prologika

Microsoft Partner
Gold Data Analytics
Silver Data Platform

Can Your Data Analytics Solution Scale?

Teo Lachev, Prologika
teo.lachev@prologika.com

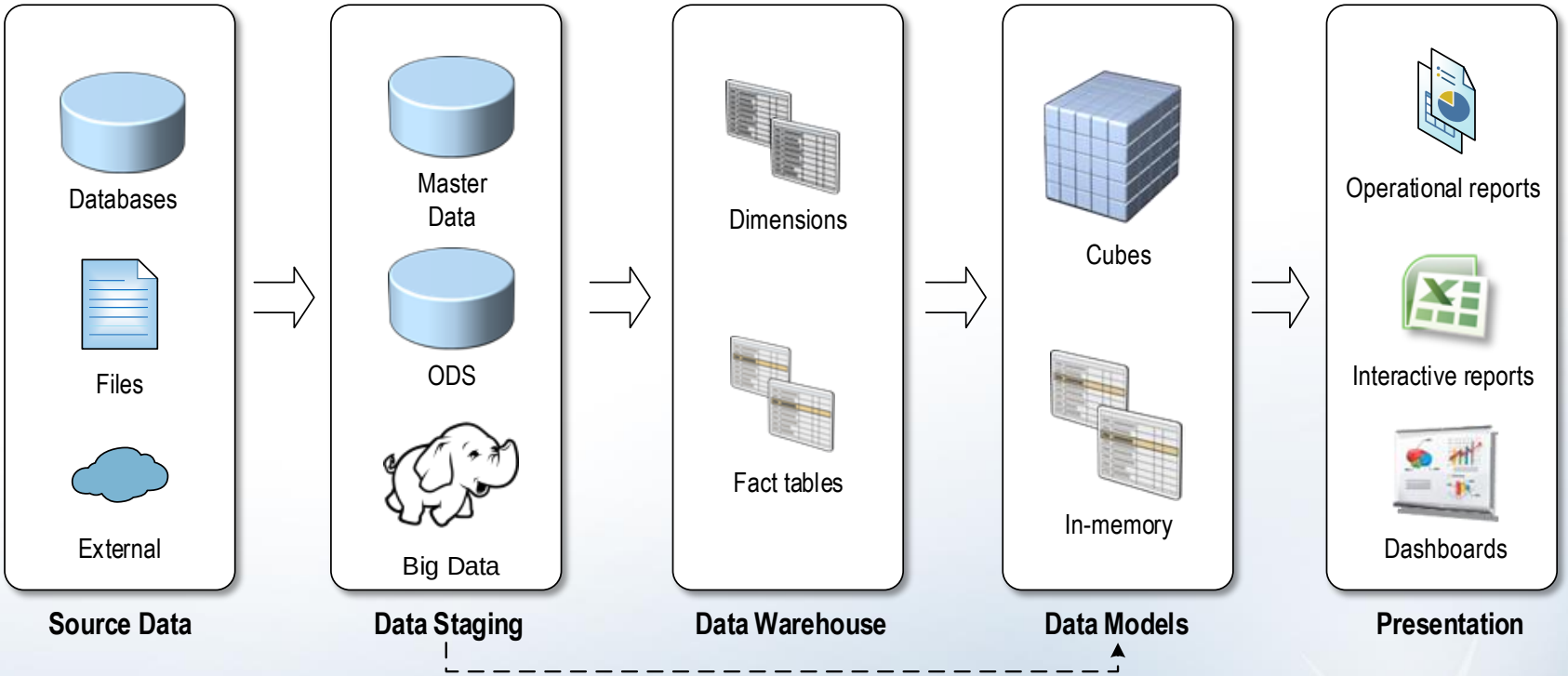
Agenda

- Share best practices and tips
  - Performance and scalability
  - Focus on organizational data analytics
- Layers
  - Database design
  - ETL
  - Reports
  - Data models
- Downloadable slides
  http://prologika.com/CS/media/

About me

- Consultant, author, and mentor with focus on BI
- Owner of Prologika – BI consulting and training company based in Atlanta (www.prologika.com)
- Microsoft Gold Partner in Data Analytics
- Microsoft SQL Server MVP for 11 consecutive years
- Founder and leader of Atlanta BI group (atlantabi.sqlpass.org)

