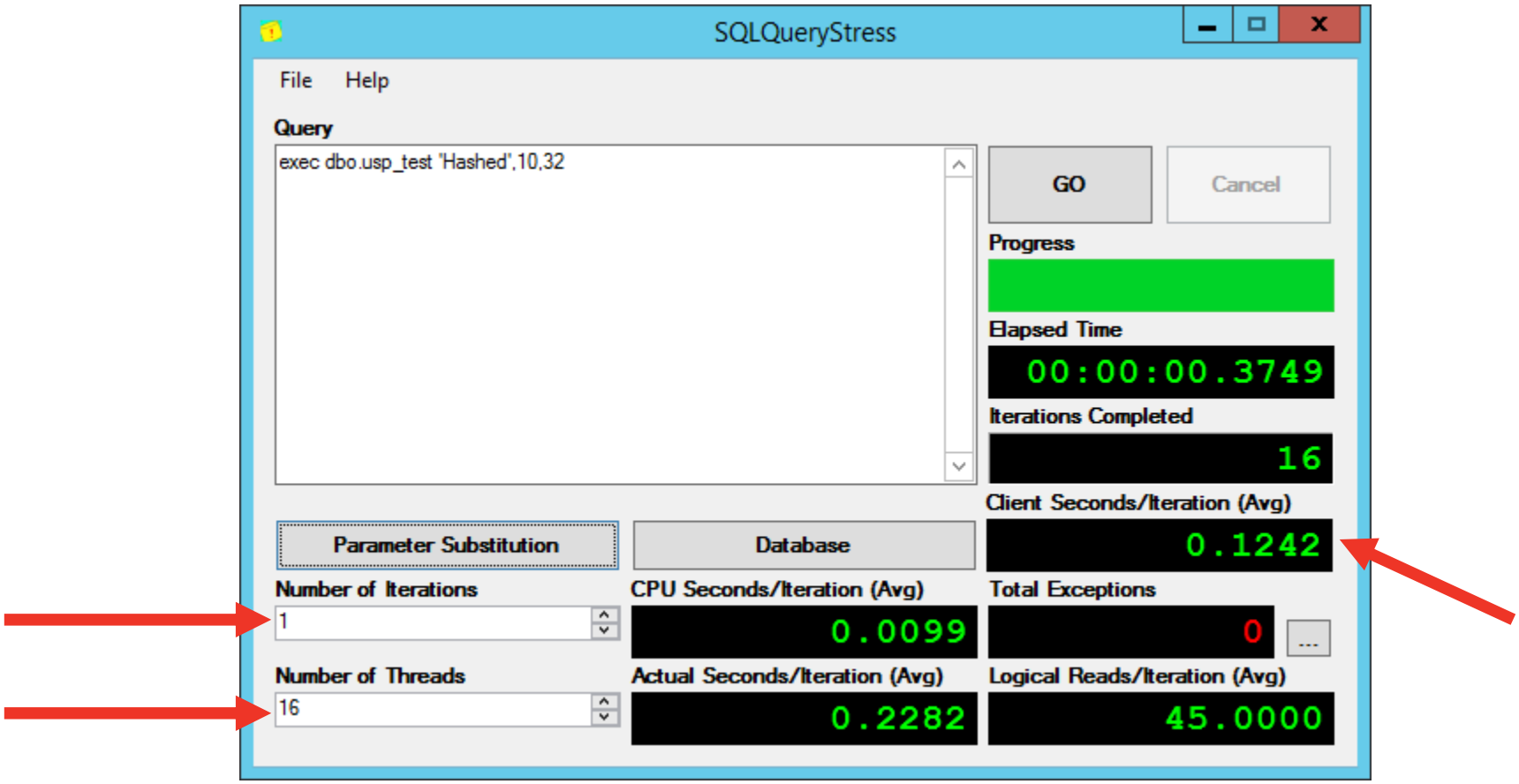
Hash partitioned index

And what effect does Delayed Durability have?

Scalability Testing



Scalability Testing



Scalability Testing

- Test process

  Stored procedure that performed 1000 inserts
  Calculated and logged duration of each insert

  Enough concurrent activity to show internal resource waits
  Not enough to overwhelm CPU resources

Scalability Testing

- Test results

Index Type	Regular Logging (ms per insert)	Delayed Durability (ms per insert)
Regular clustered index	20.5	16.4
GUID	13.9	5.0
GUID with fill-factor 50	14.6	3.6
Clustered ColumnStore	25.1	18.9
Hash partitioning	13.7	6.5

96 concurrent threads

Scalability with SQL Server

- Resource scalability
  - *Add more – Memory and CPUs*
  - *Control utilisation – Resource governor for CPU and IO*
  - *Work harder – Data compression*

- Database scalability
  - *Data type widths*
  - *Locking and page latch contention*
  - *Scale-out data model*

Performance and Scalability summary

- Performance is the most obvious quality attribute

  Easiest to define, measure and test

- Scalability is what catches people out

  It's not just about the hardware

  Nor the number of users

Session Summary

- Functional requirements define **business value**
- Non-functional requirements provide **in-life success**
- Also known as quality attributes and can be **measured**
- How can you use them to **improve** your solution's quality?

Questions?

The SQL Server Experts
Discover. Optimise. Mentor.