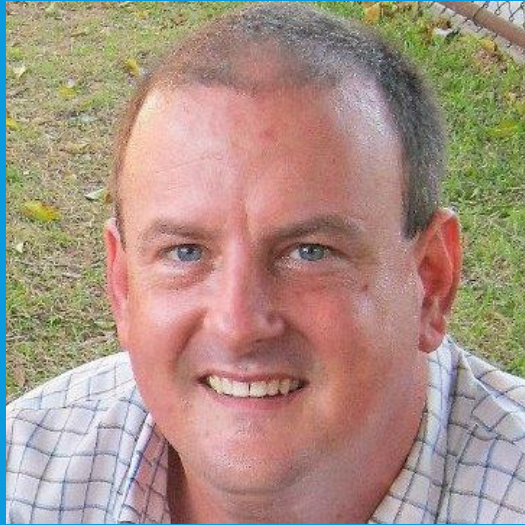# ROLLING DATABASE SNAPSHOTS

David Cobb

sql@davidcobb.net

Daveslog.com

Presenter

David Cobb

SQL Trainer / Consultant

MCT, MCSE Data Platform for SQL 2012

sql@davidcobb.net

daveslog.com

Past Roles:
Tech Support, Network Admin, Web Developer, DBA
Current:
Systems Architect for CheckAlt.com

Training SQL Since: 2002

Favorite Techs of the Moment:
PowerShell & Azure

THE PROBLEM WE'RE TRYING TO SOLVE

Many organizations need a recent version of the data for reporting, with minimal impact on production.

We know reporting from production OLTP servers can impair performance, but often the benefit of using the most recent data outweighs the cost and risk of reporting from production.

Let's look at some alternatives.

CURRENT REPORTING DATA ALTERNATIVES FOR SQL SREVER

- Log Shipping
- Replication
- AlwaysOn
- Database Mirroring

NOTE! While there may be 3rd party solutions for this problem, I investigated native solutions only.

LOG SHIPPING

- Pros:
  - (Relatively) easy to set up and administer

- Cons:
  - 15 min delay typical, can get far behind production
  - In flight queries aborted during restore step

This is the most accessible solution for both distributing reporting data, and providing a low cost disaster recovery option.

If you don't have a budget and need a disaster recovery solution, start here.

REPLICATION

- Pros:
  - Can be very fast
  - Can replicate select tables
  - Transactional & Merge options
- Cons:
  - Overhead in database performance
  - Setup and administration cost
  - Database changes can break it

ALWAYS ON

- Pros:
  - Multiple readable secondaries can distribute read workload
  - Works with groups of databases (Availability Groups)
- Cons:
  - **I DON'T HAVE SQL SERVER 2012/2014/2016** ☺
  - Need Windows Clustering
  - Higher administrative burden

*If you **do** have SQL 2012/2014/2016 Enterprise Edition, this is your best solution!*

