1º QlikDev Latino América

Optimize your QV Application

Bob Craig

Expert Services

Optimize your QlikView Applications - Agenda

- Overview of what drives QlikView Performance
- How to measure performance?
- Hardware Configuration Options
- Tools to estimate Hardware Requirements
- Real Use Case
- Data Model Performance Considerations
- QlikView Document-Specific Performance Considerations

QlikView

“Performance”

- Acceptable time the user has to wait for a result after a selection
  - 30 sec. can be acceptable when analysis queries took hours before QV
  - 10 sec. can be inacceptable for users, where prior systems showed the results within 1 second
  - acceptance is also user group dependend – executives doesn’t accept busy symbols for dashboard applications.
- General: BI systems lose acceptance, when average response time is above 15 seconds
- How long does it take to load the data in a QV model
  - The load of new data must fit in the available time window (e.g. the time between DWH- update and working day time)

QlikView

Basic factors for performance

- Number of records ?
- How complex is the data model ?
- How complex are the calculations/the charts ?
- How many users ?
- How often do the users click ?
- How many cores (processors) ?
- Is there enough memory ?

Document

Users

Hardware

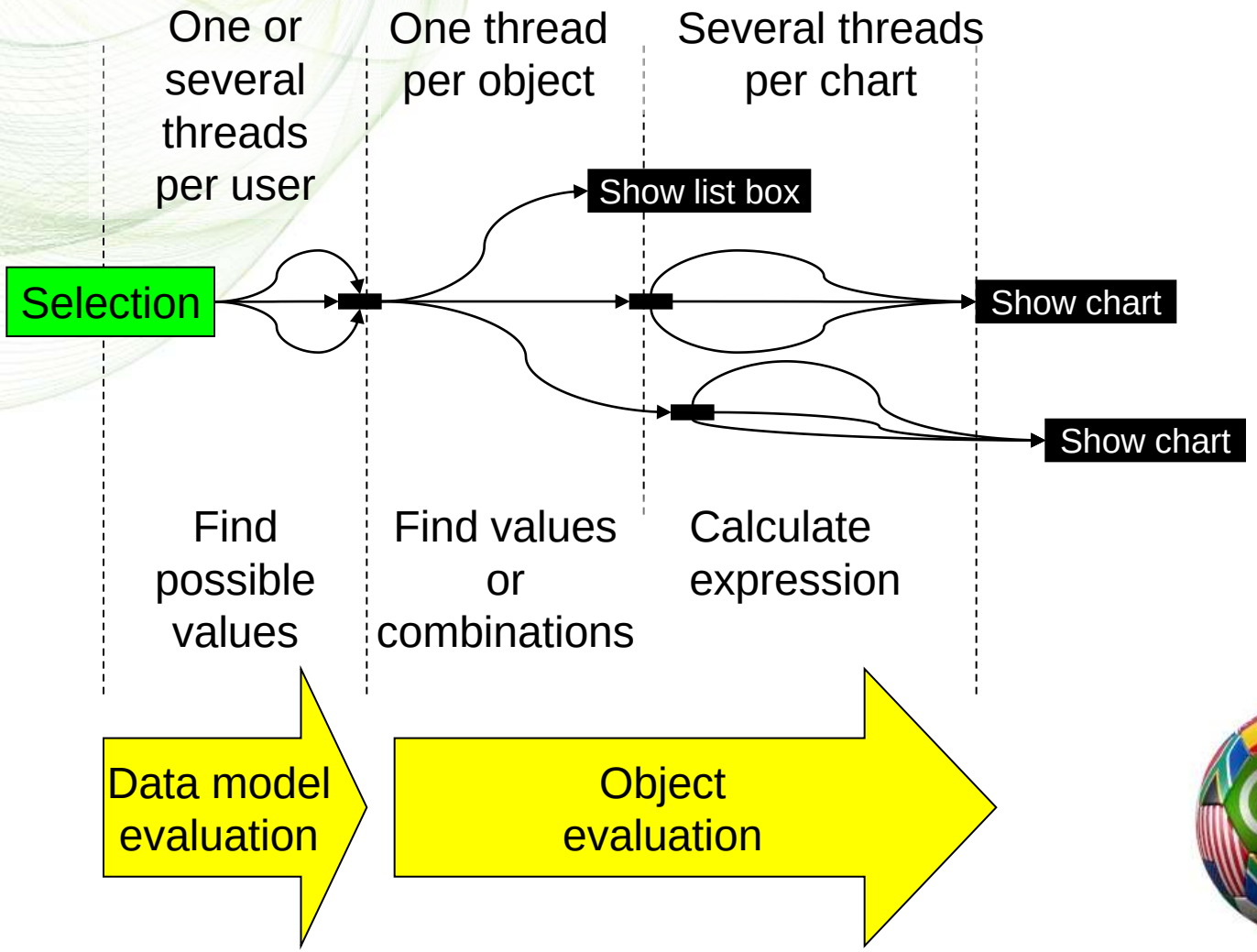
QlikView

Basic facts affecting performance

- QlikView loads detailed data – un-aggregated data on transactional level
- QlikView aggregates and calculates on demand – when the user clicks
- All heavy calculations are made on the server
- QlikView is multi threaded – all processors can be used