"Classic" Organizational Data Analytics Architecture



SQL Server Can Scale

Real-life projects

- Web analytics cube
  - 25 Billion rows
  - 2.5 TB cube
- Enterprise data warehouse
  - 5 TB compressed data
  - Two fact tables over 1 billion rows
- Gift card processing
  - Heavy OLTP loads
  - Thousands of transactions per second
- General scalability guidelines
  - < 10-15 TB – Symmetric MultiProcessing SMP single SQL Server
  - > 15 TB – Massively Parallel Processing (MPP) Parallel Data Warehouse, Hadoop

Know Your Environment

- I/O
  - `BACKUP DATABASE [ContosoRetailDW] TO DISK='NUL';`
  - Or use tools such as IOMeter or CrystalMark
  - I/O should be ideally above 500 MB/sec
  - Or use winsat, e.g. C:\winsat disk –drive c

Network

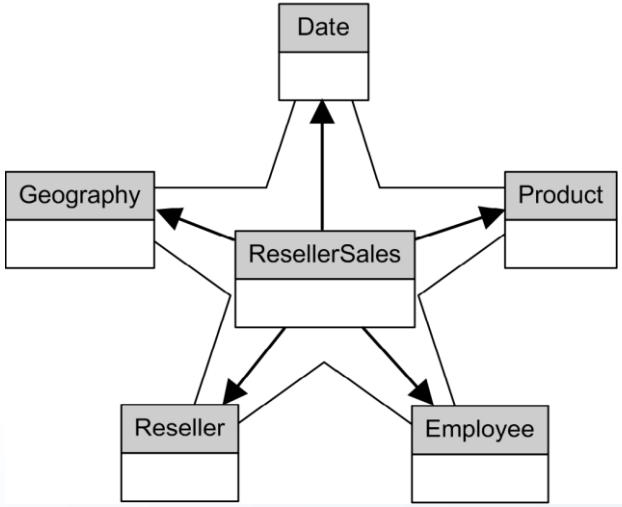
- select * from <some narrow fact table> (consider discarding query results)
- num rows/sec = row count/execution time (sec)
- Aim for > 100K rows/sec

- CPU
  - How many cores are actually licensed?
  - Avoid changing MAXDOP and hyperthreading
- Virtualization
  - Disk pass-through enabled
  - Dedicated resources

Database Design

Schema Design

- OLAP/Data analytics
  - Denormalize
  - Star schema is your best friend
  - Goal is not a single dataset though
    - You need conformant dimensions
    - Avoid loading too much data



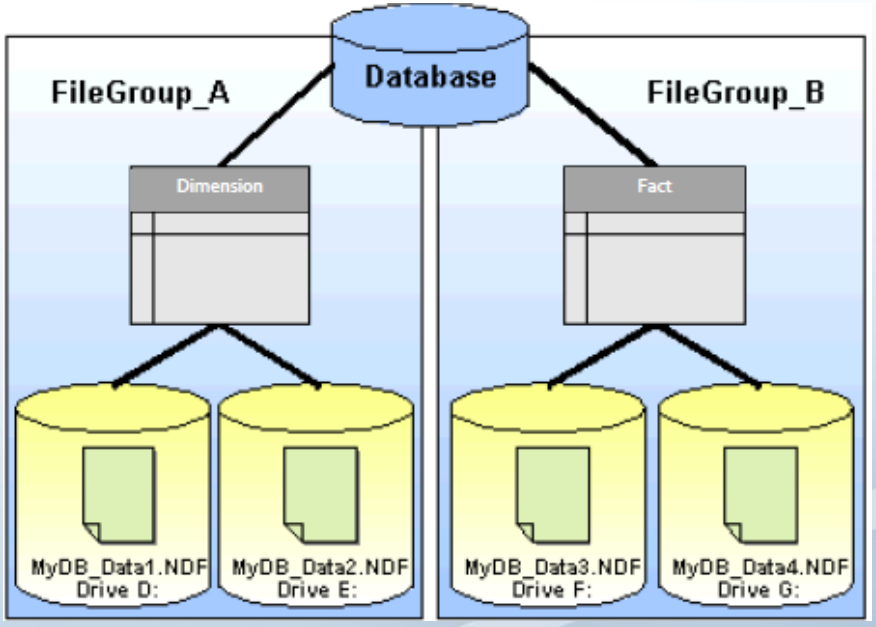
- OLTP
  - Normalize
  - 3NF is your best friend

