IN-DATABASE APPLICATION SERVER

PostgreSQL as a server-side application development platform

Luxms
g r o u p

Dmitry Dorofeev,
Sergey Shestakov

Postgres Conference 2019

THE RISE OF DATA-CENTRIC PARALLEL COMPUTING

- Gordon Moore's law: is it working?

- Next Generation Databases: NoSQL, NewSQL, MPP
  - ❖ REST API-enabled, easily shared

- Is it a good time to create in-database app servers?

- Let's compare 2-tier in-database app server
  with classical 3-tier:
  - ❖ Luxms BI, the analytical platform: our target is MPP BI

RELAX WITH NOSQL HYPE, GO BACK TO SQL!

- **NoSQL lacked … SQL support!**

- **Hybrid NewSQL: some SQL, distributed, in-memory**
  - ❖ but: no complete support for ANSI SQL 99, no procedures
  - ❖ Fog Computing: pre-collection, pre-aggregation

- **PostgreSQL/Greenplum is ready for the future:**
  - ❖ Backed by relational algebra (strong math)
  - ❖ guaranteed, consistent, clear answers to your queries
  - ❖ horizontally scalable, advanced extensions
  - ❖ in-database programming languages, in-memory processing

Luxms

HOW TO PROCESS BIG DATA?

- **Move computation to data**, ~~move data to computation~~:

  - ❖ **Data-centric MapReduce = Hadoop ecosystem**

  - ❖ **Fog Computing: pre-collection, pre-aggregation**

  - ❖ **Surprise: PL/* inside MPP database**
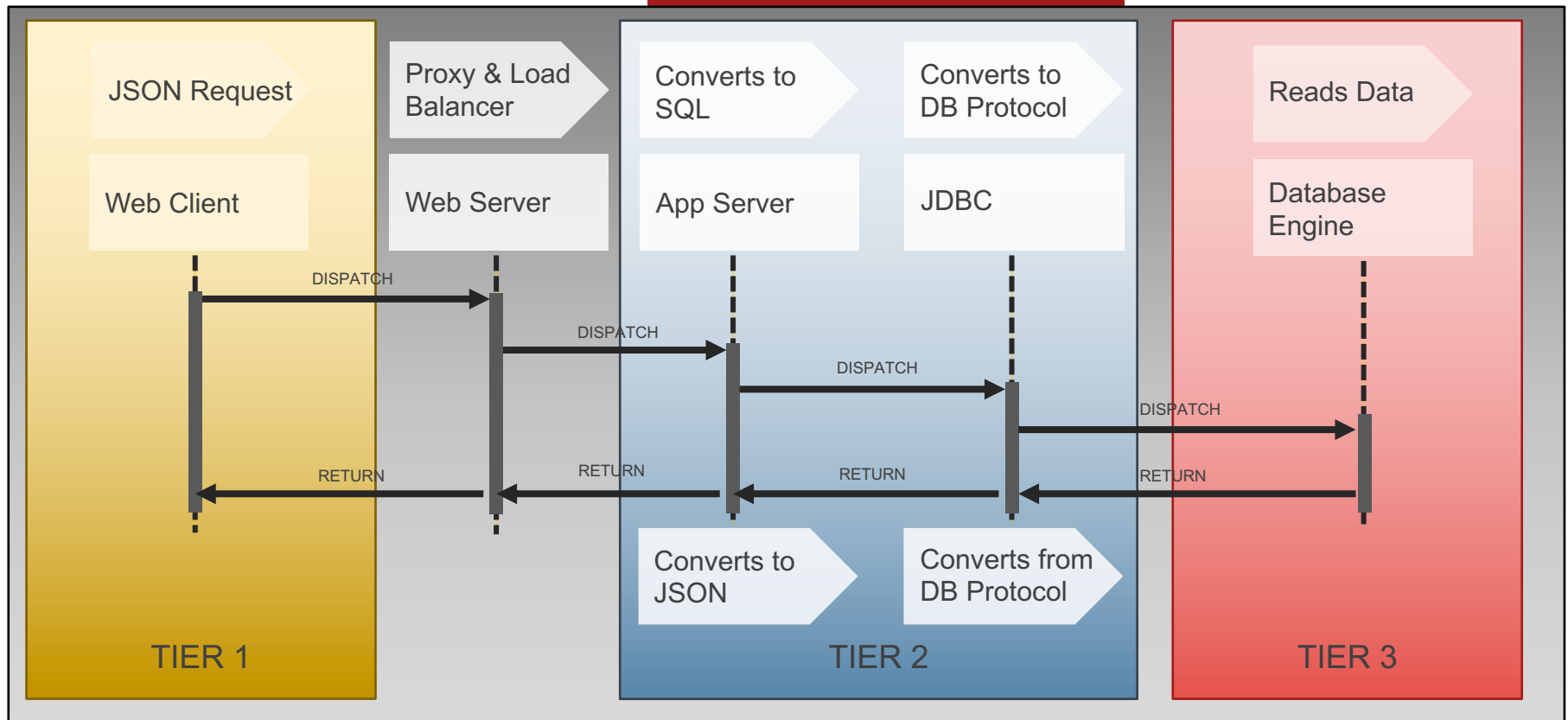
  - ❖ **PL/pgSQL in GPDB may be used like MapReduce**