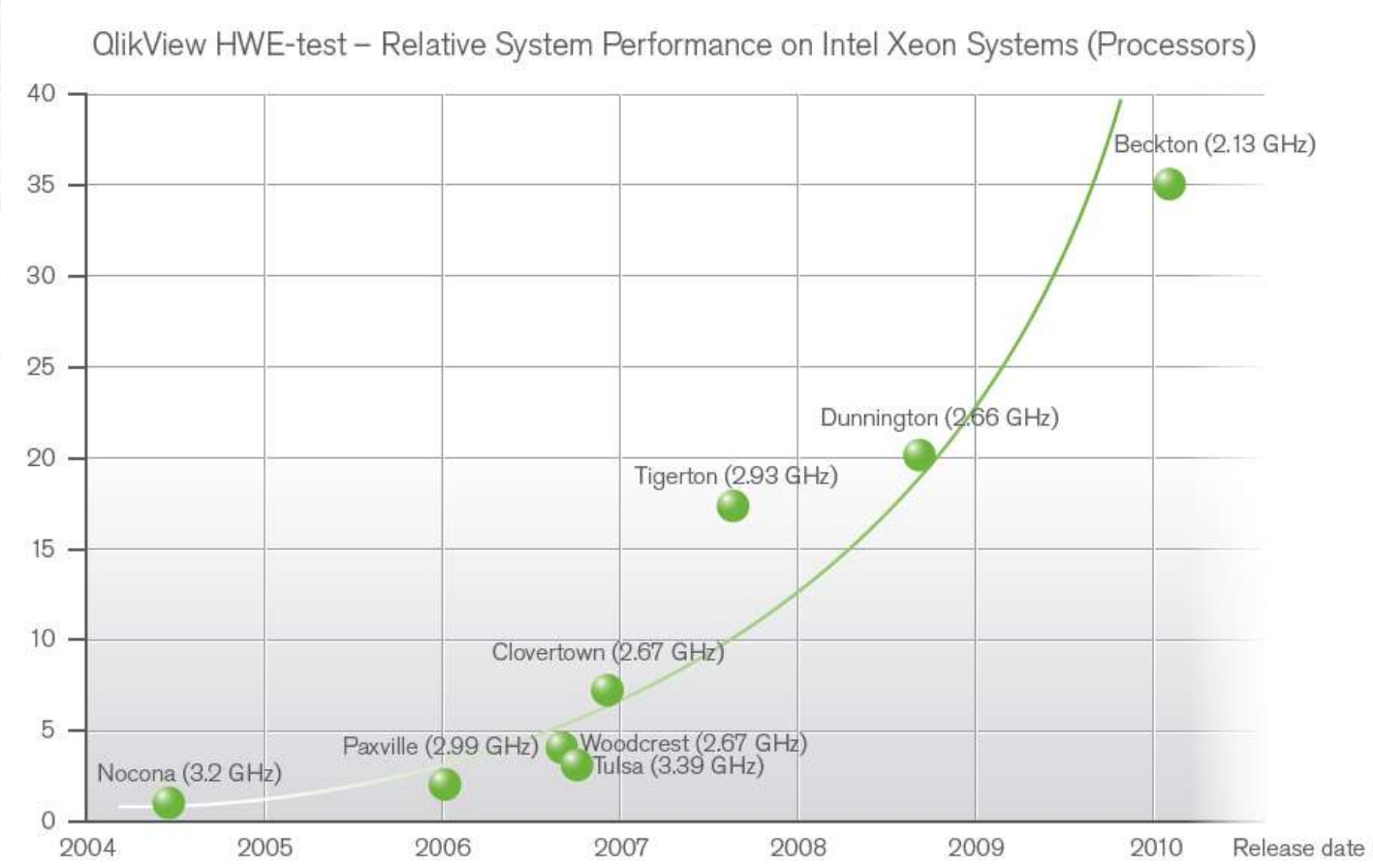
QlikView

Multi threading



QlikView

QlikView utilizes the technology!



QlikView

Current benchmarks

Platform	# of CPUs	# of cores	Clock speed	Time to complete test
Penryn	4 (X7460)	6 (total 24)	2.67 GHz	16.833 sec
NehalemEX	4 (X7560/Beckton)	8 (total 32)	2.13 GHz	9.765 sec

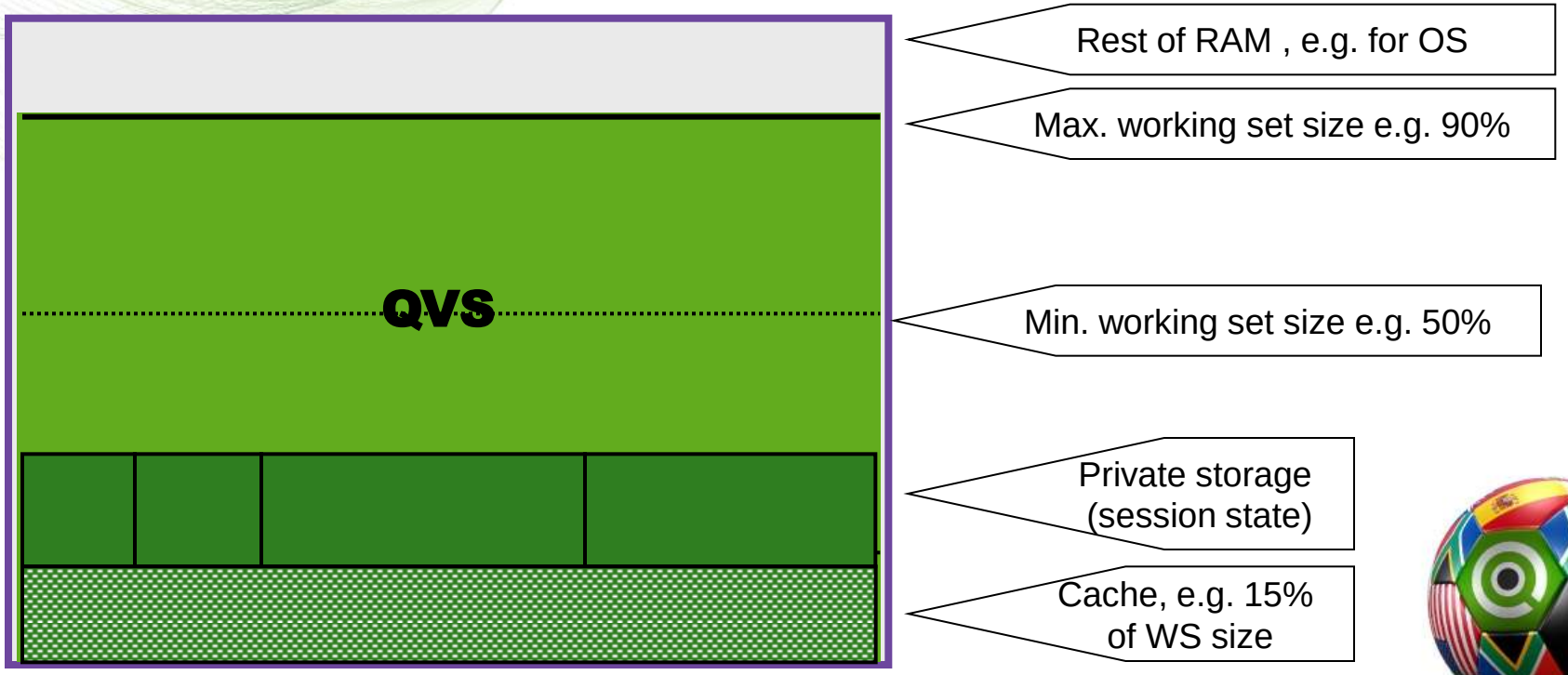
1.72x performance increase

- with only 33% increase in # of cores
- running at only 80% of the clock speed
- in less than a year...