For OLTP applications, avoid triggers (cause implicit transactions)

Optimize Physical Design

Storage

- For DW/DM, set database recovery to Simple
- Spread I/O to multiple files
  - File group for each large table
  - Files on different drives
  - Avoid using Primary file group



Optimize Physical Design

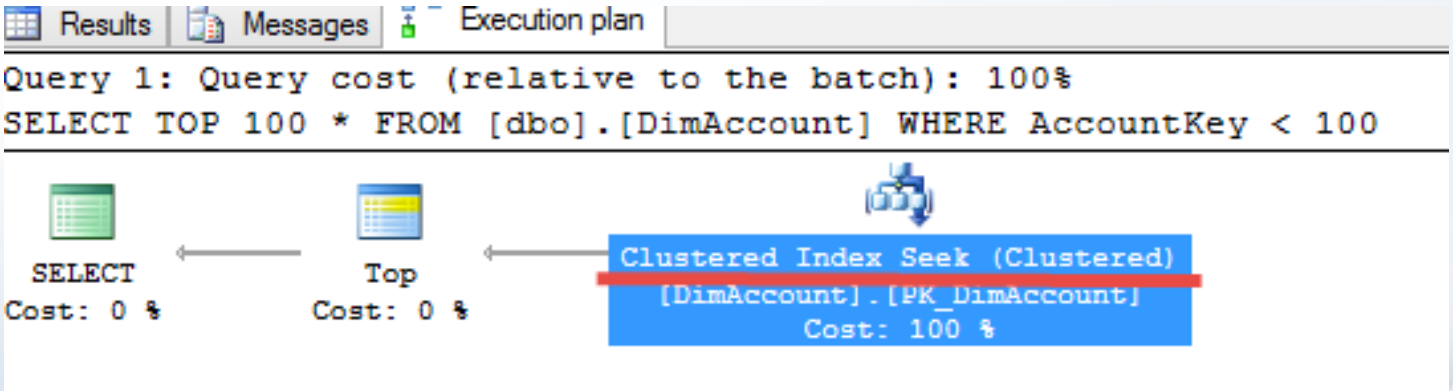
Partitions and Compression (EE features)

- Partition large tables (above 50 Gb or 100 million rows)
  - Parallel table loading
  - Partition switching
  - Better manageability
  - Partition elimination when querying data
- Compression
  - Consider page compression with large tables
  - 50-80% saving in disk space
  - To estimate storage savings:
  - Use SSMS Data Compression Wizard
  - sp_estimate_data_compression_savings stored procedure

Resources:
"Partitioned Table and Index Strategies Using SQL Server 2008" whitepaper by Ron Talmage
"Data Compression: Strategy, Capacity Planning and Best Practices" whitepaper by Sanjay Mishra

Indexing Strategies

- Clustered indexes perform best
  - Unique clustered index on target table for MERGE
  - SELECT queries from large tables
  - WHERE **clause** for SSAS partition slice
- You want to see Cluster Index Seek