- gpmapreduce

  - ▪ **Runs Greenplum MapReduce jobs as defined in a YAML specification document.**

Luxms

DATA-CENTRIC APPROACH
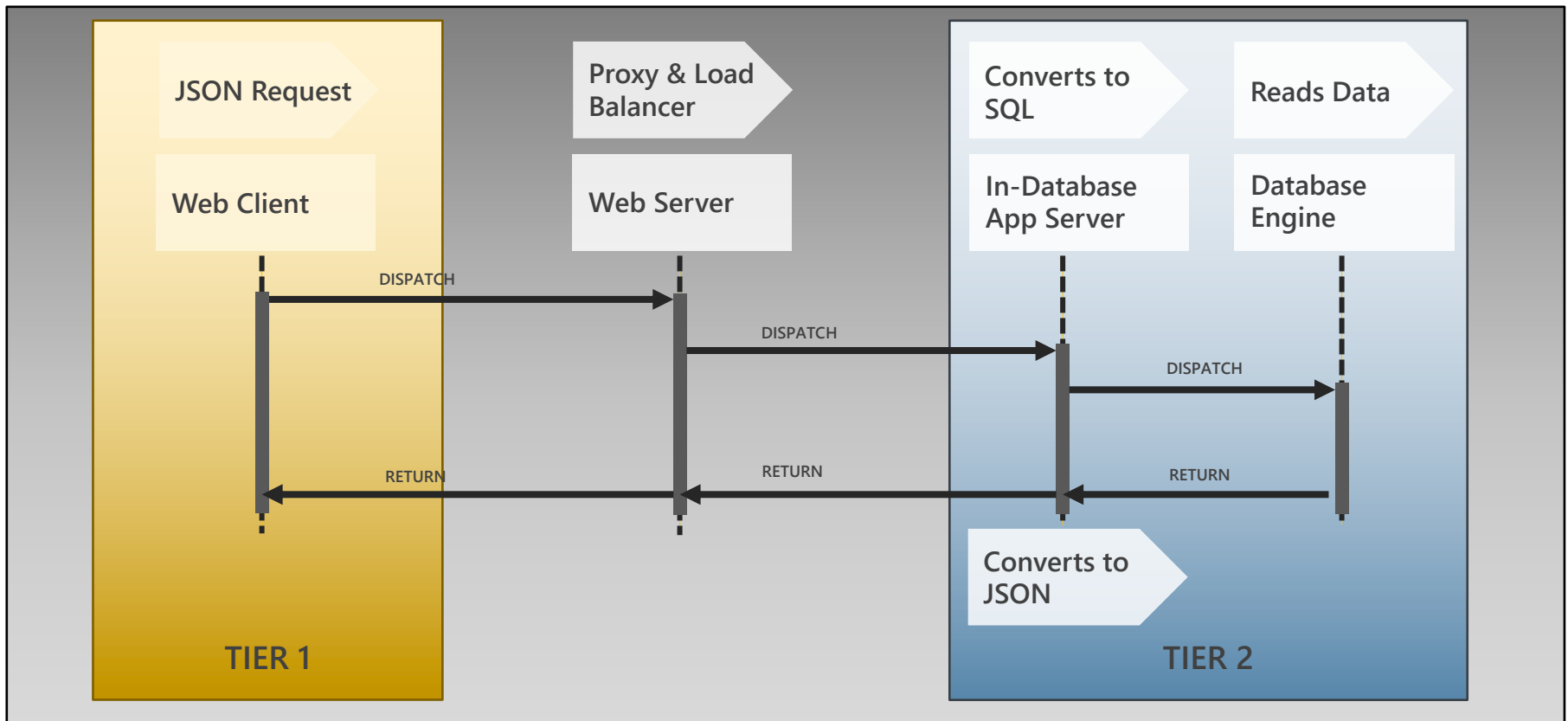# FOR ANALYTICAL PROCESSING

3-tier architecture:

REDUNDANT DATA CONVERSION

TIER 1		TIER 2		TIER 3
JSON Request	Proxy & Load Balancer	Converts to SQL	Converts to DB Protocol	Reads Data
Web Client	Web Server	App Server	JDBC	Database Engine

DISPATCH

DISPATCH