QlikView

Hardware: QlikView In The Server RAM

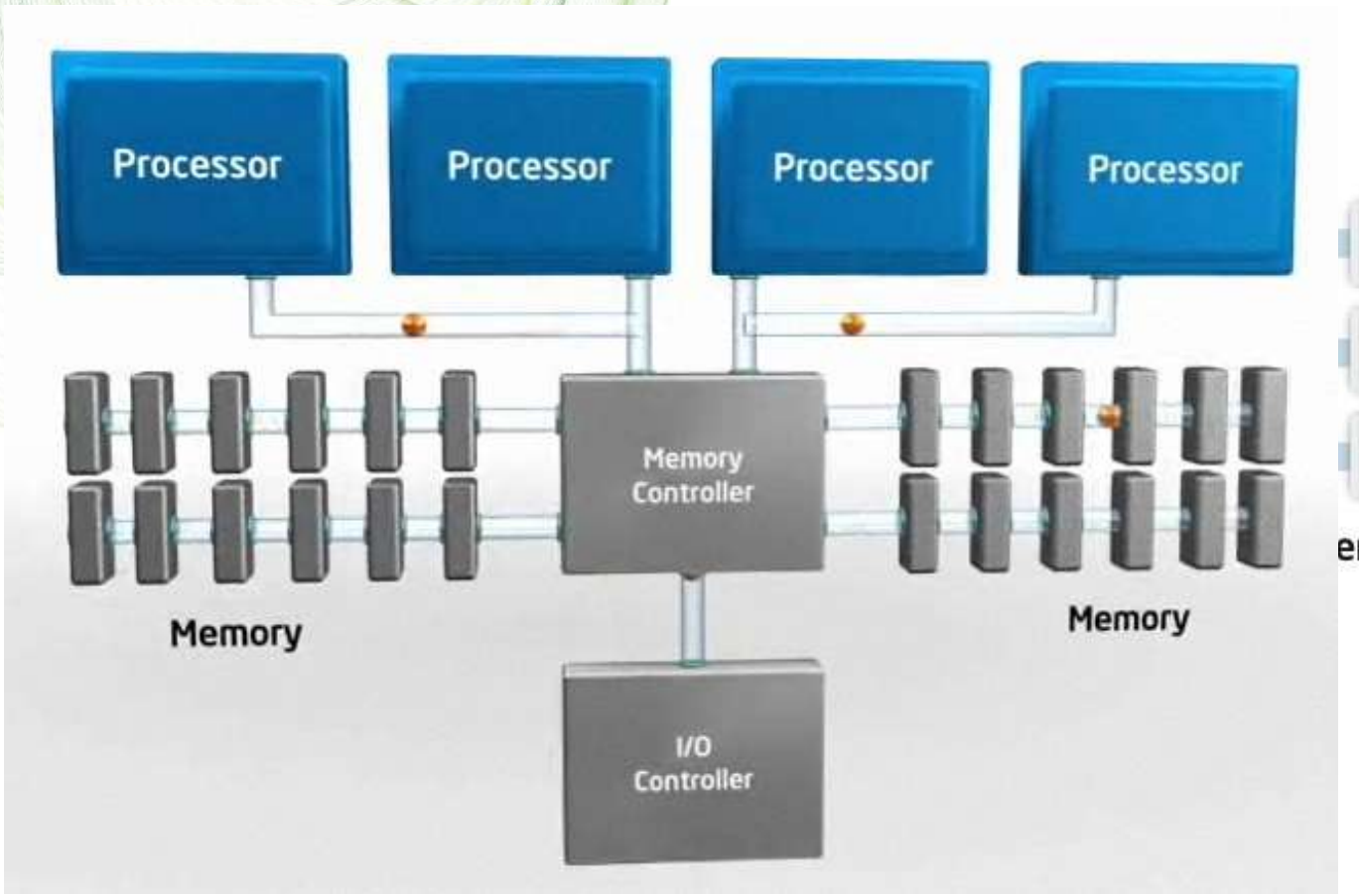
Memory Usage QVS



Working set size in % of server RAM, Cache in % of working set size

QlikView

Hardware: RAM and CPU communication



QlikView

Hardware: QlikView CoExistence with other Software

- DB systems shouldn’t be installed on the same box, because they grab all available RAM by default (for buffers etc).
- Other software which grabs higher percentage of CPU shouldn’t be on the same physical box during online time and script execution
- The usage of virtualisation…

QlikView

Hardware: Server Sizing

- **RAM**
  - Size of the source database says almost nothing.
  - Are all tables and all columns needed, how many indexes are set in the db ,
  - Does it include ”redo” data, how are the db segment fill grade parameters ...
  - Also the data content has an impact on the size within QV – e.g. the diversity in the fields
  - (e.g. time stamps in comparison to gender)
- **CPU Power**
  - Determined by the amount of data,
  - The data model,
  - Amount and complexity of formulas within the QV sheet,
  - Filter detail, Used functions and other factors.

QlikView

Hardware: Server Sizing („RAM-CPU-Requirement.XLS“)

Memory/CPU requirement for simultaneous session per application

CALCULATION RESULTS MAY DIFFER FROM REAL SIZES UP TO 50% IN EACH DIRECTION DEPENDING ON DATA CONTENT&STRUCTURE AND AMOUNT&COMPLEXITY OF THE APPLICATION FORMULAS AND AVAILABLE CPU POWER

Application	Size source data (text files)	Application file (qvw - compress ratio 1/12)	Application RAM (8 x qvw size)	Add RAM request per user session (10-13% of base data in RAM)	Amount of registered (distinct) users	Amount of simult. sessions per app. in nn % of registered users	Total RAM required: RAM application + session RAM	CPU time per query (avg. time per user and query) (2)	Amount of queries (clicks) per user session
	in GB	in GB	in GB	in GB		15%	in GB	in sec.	in a min.
app1	5	0,42	3,33	0,43	100	15	9,83	8	2
app2	2	0,17	1,33	0,17	500	75	14,33	3	2
app3	0,5	0,04	0,33	0,04	500	75	3,58	1	2
		0,00	0,00	0,00		0	0,00		
		0,00	0,00	0,00		0	0,00		
		0,00	0,00	0,00		0	0,00		
Total		**0,63**	**5,00**	**0,65**		**165**	**27,8**		

Amount of required GB & CPU-kernels

Size of alternative servers

Server alternatives	1 Quad Core (capacity: 4 CPU-Kernels x 60 Sec = 240 CPU-Kernel-sec / min.)	2 Quad Core (capacity: 8 CPU-Kernels x 60 sec = 480 CPU-Kernel-sec / min)	4 Quad Core (capacity: 16 CPU-Kernels x 60 sec = 960 CPU-Kernel-sec / min)	8 Quad Cor (capacity 3 CPU-Kerne x 60 sec =1 CPU-Kerne sec / min)
Capacity: simultaneous sessions				
Max. simultaneous sessions one box can handle:	47	94	189	377
Amount of servers needed:	4	2	1	1
GB of RAM each incl 30% buffer:	14	22	37	37
min. GB required for scripts (1):	9			

Minimum free RAM needed for QlikView on each server

1) Premise: scripts for model load are running in sequential order

2) CPU seconds <> time seconds on multi cpu environments;
e.g. 6 CPU seconds = 3 time seconds on a 1-CPU dual core box

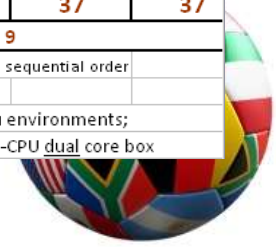
Size of the source data files (sequential text file with delimited columns or fix column width)

Amount of users per dedicated user group, who wants to access the application

The percentage of users of the user groups, who access the application at the same time (concurrent sessions)

Average time to wait (top level and detail level) for the results in a diagram object with average complexity after a filter selection on a single CPU-box (e.g. measured during prototyping or just estimated)

How often in a minute a user clicks (selections)



QlikView

Real performance test case

- One of the largest retail banks in Sweden
- Customer analysis tool for 4000-6000 advisors
- Used daily

QlikView

Real performance test lab scenario

- The solution will be used by 4000 – 6000 advisors as their primary source for customer information.
- 2000 concurrent users.
- 1.4 Terabytes of data in the source system (data for multiple periods)
- 12M customers (rows) and >600 columns (one period/month snapshot)

QlikView

Usage assumptions

- Assume that there are 4000 users, and ...
- ... each user handles 2 customers per hour, and ...
- ... for each customer, each user makes 10 selections
- Then an average of 80.000 selections per hour or ~1300 selections per minute will be made

- Assume that the test shows that a core (processor) can handle 100 selections per minute, and ...
- ... the server has 8 cores
- Then a server can theoretically deliver 800 selections per minute

QlikView

Uncertainties

- We can test many users and many selections
- We can measure average CPU time per selection
- We cannot predict user behavior:
  - How many selections per minute does a user make?
- We cannot predict variations in customer behavior:
- Do the customers call in the morning rather than right after lunch? So that we have peak hours with much higher server load?