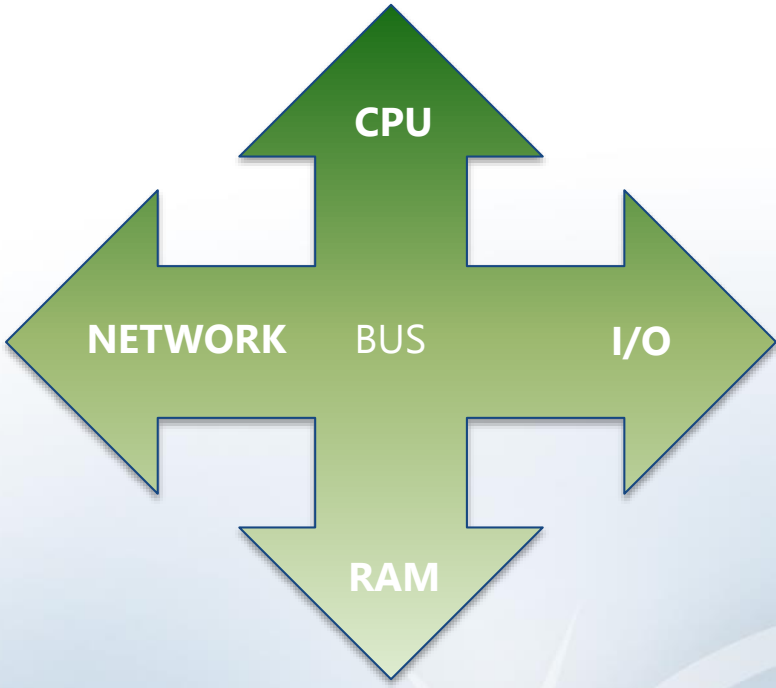


Indexing Strategies
Continued

- Eliminate read locks
  - View or single-statement hints - WITH (NOLOCK)
  - Stored procedures or free-text statements
    SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED
- Maintain indexes
  - Defragment
  - Update statistics
- Consider columnstore indexes
  - With massive aggregate queries
  - Ensure queries use batch mode

Finding Performance Bottlenecks

- Every system has a saturation point
- Use Windows performance counters to discover bottlenecks
- "Big Four" bottlenecks
  - CPU (Processor: % Processor Time)
  - Memory (Memory: Available MBytes)
  - HDD (LogicalDisk: Avg Disk Sec\Read) (5-10ms for disks, <1ms for SSD or Avg. Disk Reads/Sec (<85%)
  - Network (Network Interface: Bytes Sent/sec, Bytes Received/sec)
- Priorities
  1. Optimize
  2. Parallelize
  3. Upgrade hardware



Resources:
"Troubleshooting Performance Problems in SQL Server 2008" whitepaper by Microsoft

