Strategic Sponsor

Microsoft

Gold Sponsors

PYRAMID ANALYTICS

KRAJOWY REJESTR DŁUGÓW www.krd.pl

devart Database Interaction Solutions

Silver Sponsors

Profisee®

SQL EXPERT.pl

CREDIT SUISSE

Technical Partners

Microsoft

PTI POLSKIE TOWARZYSTWO INFORMATYCZNE ODDZIAŁ GÓRNOŚLĄSKI

Academic Partners

Wyższa Szkoła Zarządzania i Bankowości w Krakowie

Wyższe Szkoły Bankowe

Media Partners

GEEKCLUB.pl wss.pl

ERP24.pl

4programmers.net

IT professional

IN FACT PR

itnews24

virtual-it.pl

IT WORKS POLSKA

VirtualStudy

MAGAZYN IT PL

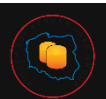
Made in Wro Do IT here!

ERP-view.pl Portal rozwiązań IT w biznesie

Introducing SQLProxy

Maciej Pilecki

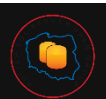
Microsoft Certified Master



About me...

- Microsoft Certified Master in SQL Server
- I’ve been a SQL Server consultant for quite a while...
- Working on large-scale SQL Server deployments worldwide
  - Large-scale is not the same as big-data
  - It’s the workload that matters most
- I’ve seen my share of scalability problems