QlikView

Performance lab environment

- Hardware
  - Server hardware specification:
    Processor: Intel Xenon CPU X7460 @ 2.66GHz (4 procs, 24 cores) (Pentryn)
    RAM: 256GB
- Number of QlikView documents
  - 60 “Publisher split”-documents
- Size of data
  - 37 000 – 116 000 customer records

- Number of concurrent users;
- User click-behaviour in terms of number of selections (queries) per hour and user authorization
- Real data supplied by the customer

QlikView

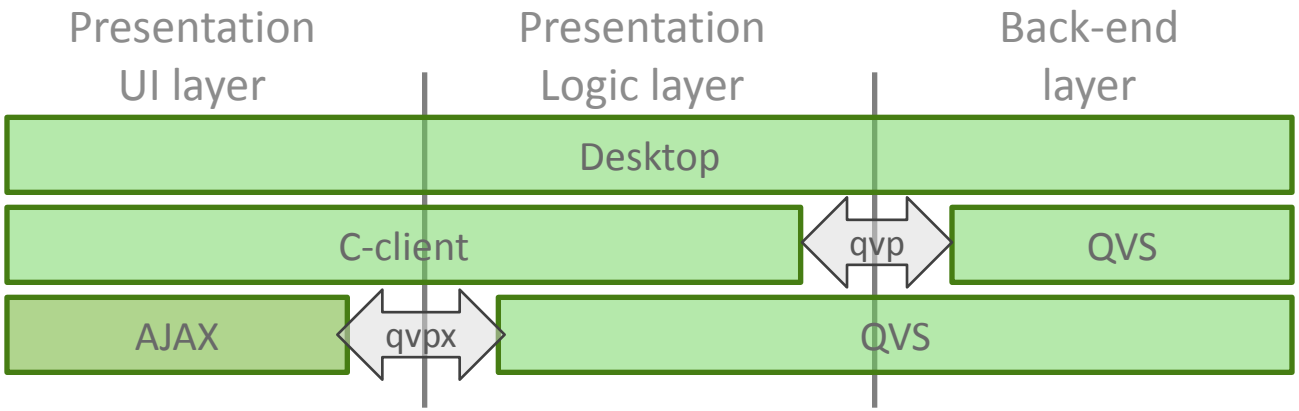
What to test?

- Number of selections/clicks/queries per minute
- Memory consumption
- CPU usage
- Network bandwidth consumption
- Response times

QlikView

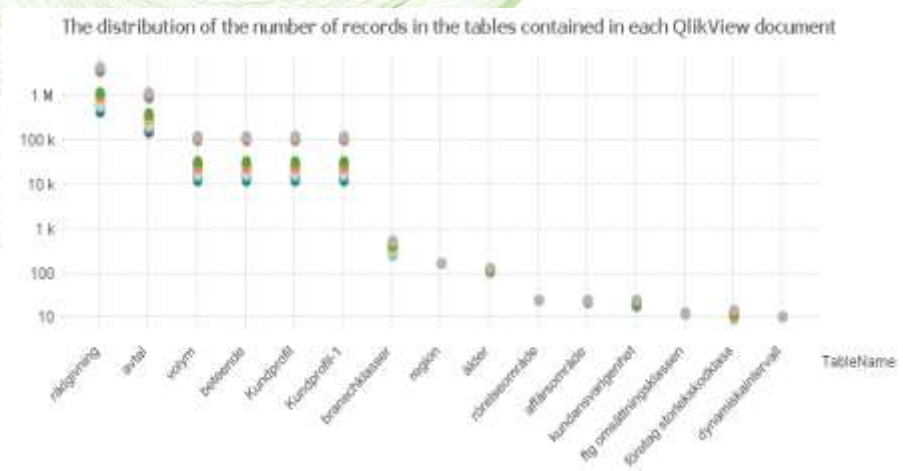
Two clients:

- Using the C client (the fat client), testing
  - Logical inference and calculations on server
- Using the AJAX client (the thin client), testing
  - Logical inference and calculations on server
  - Graph generation
  - XML generation
  - Web server efficiency
  - HTML compression