DISPATCH

DISPATCH

RETURN

RETURN

RETURN

RETURN

Converts to JSON

Converts from DB Protocol

Luxms

DATA-CENTRIC APPROACH FOR ANALYTICAL PROCESSING

Data-centric 2-tier architecture:

IN-DATABADE ANALYTICAL SERVER (PostgreSQL)

PL/PGSQL DEVELOPMENT

It is more like Smalltalk/LISP/Erlang:
fixing code in the nonstop system
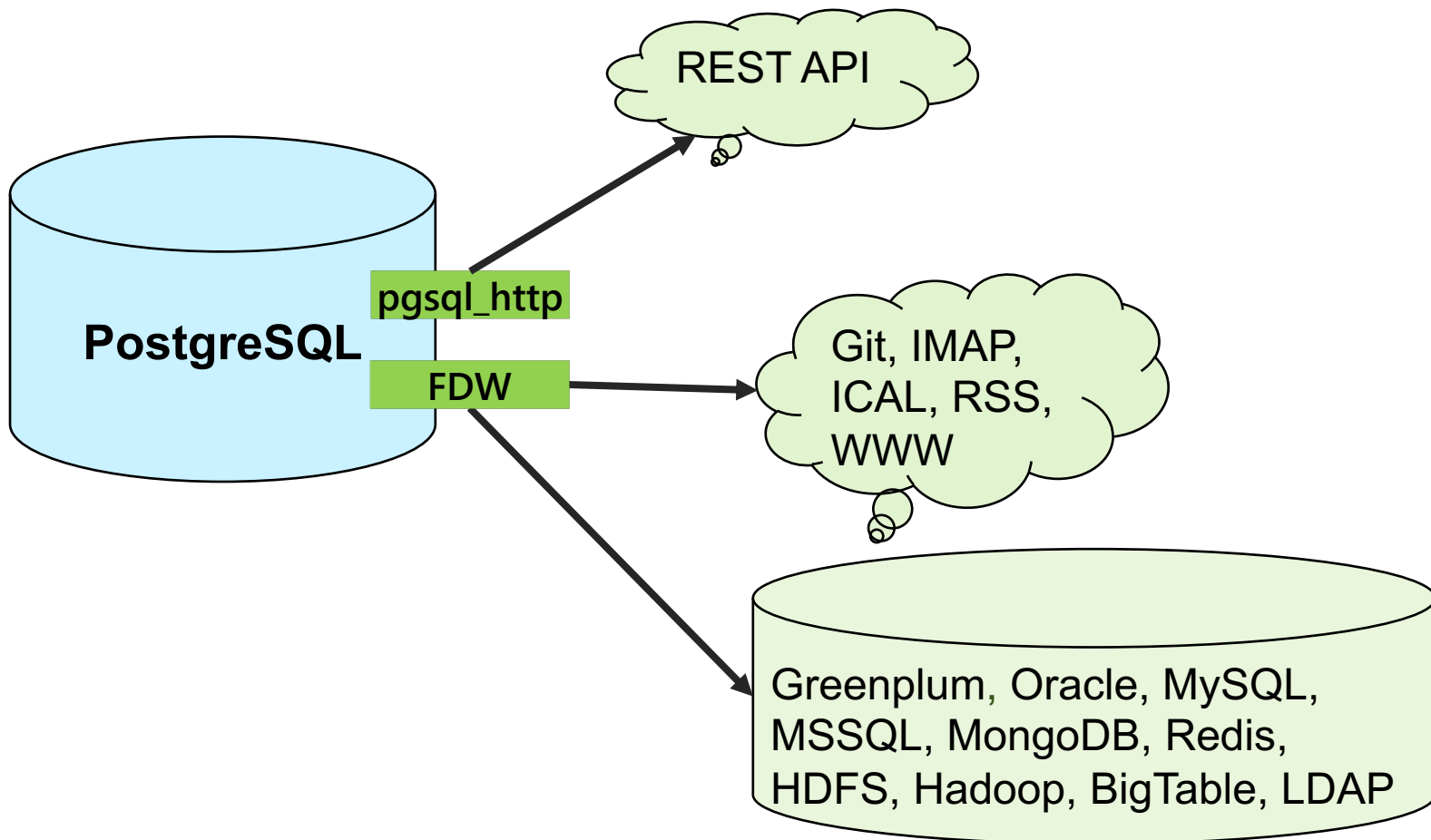
We should learn from Pharo (modern and free Smalltalk):