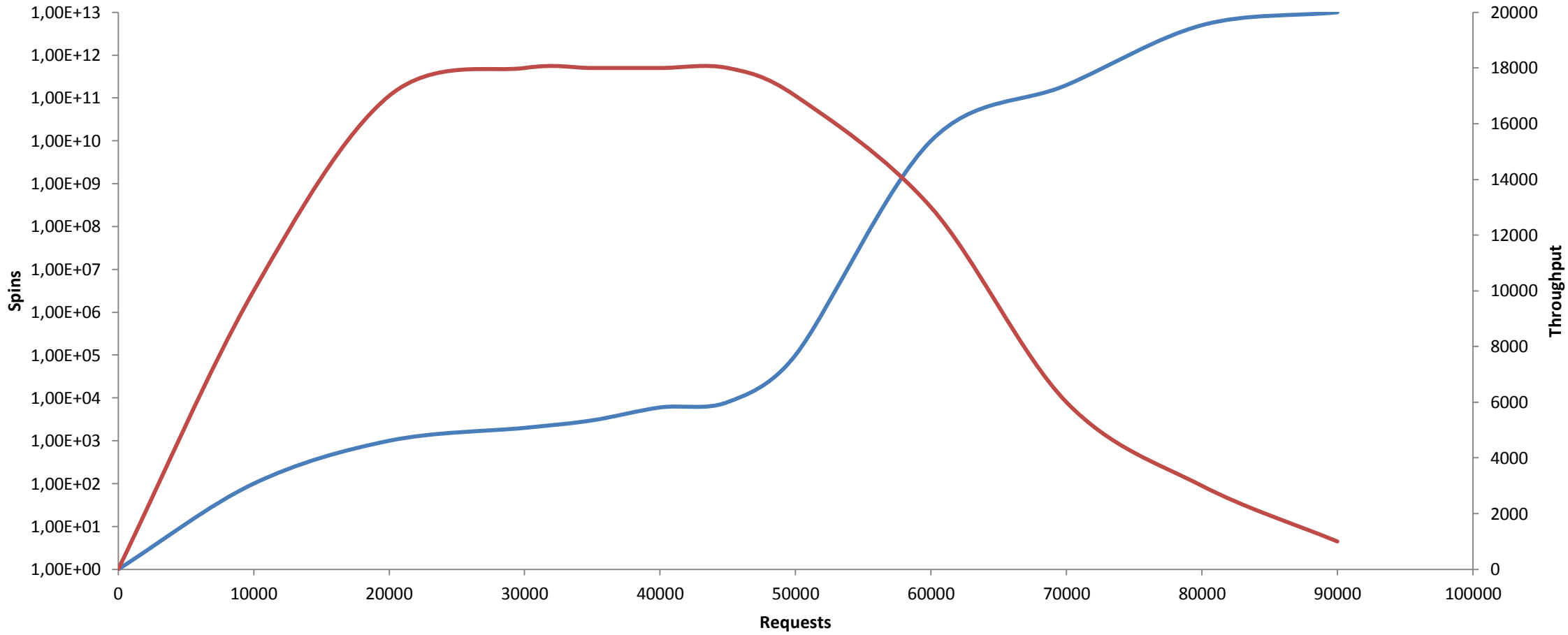
THE PROBLEM WITH DATABASE SCALABILITY



Everything changes at large scale

- As the workload increases and becomes more complex, you face more and more contention on various resources:
  - Locks
  - Latches
  - Spinlocks
  - Memory
  - Tempdb
  - Even the Query Optimizer itself

What it MIGHT look like



Two approaches to scalability

- Scale-up – „let's just buy a bigger server"
  - Expensive (Big Iron, Enterprise Edition licensing)
  - Does not always work (esp. for contention related problems)
- Scale-up – „let's spread it across many smaller servers"
  - Usually, some form of data distribution and workload offloading
    - Replication, Federated Databases, Log Shipping, AlwaysOn Readable Secondaries
  - Complicated to the point of impossibility
  - Requires code changes and smart application design
  - AlwaysOn Readable Secondaries are cool, but...

The problem with AlwaysOn

- Readable Secondaries allow R/O queries
- Sounds like a perfect solution for scale-out and load-balancing, but...
- There's nothing in the SQL Native Client to enable that
- In order to do that in practice you need:
  - Two database connections in the application
  - Some logic in the application to distinguish R/W from R/O
  - All that means serious code and architecture changes
  - Most likely not possible for an existing, deployed application

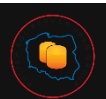
Interesting thing about workloads...

- In most workloads, vast majority of batches are SELECTs
- These are often small, R/O queries, looking quite innocent...
- But mixed with even a small number of writes, they become a WMD
- Reads blocking writes
- Writes blocking reads
- Lots and lots of contention...

Another interesting thing...

- Many workloads include a high number of queries that return the same data each time they are executed
- The application asks the same question, over and over and over...
- And gets the same answer...
- Even if the data changes, it changes infrequently
- But the work has to be done each time:
  - Resource usage (CPU, memory, tempdb etc.)
  - Possible contention in many places
- That means lots of „unnecessary” queries

I STARTED LOOKING FOR A SIMPLE SOLUTION...



I asked myself some questions...

- What if we could cache queries?
- What if we could scale-out with out the need for code changes?
- What if we could fix some of those „monster queries”?

- Wouldn’t that be a dream-tool of every SQL Server consultant?