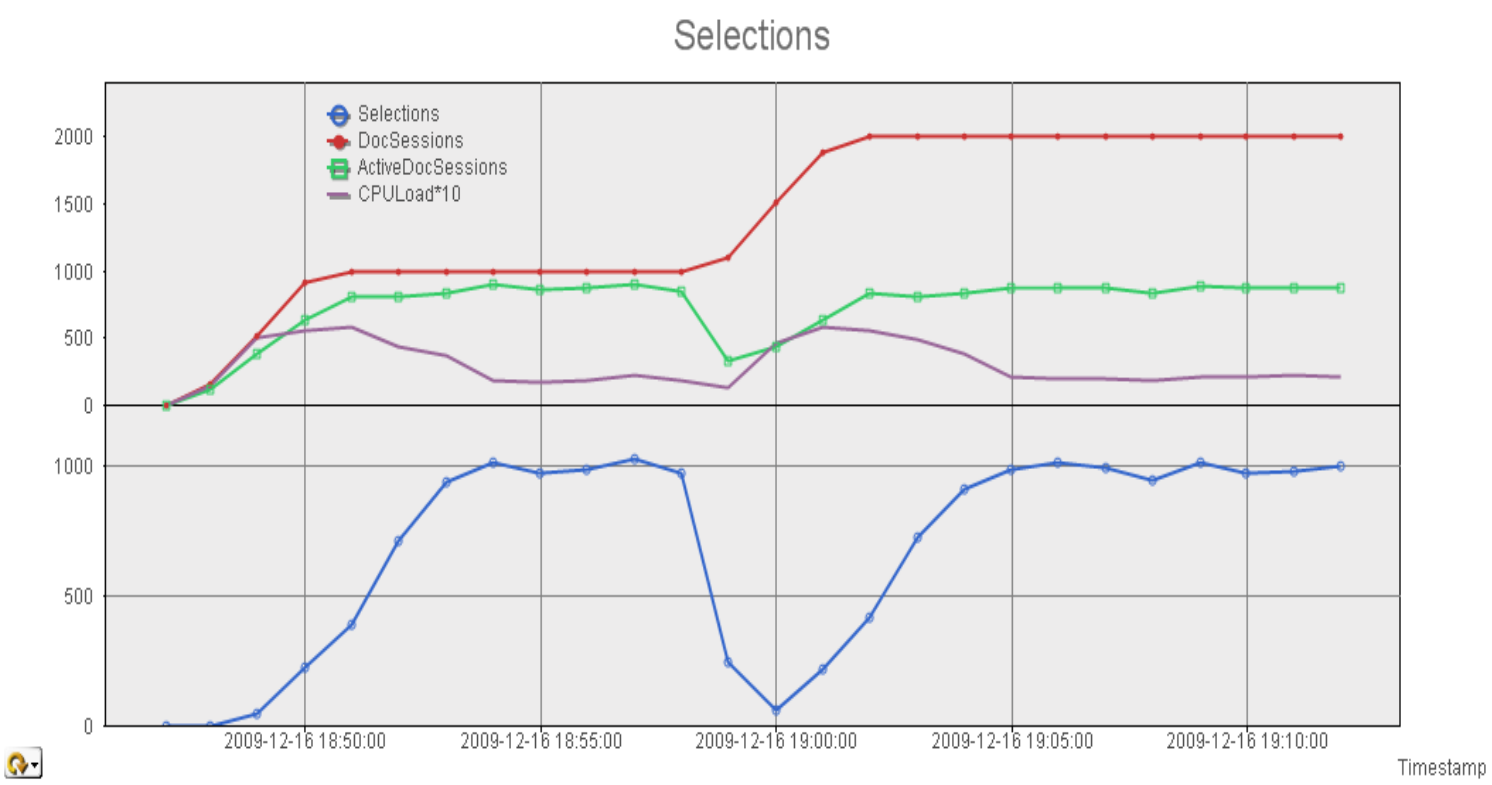
QlikView

Number of records and GUI



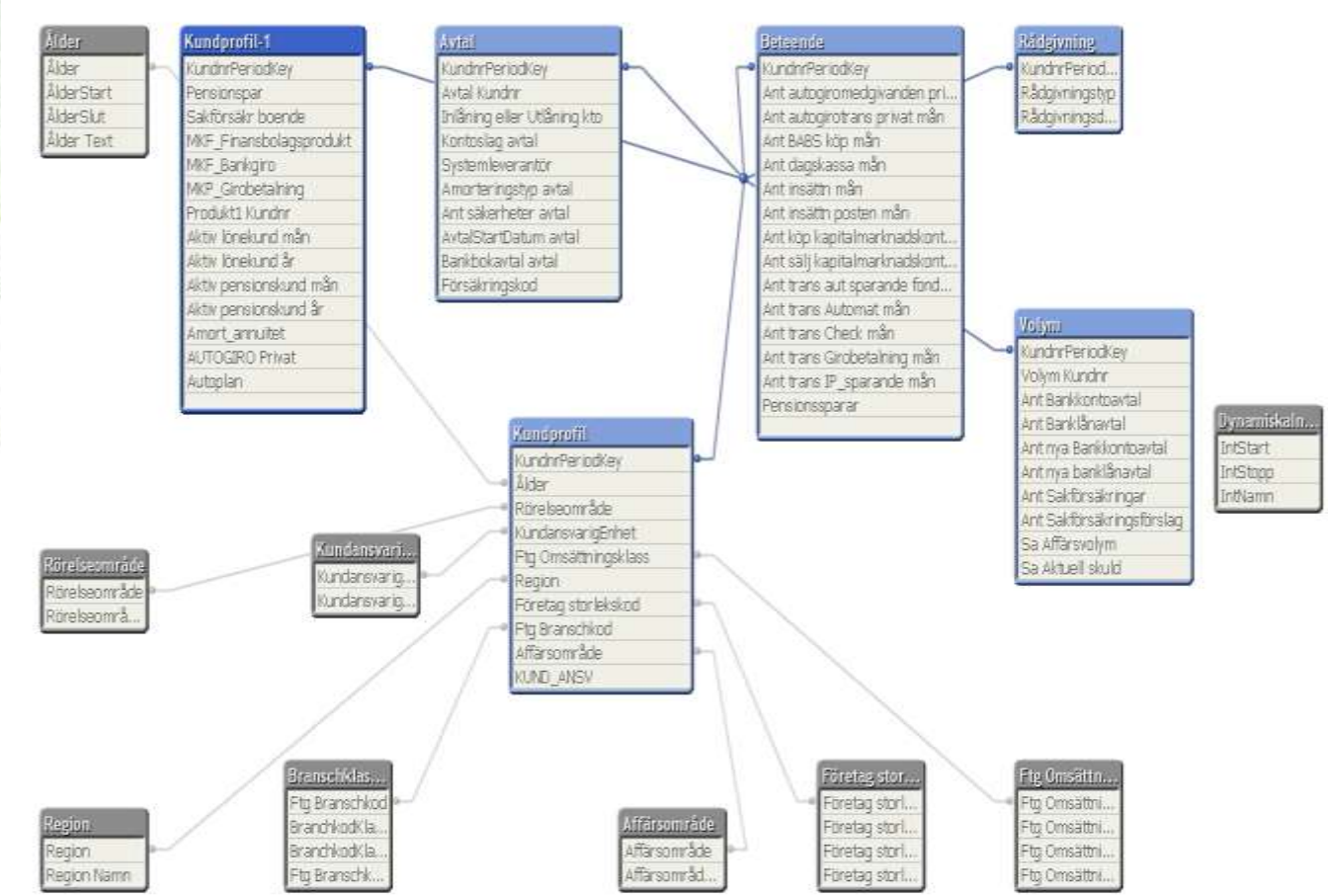
QlikView

The QlikView Server log



QlikView

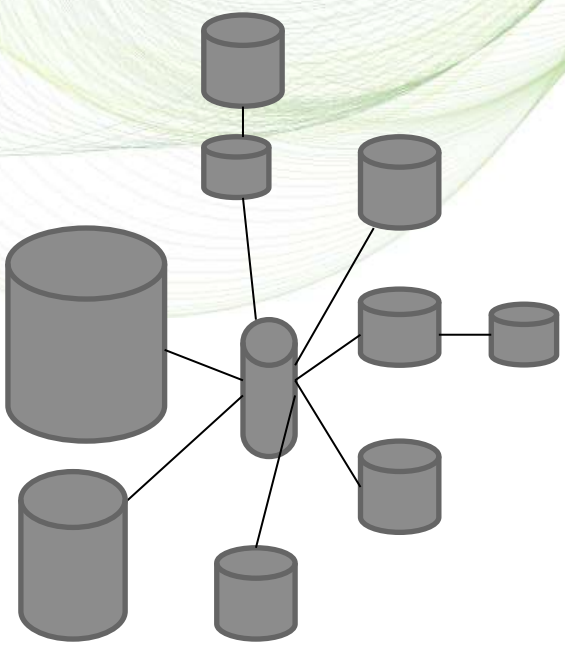
The data model



QlikView

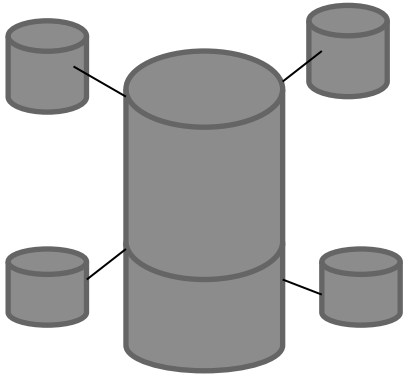
Data Model: Alternatives for data models with multiple facts