DATABASE MIRRORING

- Pros:
  - Works in SQL 2005 and up
  - Easy to configure
  - Great for disaster recovery/failover
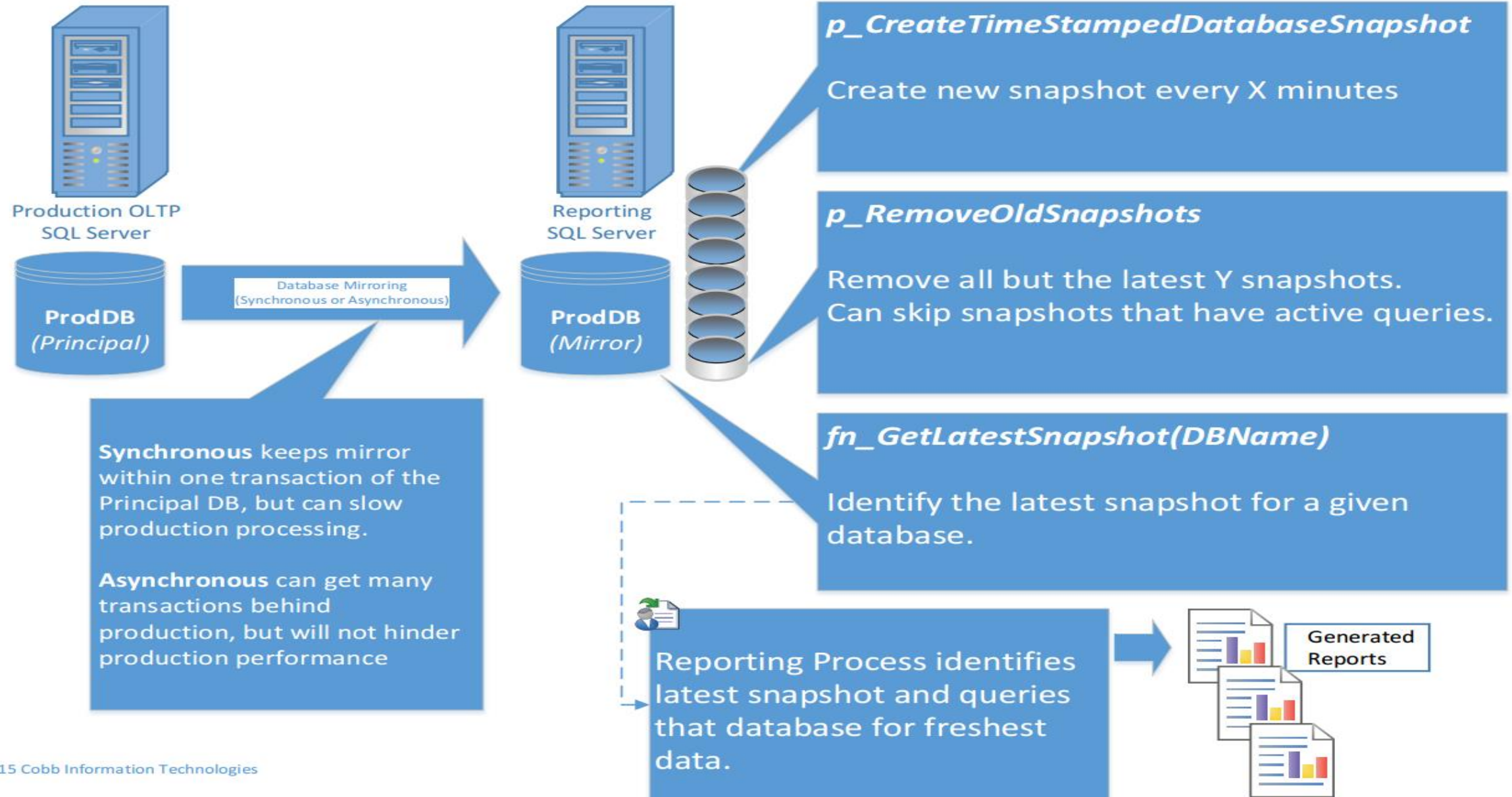
- Cons:
  - Both SQL Servers must be same version and edition, or it's not supported.
  - *Mirrored databases are unreadable unless you fail over to them!*
    (UNLESS YOU USE DATABASE SNAPSHOTS ☺)

Database mirroring is primarily useful for disaster recovery.

The solution I'm proposing today combines it with database snapshots, which makes it useful for reporting as well.

Rolling Database Snapshots – Concept Diagram

Production OLTP
SQL Server

ProdDB
(Principal)

Database Mirroring
(Synchronous or Asynchronous)

Reporting
SQL Server

ProdDB
(Mirror)

Synchronous keeps mirror within one transaction of the Principal DB, but can slow production processing.

Asynchronous can get many transactions behind production, but will not hinder production performance

p_CreateTimeStampedDatabaseSnapshot

Create new snapshot every X minutes

p_RemoveOldSnapshots

Remove all but the latest Y snapshots.
Can skip snapshots that have active queries.

fn_GetLatestSnapshot(DBName)

Identify the latest snapshot for a given database.

Reporting Process identifies latest snapshot and queries that database for freshest data.

Generated Reports

DEMO

FAQ

- What about overhead of snapshots? What's the performance impact?
  - See http://bit.ly/db-snapshot-perf

  - Recommended to limit snapshot use in high volume OLTP environment, but very suitable for separate reporting server.

- Can I have a copy of your code?
  - I've started using github, the code is available at:
  - https://github.com/dave-007/MSSQL-Rolling-DB-Snapshots

- Slides and link to scripts will be on SQLSaturday site and daveslog.com

QUESTIONS?

- SPEAKERS CRAVE FEEDBACK!
- sql@davidcobb.net
- Please fill out evals!