SQLDay 2015

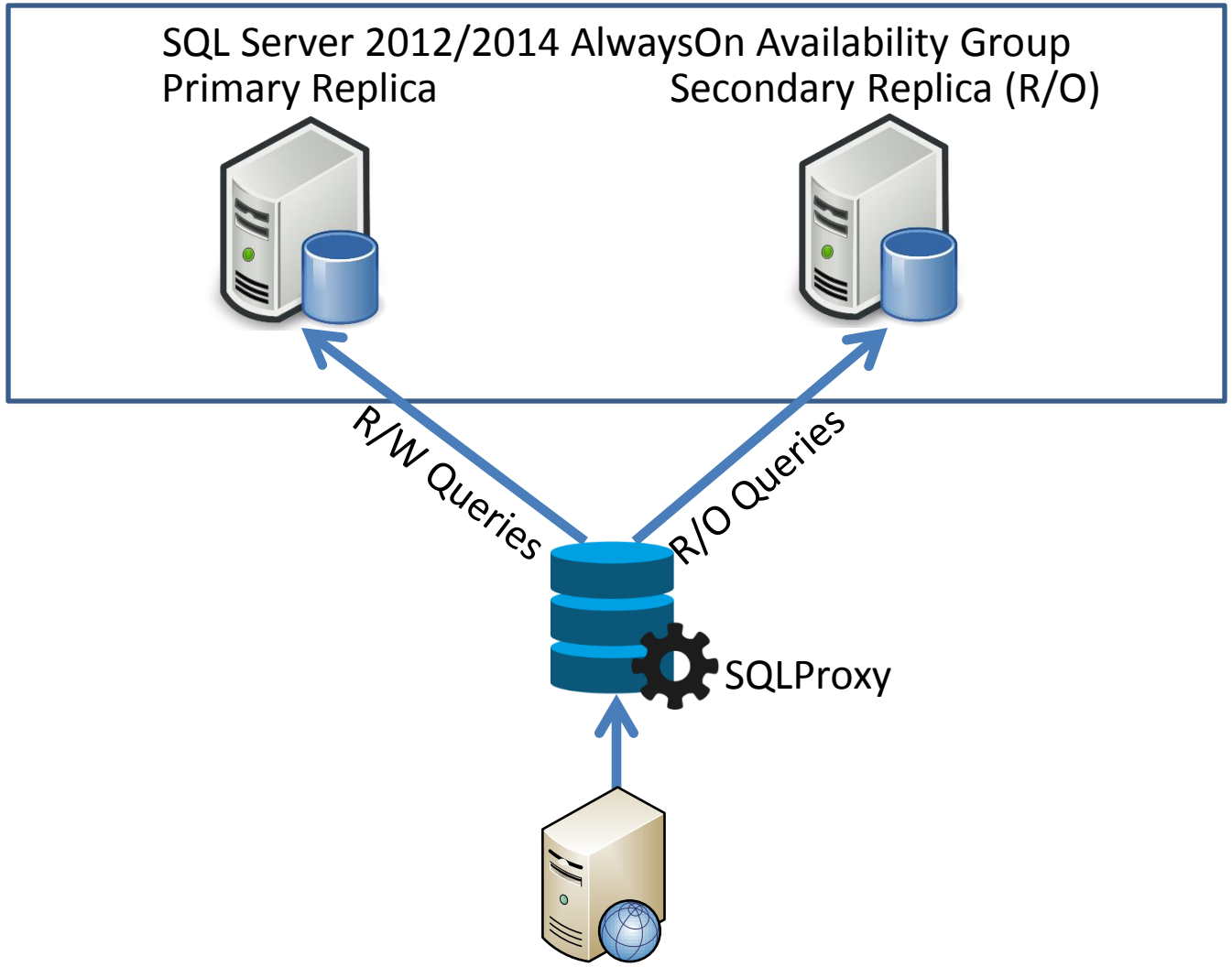
Introducing SQLProxy

- Fully transparent TDS proxy
- SQL Server and T-SQL aware
- Functionalities include:
  - Routing of R/O queries to the Readable Secondary
  - Caching of the query results
  - Dynamic query rewrites

Routing of read-only queries

- Allows scale-out and load-balancing using AlwaysOn Readable Secondaries
- Currently, SNAC does not support this natively and major application changes are needed to achieve that
- SQLProxy provides query routing that is:
  - fully transparent to the application
  - fully configurable
  - customizable