Detailed Analysis

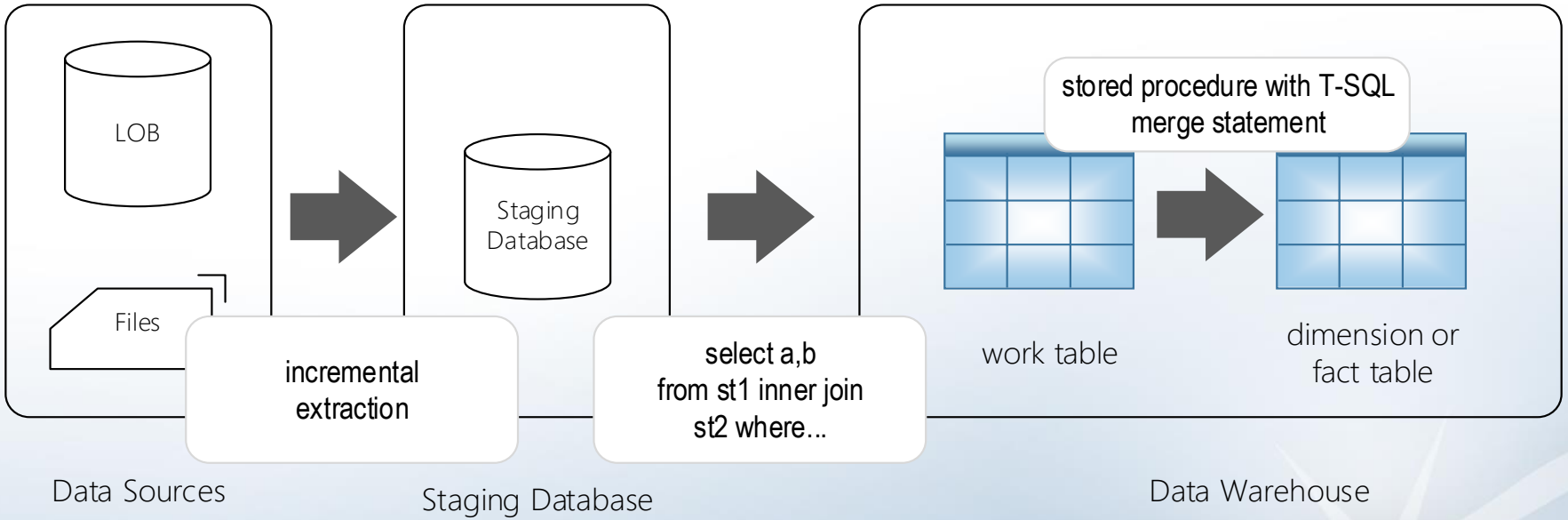
Dynamic Management Views

- Purpose
  - Monitor health
  - Diagnose problems
  - Tune performance
- Examples
  - Discover SQL Server waits
    http://msdn.microsoft.com/en-us/library/ms179984.aspx
  - Discover missing indexes

ETL

Consider ELT Design Pattern

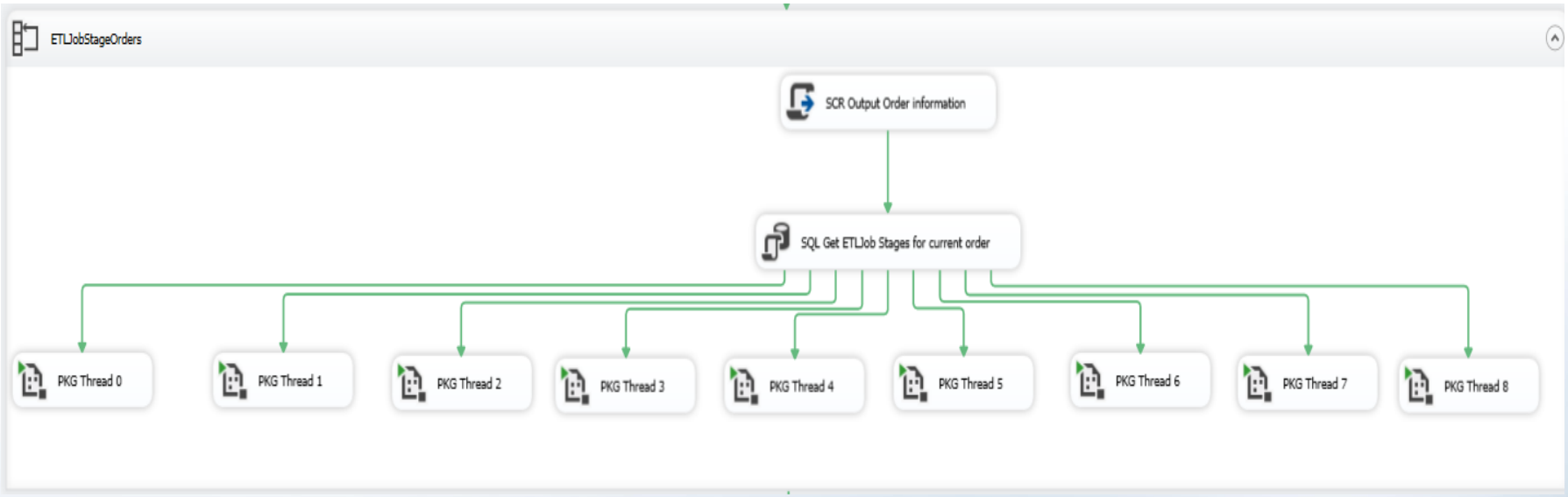
- More efficient than SSIS data flow
- More flexible than SSIS lookups
- Easier to maintain
- Easy to automate



If the staging and DW databases are on the same server you don't even need a work table

Use Parallel Task Execution

- Use parallel package execution whenever possible
- Invest in an SSIS framework that supports parallelism



Task-level Analysis

- SSIS 2012 project deployment mode
  - Basic level should be enough (or Performance)
  - Statistics logged in [catalog].[executable_statistics]

```
SELECT execution_id,status,
CASE  WHEN [status] = 1 THEN 'created'
   WHEN [status] = 2 THEN 'running'
   WHEN [status] = 3 THEN 'canceled'
   WHEN [status] = 4 THEN 'failed'
   WHEN [status] = 5 THEN 'pending'
   WHEN [status] = 6 THEN 'ended unexpectedly'
   WHEN [status] = 7 THEN 'succeeded'
   WHEN [status] = 8 THEN 'stopping'
   WHEN [status] = 9 THEN 'completed'
END AS [status_text], DATEDIFF(ss,start_time,end_time) DurationInSeconds FROM  catalog.executions e
```