Snowflake Scheme



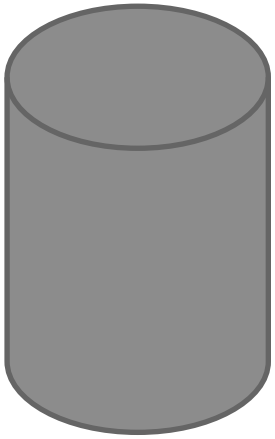
Snowflake with a link table between dim tables and fact tables

Star schema



Simple star schema
concatenated facts tables
de-normalized dimensions

Big Flat Table



“Big” table with concatenated fact tables
joined dimension tables

QlikView

Data Alternative: Fact Table Concatenation

Region	Product	Date	Sales
Region A	P1	2009-01-31	100
Region A	P1	2009-02-28	120
Region A	P1	2009-03-31	140
Region A	P2	2009-01-31	500
Region A	P2	2009-02-28	550
Region A	P2	2009-03-31	600
Region B	P1	2009-01-31	50
Region B	P1	2009-02-28	55
Region B	P1	2009-03-31	60
Region B	P2	2009-01-31	200
Region B	P2	2009-02-28	180
Region B	P2	2009-03-31	160

Sales

Plan Yearly

Region	Date	Plan
Region A	2009-01-1	8000
Region B	2009-01-1	10000

Procurement Cost

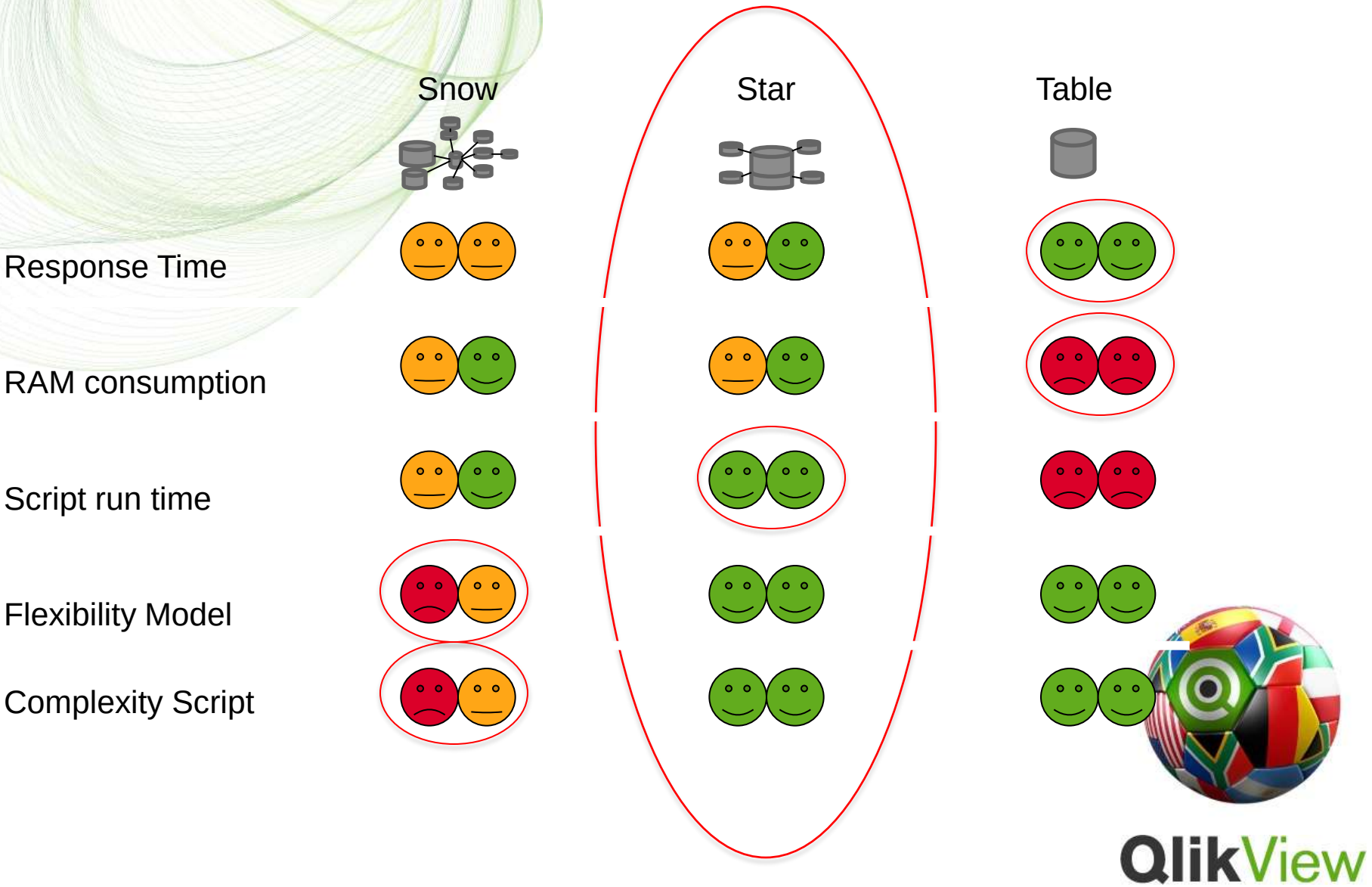
Product	Date	Cost
P1	2009-01-31	130
P1	2009-02-28	1400
P1	2009-03-31	1600
P2	2009-01-31	500
P2	2009-02-28	650
P2	2009-03-31	600

Region	Product	Date	Sales	Plan	Cost
Region A	P1	2009-01-31	100		
Region A	P1	2009-02-28	120		
Region A	P1	2009-03-31	140		
Region A	P2	2009-01-31	500		
Region A	P2	2009-02-28	550		
Region A	P2	2009-03-31	600		
Region B	P1	2009-01-31	50		
Region B	P1	2009-02-28	55		
Region B	P1	2009-03-31	60		
Region B	P2	2009-01-31	200		
Region B	P2	2009-02-28	180		
Region B	P2	2009-03-31	160		
Region A		2009-01-1		8000	
Region B		2009-01-1		10000	
	P1	2009-01-31			130
	P1	2009-02-28			1400
	P1	2009-03-31			1600
	P2	2009-01-31			500
	P2	2009-02-28			650
	P2	2009-03-31			600

Concatenated Facts

QlikView

Data Model: Implications on …



QlikView

Cockpit Applications (Dashboards) and large Datavolumes

- The following 4 methods can be used to solve this issue:

1. Creation of an aggregated fact table within the model (connected or data island)
2. QV Document Chaining (aggregates and details) and filter transfer
3. Pre-filling of the object cache after data load (via VBS)
4. Remove all unused fields from the data model

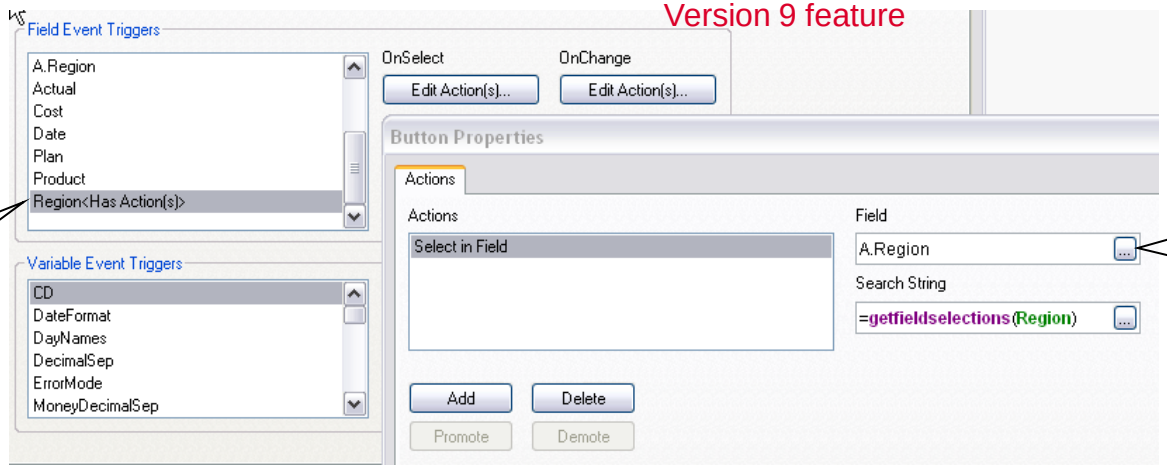
QlikView

Method 1: Aggregated Fact Table

- Necessary in script for creating the aggregated fact table:

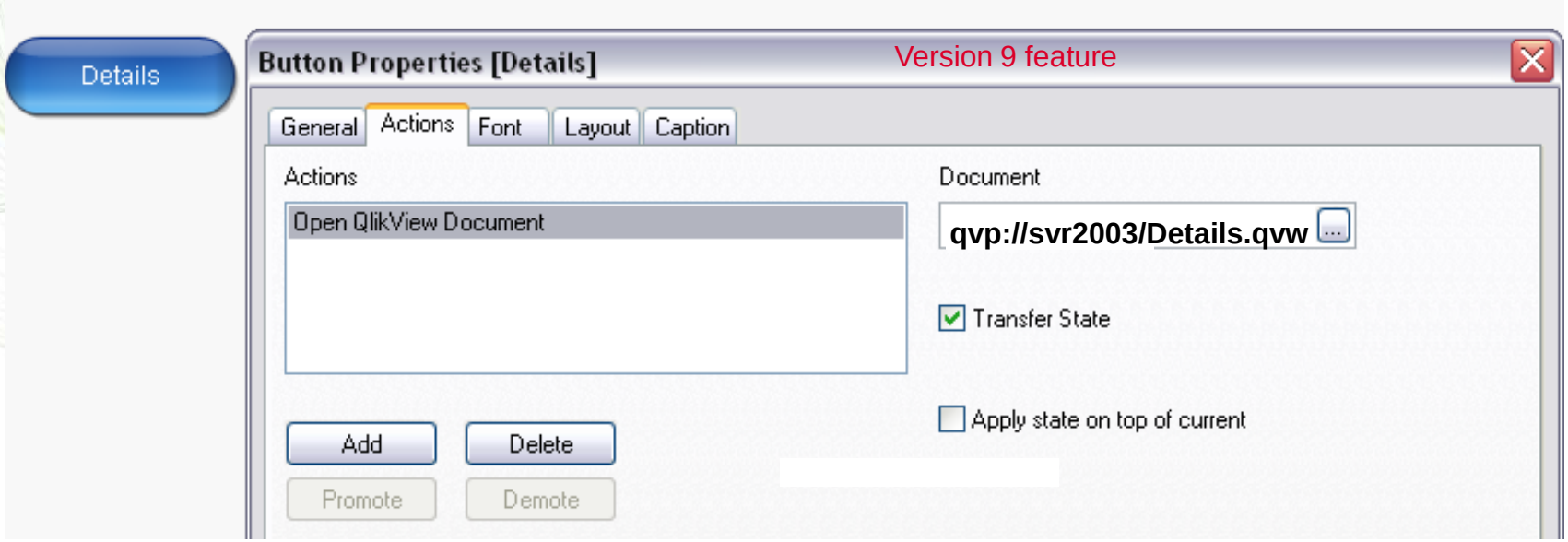
```
Select
    … as … ,
    sum(…) as …
resident group by …;
```

- Transfer Field selections via triggered Actions using the QV GUI when the aggregated Fact table is a data island



QlikView

Method 2: Document Chaining



Limitation: Only applications on the same server can be opened

QlikView

Method 3: Prefilling of Object Cache

```
set x = CreateObject("QlikTech.QlikView")
set doc = x.OpenDoc("qvp://BISERVER/xyz/Value_Management_Dashboard.qvw")

loop_through_objects(doc)
doc.CloseDoc
x.Quit

'I run through the application. Called by IOpenTheDocument
Sub loop_through_objects(doc)

  For i = 0 to doc.NoOfSheets - 1

      doc.ActivateSheet i
      ' Selection-loop thru the field Region
      Set val=doc.Fields("Region").GetOptionalValues
      For y=0 to val.Count-1

       INH=val.Item(y).Text
       doc.Fields("Region").select INH

       Objects = doc.ActiveSheet.GetSheetObjects

       For j = lBound(Objects) To uBound(Objects)
          set table = Objects(j)
          select case Objects(j).getObjectType
          case 1,4,10,11,12,13,14,16,20,21,27
           On Error Resume Next
            CellRect = doc.GetApplication().GetEmptyRect()
           CellRect.Top = 0
           CellRect.Left = 0
           CellRect.Width = table.GetColumnCount
           CellRect.Height = table.GetRowCount
           set CellMatrix = table.GetCells( CellRect )
           For RowIter = 0 to CellMatrix.Count-1
           Next
          End Select
       Next
      Next
      doc.Fields("Region").clear
  Next

End Sub
```

This VBS runs through all folders of the application, opens all diagrams and selects all fields in the dimension Region

Requirements:

- The enterprise control panel setting: Cache should be > 20%
- At the computer where the VBS is active the QV-Plugin client must be installed

Alternative Call:
set doc = x.OpenDoc("qvp://BISERVER/xyz/Value_Management_Dashboard.qvw?IIS_AUTHENTICATE")