Routing of read-only queries



How R/O routing works

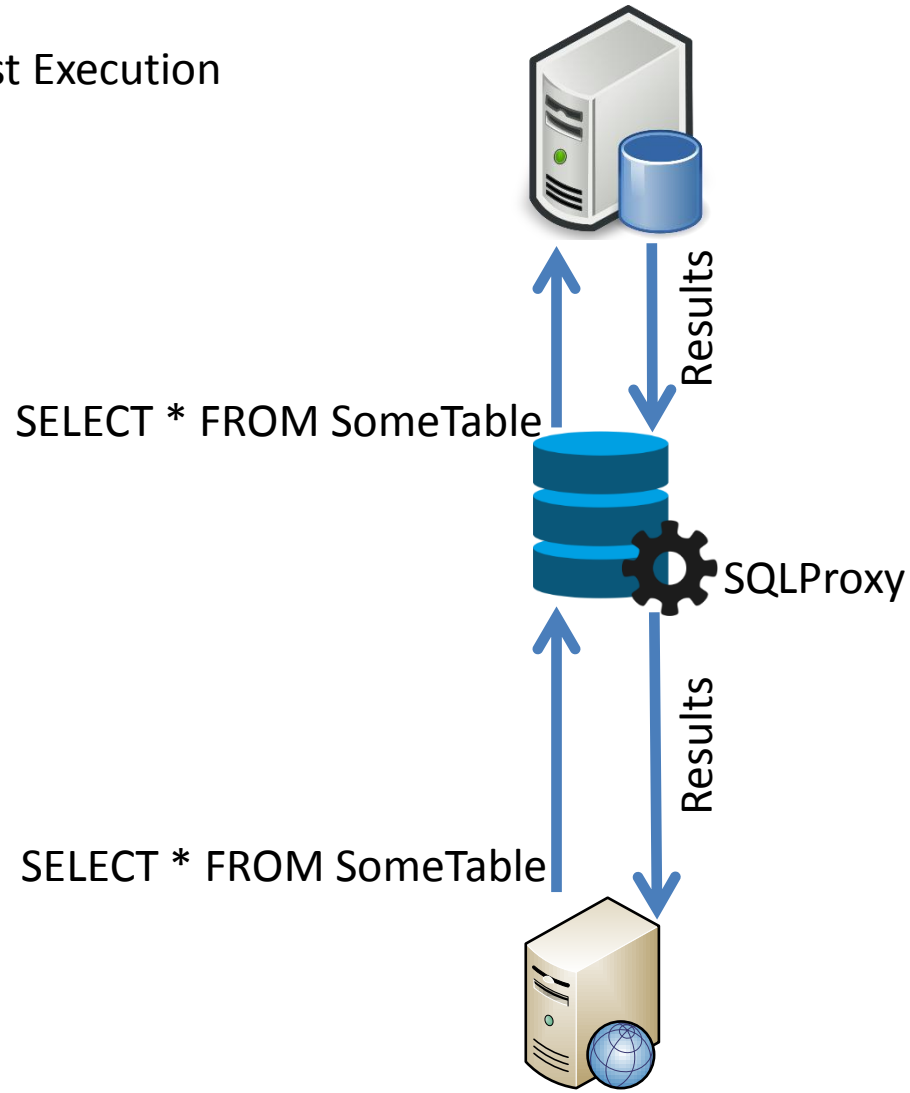
- Configure SQLProxy to connect to the AW Listener
- SQLProxy uses AW routing information to open a R/O connection to the Readable Secondary Replica
- Application opens a single connection to the SQLProxy
- Every batch is classified and executed on an appropriate server
- Classification can be:
  - Declarative (you specify which queries should be routed where)
  - Dynamic (by parsing the batch text and examining it)

Query result caching

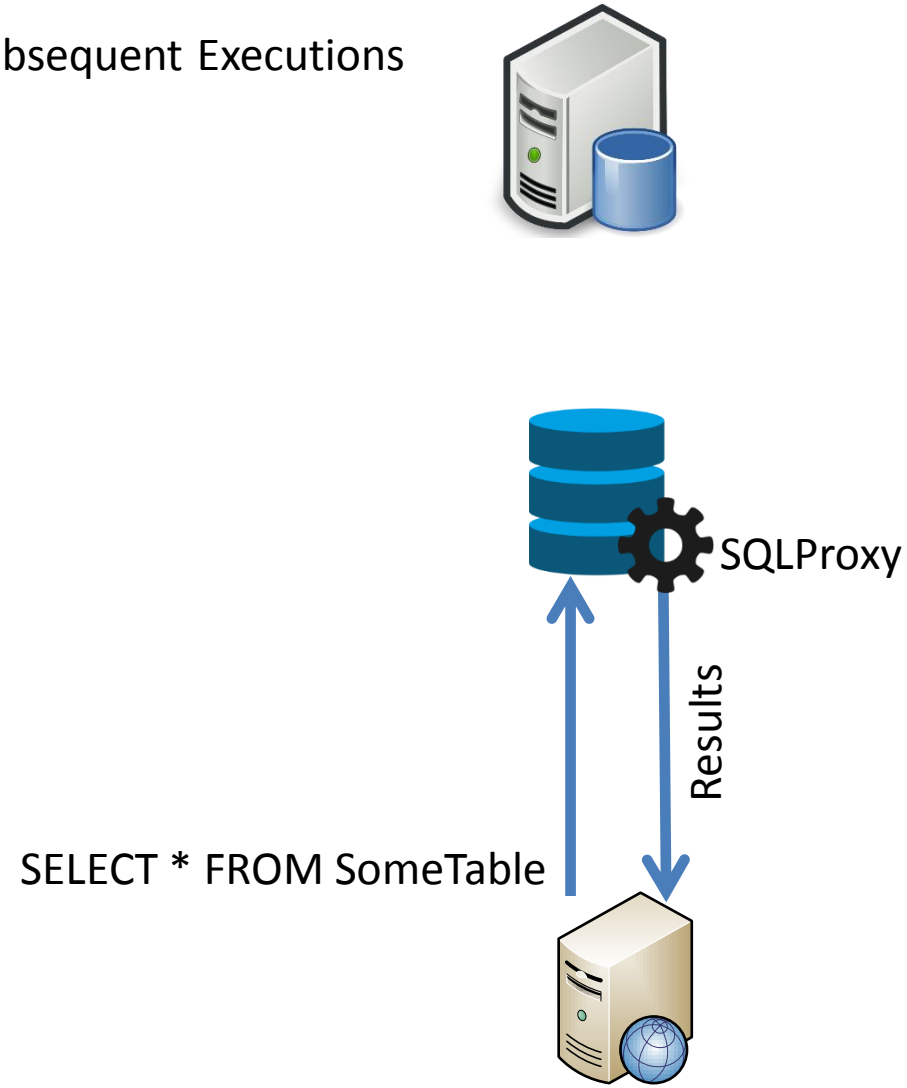
- SQL Server caches a lot of things, including:
  - Data pages
  - Execution plans
- But your query is executed EVERY TIME you send it, taking server resources
- Solution:
  - Caching the query results