	Pharo	**PostgreSQL**
Source code management	Iceberg	??????
Debugging	Live debugger	omnidb.org
Code quality	Live Quality Assistant (lint on steroids)	plpgsql_check
Unit testing	Sunit	pgTAP

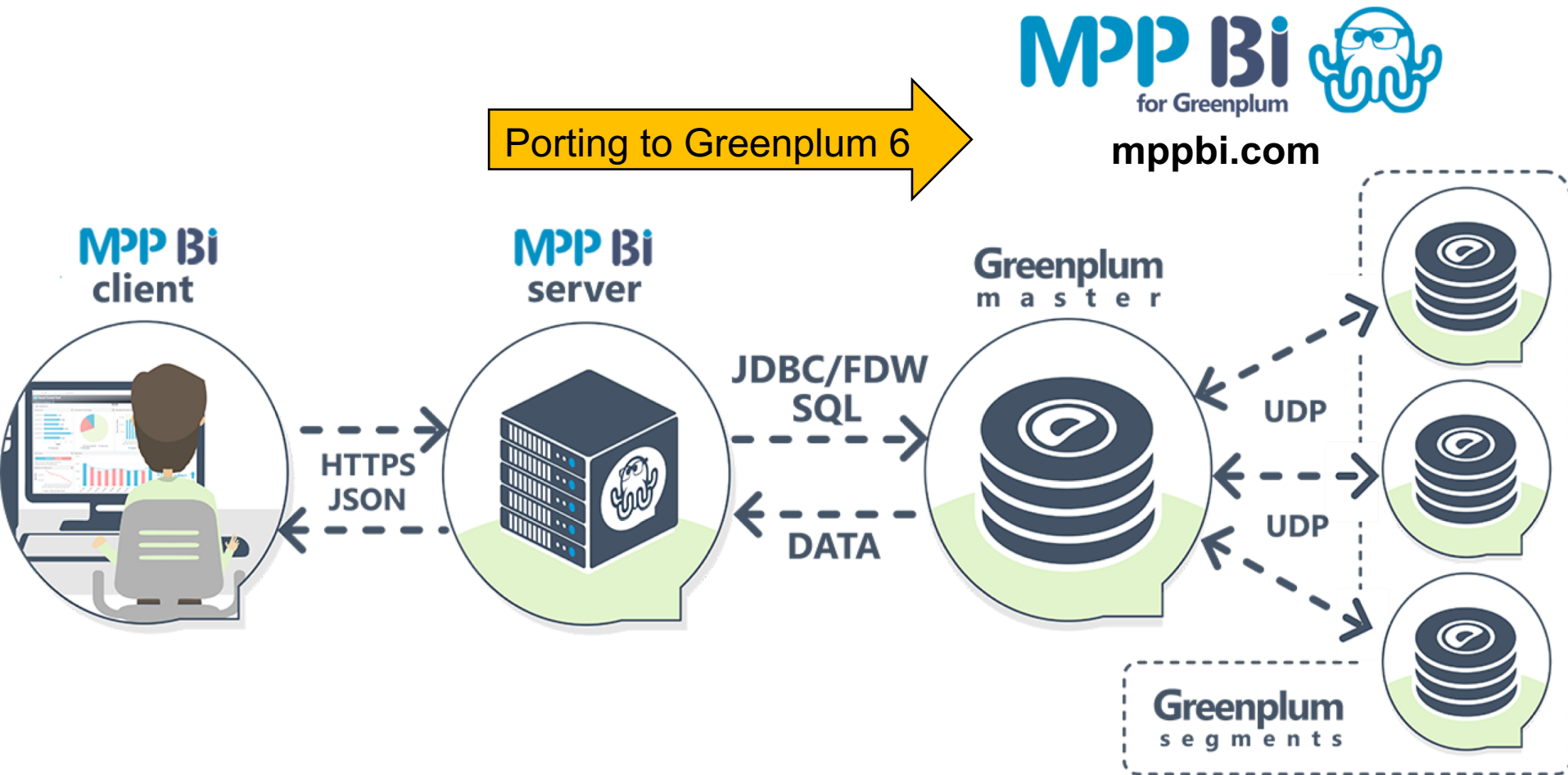# EMULATING HTTP QUERY WITH PLAIN SQL

```
SELECT * FROM webapi.route('POST',              HTTP METHOD
