Power BI Performance
...Tips and Techniques
Optimize your data lifecycle.
Rachael Martino
Principal Consultant
rmartino@pragmaticworks.com
@RMartinoBoston
About Me

• SQL Server and Oracle developer and IT Manager since SQL Server 2000
• Focused on BI and building a data culture of excellence
• Boston area resident
• Ravelry fan and avid knitter
Agenda

Impact of poor performance

Performance Tips and Techniques

Demonstration
Power BI is fast

Or, why worry about performance?
Power BI Tools

SharePoint
Share Results
Data Refresh

Power View
Data Exploration
Interactive Mapping

Power Pivot
Relationships
Hierarchies
KPIs

Power Map
3-D Mapping
Animated Data
Video Output

Power BI Sites
Share Results
Data Refresh
Q&A

Mobile Power BI
Anytime
Anywhere
Any Device
Architecture

- **xVelocity in-memory analytics engine**
  - Columnar storage
  - Compression
  - In-memory cache

- "Microsoft’s family of in-memory and memory-optimized data management technologies"

Power BI leverages PowerPivot and PowerView (and Power Query)

In-memory, columnar database and formula engine are fast

“Now is 3 seconds”

Performance impacts

Slow Processing on data loads

Long waits during Design, especially:
• Calculated column
• Relationships

Visualization:
• Slow slicers
Large file size of pbix file:
• Not necessarily indicator of bad performance
• Sudden changes

Memory usage
• Direct impact on performance

Screenshot of my local drive, showing improvements in file size as I resolved data issues.
Performance impact demo

Behaviors affect:
1) Data Load
2) Design
3) Visualizations
Tips and Techniques

Let’s solve this...
Tip #1: Tall, narrow tables are faster*

- Corollary Tip #1a: remove any unused fields
- Tables must efficiently compress columns for speed
- Remove relationship ID’s not in use – these may have high cardinality and are unnecessary
- Remove all fields not used for analysis

*The exception:
In the case of tables with 10’s of millions of rows, the 1M-row partitions or “Chunking” may interfere with efficient compression rates
Tip #2: Integers are faster than strings

- Strings, stored in hash table, require two queries to get a single value.
- Hash table uses less memory unless there is high cardinality, then the hash table becomes overhead.
- Strings used as ID’s can use unreasonable amounts of memory and slow performance.
Tip #3: Slicers use multiple queries

- Slicers issue two queries each:
  - The first to get the list
  - The second query to check which rows of the pivot tables are related
- Cross-filtering slicers cause those same two queries to be executed for multiple sets of slicers.
- High cardinality slicers from large tables make poor user experience (too many options) and are slow
Tip #4: Understand DAX functions

• Understand formula engine interaction with the xVelocity engine for your DAX
• The FILTER statement must check every row individually (no bulk scans)
  - [link](http://www.powerpivotpro.com/2014/02/speed-another-reason-to-trim-calendar-tables/)
• MIN will have to scan the entire table to find the answer
  - [link](http://www.powerpivotblog.nl/tune-your-powerpivot-dax-query-dont-use-the-entire-table-in-a-filter-and-replace-sumx-if-possible/)
Tips #5 & 6:

5. Remove unnecessary rows in Calendars:
Extra rows in reference tables cause unnecessary attempts at computation; rows not shown on tables or slicers have been computed against the fact data
http://tinylizard.com/power-pivot-performance-gotchas/

6. Remove unnecessary precision or split granularity values to reduce cardinality
For example: split datetime into Date and Time
http://tinylizard.com/power-pivot-performance-gotchas/
Sometime Tip #7: Caution with calculations

• The formula engine is I/O intensive and runs on one thread only, if processing performance is problematic, move simple calculations to the database
• Once processed calculated columns are static values in the data store
• Measures are calculated during query execution

• Caution: In design every change to the data model requires a recalc of all calculated columns
Technique #1: Check your memory usage

- File size is a rough estimate of performance, but not 100% accurate.

- Measure relative memory usage of each table and field using Kasper de Jonge’s Power Pivot Memory Usage tool in Excel
Technique #2: Check your DAX

- Slow measures and calculations can cause big problems at design time and in visualizations
- Lookup DAX

- You may want to test and evaluate DAX using DAX Studio – I have not used this tool.

https://www.sqlbi.com/tools/dax-studio/
So, is this better?

....Let’s see
Performance improvement demo

Using our tips and techniques
Conclusions
Power BI is fast and will perform

- Performance is very good in Power BI and PowerPivot models
- Large is relative depending on efficiency of data
- Think about your data model and calculations
References (not on slides)

• Excel and Power Pivot by Bill Jelen and Pragmatic Works Tabular and Power Pivot On Demand
• Power Pivot Pro
• Brad Gall’s Power BI v2 and Beyond, January 19, 2016
• http://www.powerpivotpro.com/2015/08/so-your-detailedflat-pivot-is-slow-and-doesnt-sort-properly-try-text-measures/