Data caching

First Execution



Subsequent Executions



How query caching works

- Fully transparent to the application
- Fully configurable, with cache expiration
- Works for R/O queries only (obviously)
- Works with parameterized queries and SP calls
- Time-bases expiration
- Configurable „on the fly”, with management interface

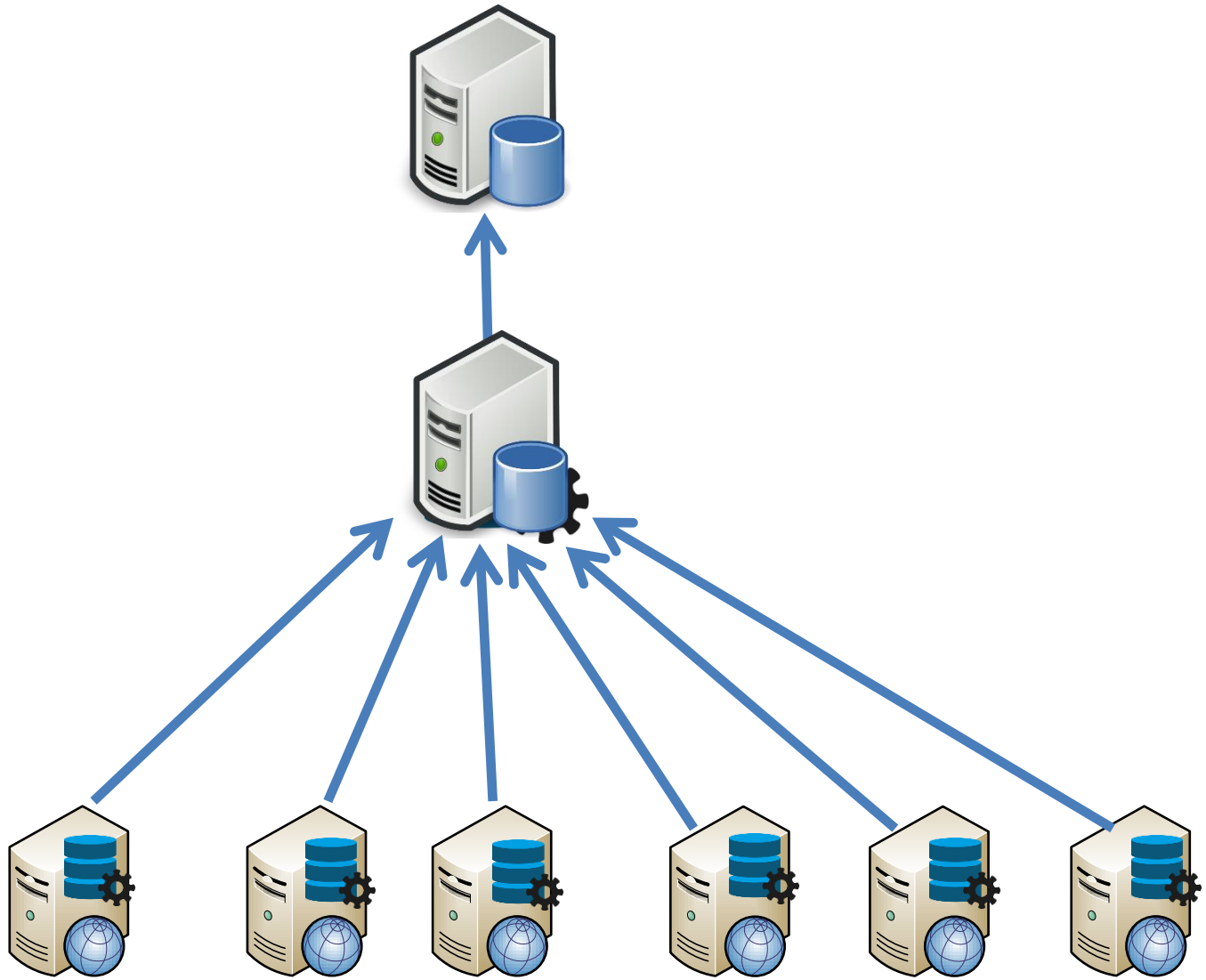
What about cache invalidation?