'/api/data/ds_270/colors',            URL
'{"luxmsbi-user-session":"secret-session-key"}'::JSON,   HEADER
'{"version":"2.0","cube":{
    "parameters":[2800],
    "metrics":[35],                   BODY
    "locations":[10001,10002],
    "periods":[16052]}}'::JSON,
'{}'::JSON);           QUERY STRING


-[ RECORD 1 ]--------------------------------------------------------
status | 200
headers| {"content-type": "application/json"}
body   | [{"metric_id":35, "loc_id":10008, "period_id":16052,"color":"red"}]
```
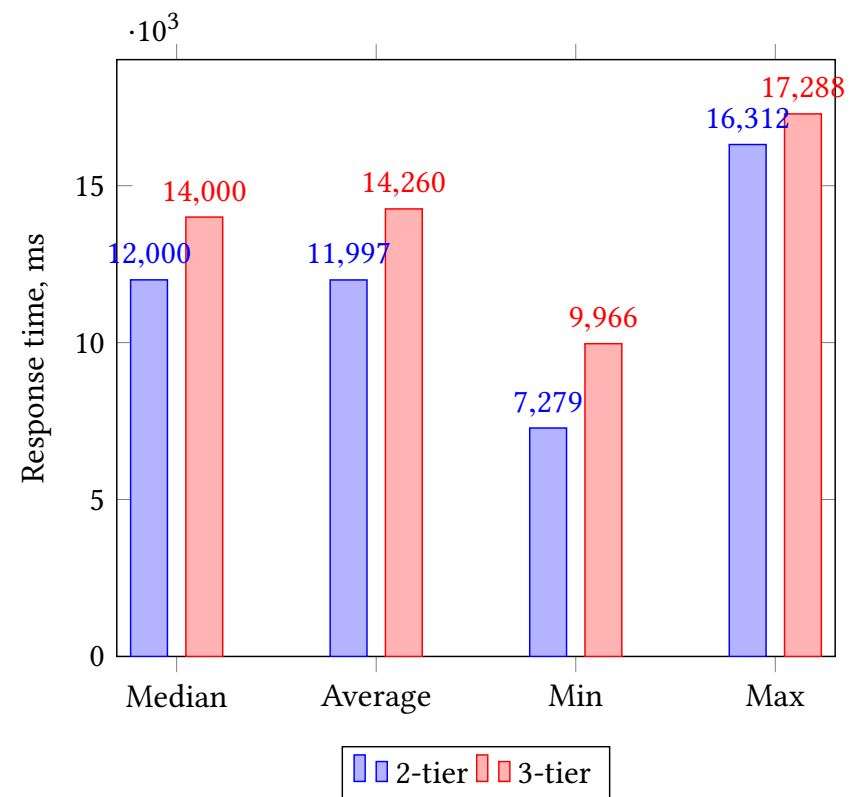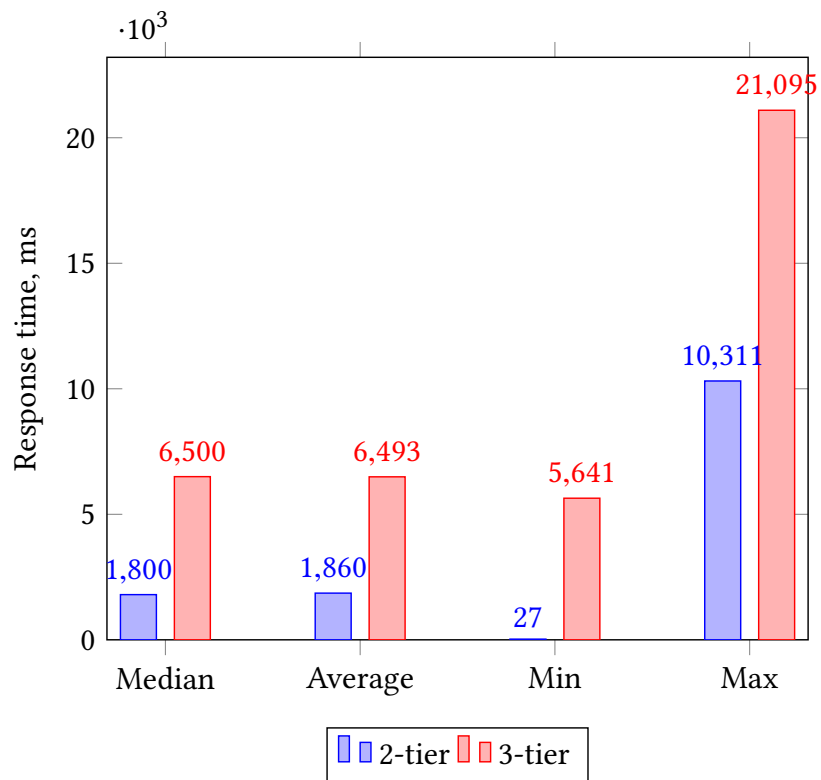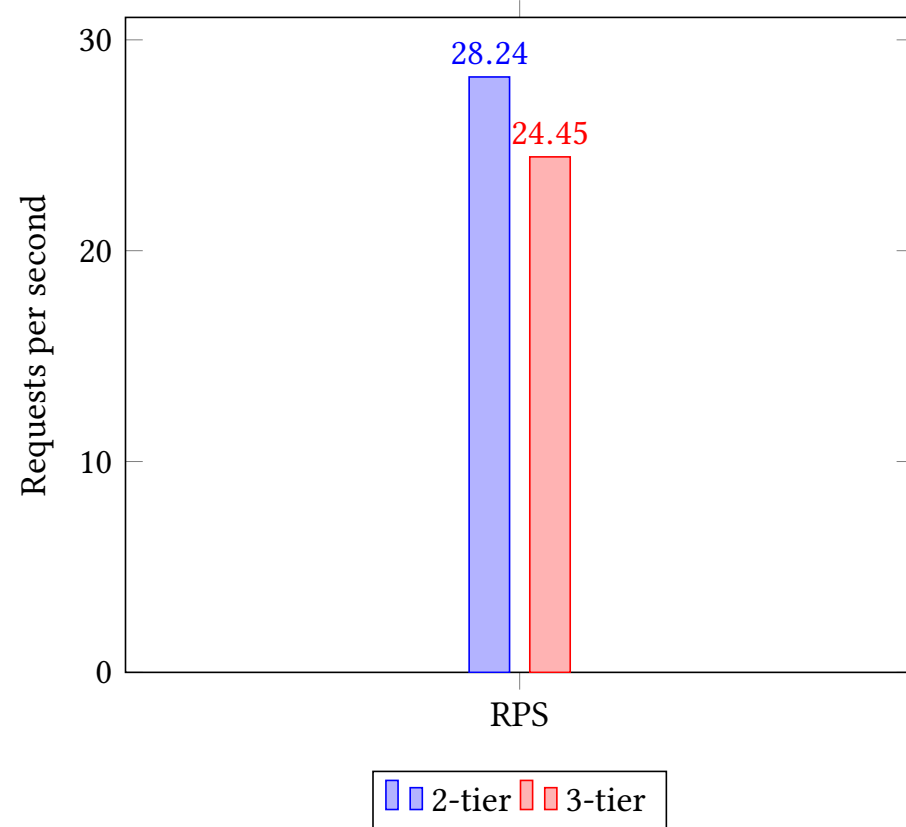
POSTGRESQL EXTENSIONS