  - Not using project deployment mode?
    - Enable Logging to SQL Server
    - Select OnPreExecute, OnPostExecute, and OnProgress events

```
SELECT      executionid Execution,
          PackageName = ( SELECT  TOP 1 source FROM dbo.sysssislog S WHERE S.executionid = L.executionid  AND S.[event] = 'Packag
          PackageStep = l.source,
          Runtime_min            = DateDiff(minute, MIN(L.starttime), MAX(L.endtime)),
           Runtime_hr            = DateDiff(hour, MIN(L.starttime), MAX(L.endtime)),
          StartTime     =  MIN(L.starttime),
          EndTime            = MAX(L.endtime),
   FinishDate     =  CAST(MAX(L.endtime)AS DATE)
FROM  dbo.sysssislog L  WHERE L.[event]         != 'PackageStart'
GROUP BY executionid, L.source
having DateDiff(minute, MIN(L.starttime), MAX(L.endtime))>0
```

Stored Procedure Analysis

- Use Extended Events (XEvents)
- Create event session

```
CREATE EVENT SESSION [query_performance] ON SERVER
ADD EVENT sqlserver.sql_statement_completed(
ACTION(sqlserver.query_hash,sqlserver.client_hostname,sqlserver.database_id,
sqlserver.plan_handle, sqlserver.nt_username,
sqlserver.client_pid,
sqlserver.client_app_name)
WHERE sqlserver.database_id >= 6 AND sqlserver.query_hash <> 0 AND
[sqlserver].[like_i_sql_unicode_string]([sqlserver].[session_nt_user],N'''%teo%''')
```

- Start session
- Execute queries and stored procedures
- Stop session
- Analyze data

Mapping Extended Events to SQL Trace
https://www.sqlskills.com/blogs/jonathan/an-xevent-a-day-20-of-31-mapping-extended-events-to-sql-trace/

Index Considerations

- Drop non-clustered indexes and constraints
  - With massive changes (10% or more)
  - Enables minimally-logged load
- Columnstore indexes
  - Watch out for parallel index rebuild with partitioned tables (currently, under open support case with MS)
  - Index rebuild should max out all licensed cores

"The Data Loading Performance Guide" whitepaper by Microsoft

Reports

Find Slow Reports

- Examine ExecutionLog3 view
- Execution times (ms)
  - TimeDataRetrieval – time spent in getting data
  - TimeProcessing – time processing reports
  - TimeRendering – time exporting reports
- Other useful info
  - RowCount – number of dataset rows
  - Status – rsSuccess is report executed successfully
  - ByteCount – report payload
  - Source – Live, Cache, Snapshot, History
  - Parameters – report parameters and values
  - AdditonalInfo – additional execution info http://blogs.msdn.com/b/robertbruckner/archive/2009/01/05/executionlog2-view.aspx

Can Our Reports Scale?

- Step 1: Establish performance goal
- Step 2: Prepare load tests
- Step 3: Run and analyze load tests
- Do we meet the goal?
  - Yes – we are done
  - No – identify and eliminate performance bottlenecks, go to Step 3

Example:

- Peak report usage (from ExecutionLog) – November 22th, 9 AM – 10 AM
- 200 reports executed by 20 distinct users
- 200/3,600 = 0.05 reports/sec