- Forget about it! ☺
- Too complex and resource intensive to implement in practice
- We use pure time-based expiration
- TTL of the result in cache is your decission:
  - Some applications will be just fine with the „stale” data
  - Some other may require more up-to-date information

What about scaling the SQLProxy itself?

- Will it become a bottleneck or single point of failure?
- NO!
- It’s just a gateway and it’s more-or-less stateless, so no problem:
- Just have more instances of SQLProxy and spread the load
- SQLProxy supports Windows Failover Clustering
- And it works with the –FailoverPartner switch in SNAC

Scaling the SQLProxy



What's in version 1.0?

- Query result caching
- Query routing between AG replicas
- Early implementation of on-the-fly query transformations
- Server connection pooling

- With some limitations:
  - SQL Authentication only in v. 1.0

What more can we do with it...

- Some ideas for version 2.0:
  - Auditing, logging etc.
  - Security (enforcing access rules, preventing SQL injection)
  - Error detection/correction (i.e. 1205 – Deadlock victim)
  - Translation between SQL dialects
  - Application compatibility with new versions of SQL Server
  - Data obfuscation
  - T-SQL language extensions

What's next?

- We are in a private Beta stage
- If you are interested, we want to talk to you!
- Wanna try it with your application? We want to talk to you!!!

- Come and grab me during the Party, I will answer all your questions
- AND we can have a drink together! ☺

- Visit us at www.sqlproxy.net for updates and more info

SQL PROXY

www.sqlproxy.net

THANK YOU!

DATA MASTER

LET IT SCALE

Strategic Sponsor

Microsoft

Gold Sponsors

PYRAMID ANALYTICS

KRAJOWY REJESTR DŁUGÓW www.krd.pl

devart Database Interaction Solutions

Silver Sponsors

Profisee®

SQL EXPERT.pl

CREDIT SUISSE

Technical Partners

Microsoft

PTI POLSKIE TOWARZYSTWO INFORMATYCZNE ODDZIAŁ GÓRNOŚLĄSKI

Academic Partners

Wyższa Szkoła Zarządzania i Bankowości w Krakowie

Wyższe Szkoły Bankowe

Media Partners

SPECJALISTA IT PROGRAMISTA DESIGNER GEEKCLUB.pl wss.pl

ERP24.pl

4programmers.net

IT professional

IN FACT PR

itnews24

virtual-it.pl

IT NEWS POLSKA

VirtualStudy

MAGAZYN IT PL

Made in Wro Do IT here!

ERP-view.pl Portal rozwiązań IT w biznesie