REST API

PostgreSQL

pgsql_http

FDW

Git, IMAP, ICAL, RSS, WWW

Greenplum, Oracle, MySQL, MSSQL, MongoDB, Redis, HDFS, Hadoop, BigTable, LDAP

Luxms

FUTURE PLANS: MPP BI for Greenplum 6
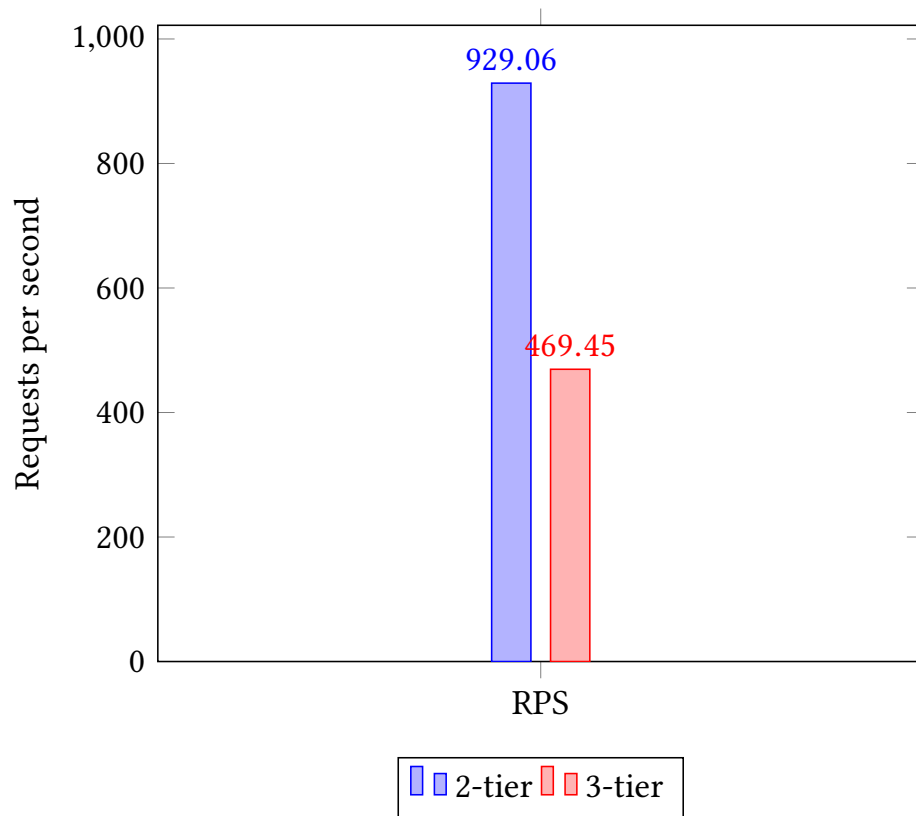
BENCHMARKING

Lightweight and Medium Workloads: Latency

Luxms

BENCHMARKING

Lightweight and Medium Workloads: RPS

Luxms

THANK YOU!

QUESTIONS?

Serg Shestakov   Serg@Luxmsbi.com

Dmitry Dorofeev   Dima@Luxmsbi.com