Estimate future loads

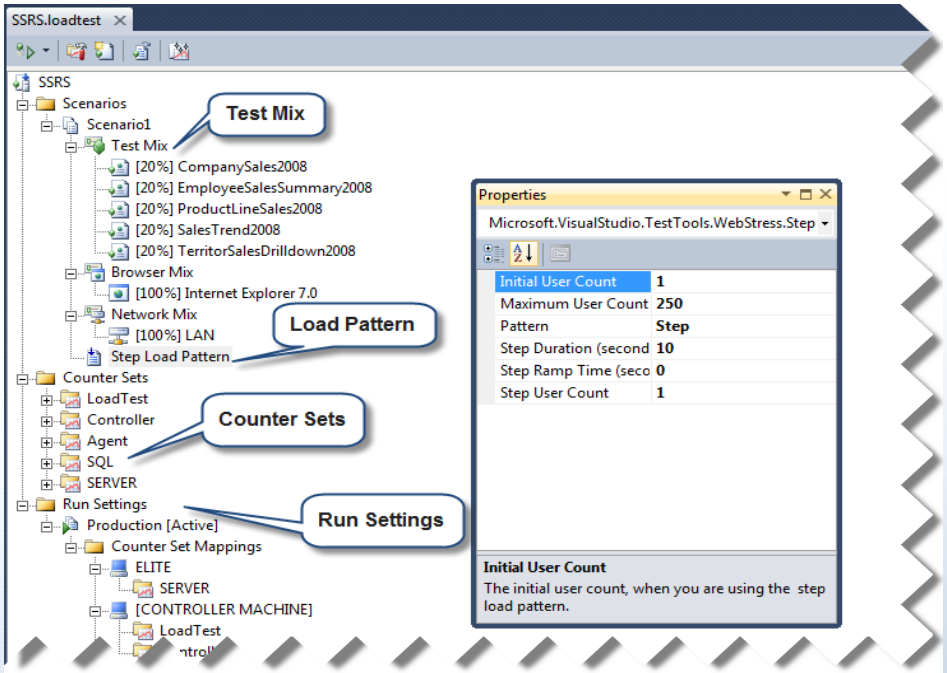
- Scale to 500 users – x25 increase (500/20)
- 0.05 x 25 = 1.25 reports/sec

Derive performance goal

- Let's double 2 x 1.25 = <u>2.5 reports/sec</u>

Load Testing Reports

- Use Visual Studio (Ultimate or Test edition) Create New Load Test wizard
  - Load pattern – constant or stepped
  - Test mix – a collection of web or/and unit tests
  - Counter sets – CPU, memory, disk utilization
  - Run settings – counters, warm-up time, test duration or iterations



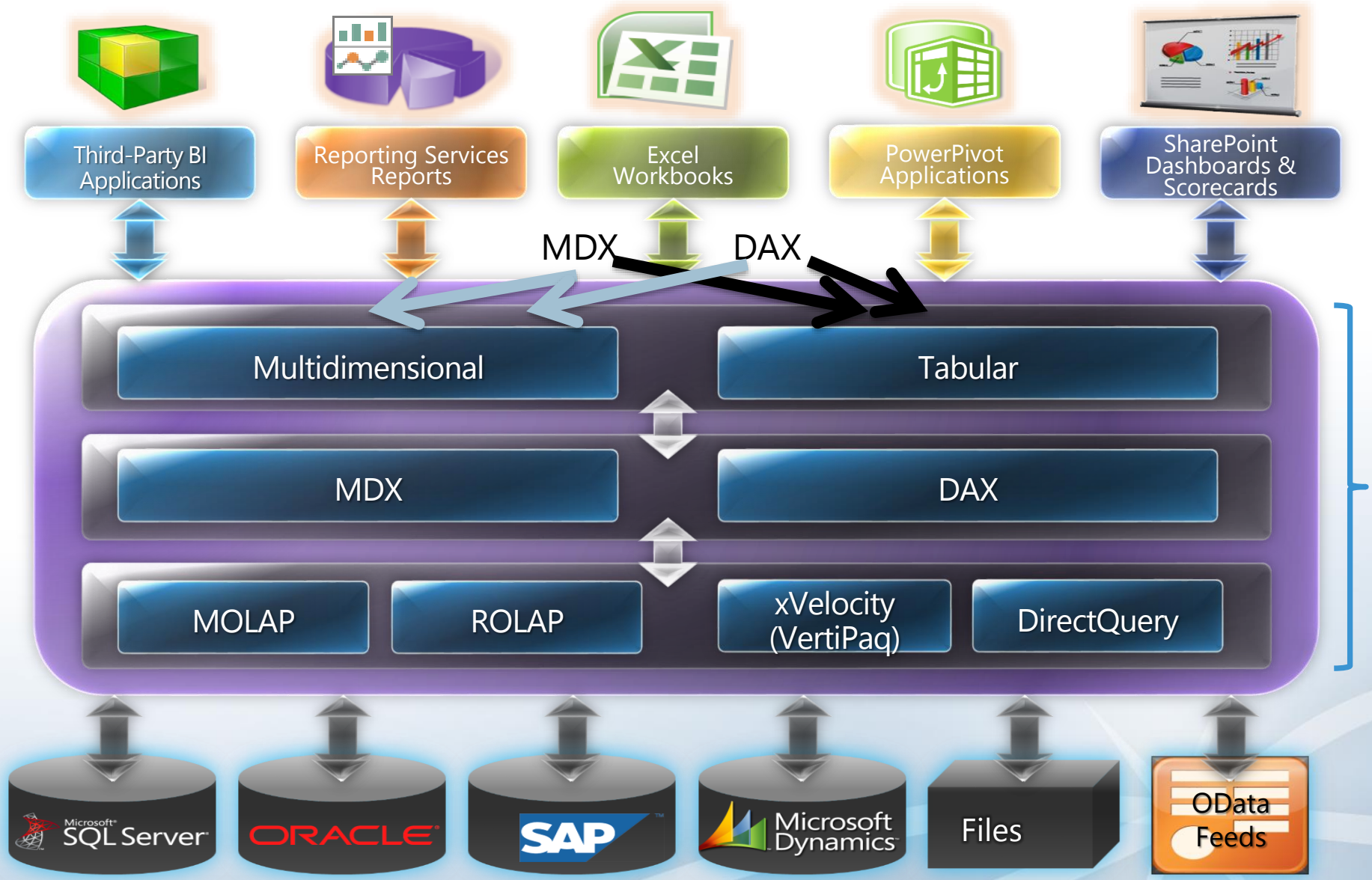
Resources and tips:

"DBI330: Can Your BI Solution Scale?" video presentation by Teo Lachev

Increase the MaxActiveReqForOneUser setting in rsreportserver.config to a large value, e.g. 999999

Data Models

Data Models


Third-Party BI Applications
Reporting Services Reports
Excel Workbooks
PowerPivot Applications
SharePoint Dashboards & Scorecards
MDX
DAX
Multidimensional
Tabular
MDX
DAX
MOLAP
ROLAP
xVelocity (VertiPaq)
DirectQuery
Microsoft SQL Server
ORACLE
SAP
Microsoft Dynamics
Files
OData Feeds

Choose Model Wisely

- Decision checkpoints
  - Data volumes
  - Complexity
- Scenarios for considering Multidimensional
  - Data warehousing
  - Large data volumes
  - Complex models
- Scenarios for considering Tabular
  - Promoting PowerPivot models to organizational models
  - Rapid development for simple models
  - Transactional reporting? (be careful)
- Special considerations
  - DistinctCount – favors Tabular
  - Flexibility and brute scans – favors Tabular
  - Maturity and features (M2M, semi-additive) – favors Multidimensional
  - Data doesn't fit in memory – favors Multidimensional

Top SSAS Best Practices

- Partition large measure groups (tables) e.g. above 100 mil rows
  - Incremental model processing
  - Partition elimination (Multidimensional)
- Avoid large dimension tables, e.g. invoice details
  - Move text attributes to fact tables
  - Use report actions in Multidimensional
- Write efficient queries
  - Queries should execute within milliseconds
  - Use SQL Profiler to get SE and FE time breakdown
  - Write efficient queries – example below http://prologika.com/CS/blogs/blog/archive/2014/06/16/optimizing-arbitrary-shaped-sets.aspx

Load Testing Analysis Services

- Scale out should be last step (really)
- A single server should be more than enough
- When in doubt, load test SSAS

Resources and tips:
"DBI330: Can Your BI Solution Scale?" video presentation by Teo Lachev

Summary

- When it comes to performance
  - Ignorance is not a bliss
  - Monitor and optimize
- For more information
  - teo.lachev@prologika.com
  - http://www.prologika.com
  - Twitter: @tlachev