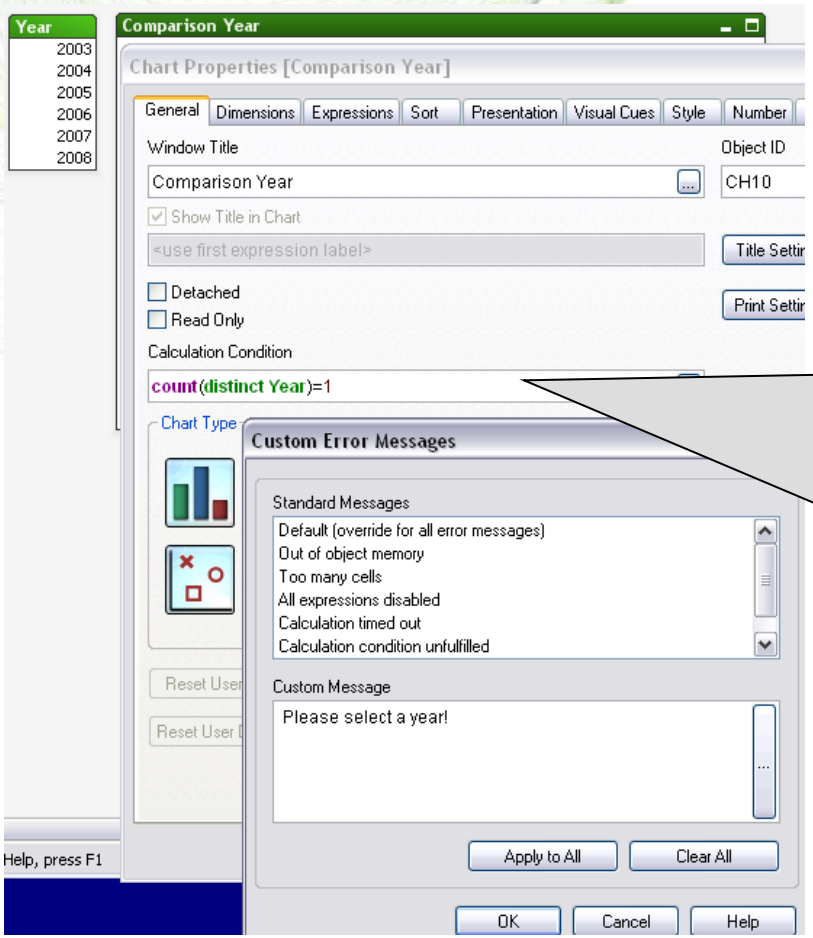
QlikView

Method 4: Remove Unused Fields from Data Model

- **This measure has an indirect impact on performance, e.g. application loading is accelerated, more space for cache, better handling of the application during update, clarity of model …**

QlikView

GUI Design: Conditional Calculations on GUI



Condition examples:

- Count(distinct *dim-field*)>1
- not isnull(only(*dim-field*))
- getselectedcount (*dim-field*) >0
- vSelectDim1 and vSelectDim2
  where vSelectDim1 = not isnull(only(dim-field1))
  and vSelectDim2 = not isnull(only(dim-field2))

GUI Design: Usage of right formulas for time calculations

Following you see 5 examples of how to calculate (“Sales last year month”):

sum({<Date={"$(vPYMonthStart)"}>}Sales)

sum({<Date={"$(vPYMonthStart)"}>}Sales)

- vPYMonthStart = date(floor(monthstart(%Date,-12)))
- vPYMonthEnd = date(floor(monthend(%Date,-12)))

QlikView

GUI Design: Usage of right formulas for time calculations

Here is the overview of formulas to calculate (“Sales last year month”):

	8.5	9
sum(if(inmonth (Date,date(max(total %Date)),-12), Sales))	18	
sum(inmonth (Date,date(max(total %Date)),-12) * -1 * Sales)	15	
sum(if(inmonth (Date, vPYMonthEnd,0), Sales))	18	
sum(if(Date>= vPYMonthStart and Date <= vPYMonthEnd, Sales))	1	
sum({$<Date={">=$(vPYMonthStart) <=$(vPYMonthEnd)"}>} Sales)	5	

- vPYMonthStart = date(floor(monthstart(%Date,-12)))
- vPYMonthEnd = date(floor(monthend(%Date,-12)))

QlikView

Further Recommendations

- Split timestamp into date and time fields when date and time is needed
- Remove time from date by floor() or by date(date#(..)) when time is not needed
- Reduce wide concatenated key fields via autonumber(), when all related tables are processed in one script
- Use numeric fields in logical functions (string comparisons are slower)
- (a – b) / b better: (a / b) – 1
- date(max(SDATE,'DD.MM.YYYY')) is factor xxx faster than max(date(SDATE,’DD.MM.YYYY’))
- Is the granularity of the source data needed for analysis?
  - “sum() group by”

QlikView

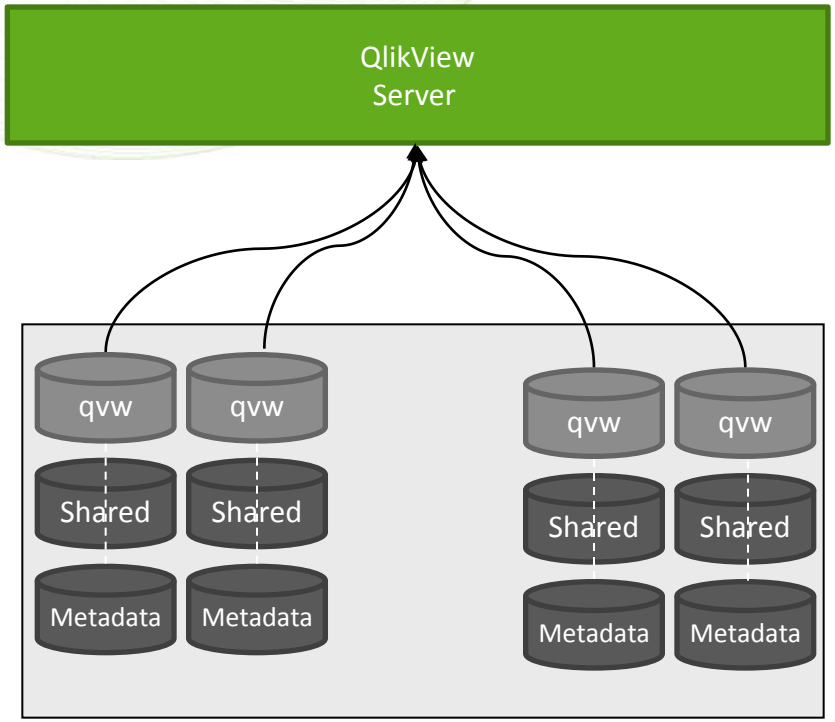
Further Recommendations

- Use numeric flags (e.g. with 1 or 0) , which are pre-calculated in the script
- Reduce the amount of open chart objects
- Calculate measures within the script (model size <> online performance)
- Limit the amount of expressions within chart/pivot objects, distribute them in multiple objects (use auto minimize)
- Be very careful using Macros!

QlikView

1º QlikDev Latino América

Hardware: Scalability – Single Server



The server:

- Simple administration
- One QVS license
- Scalibility is depended on max. CPUs / RAM per box

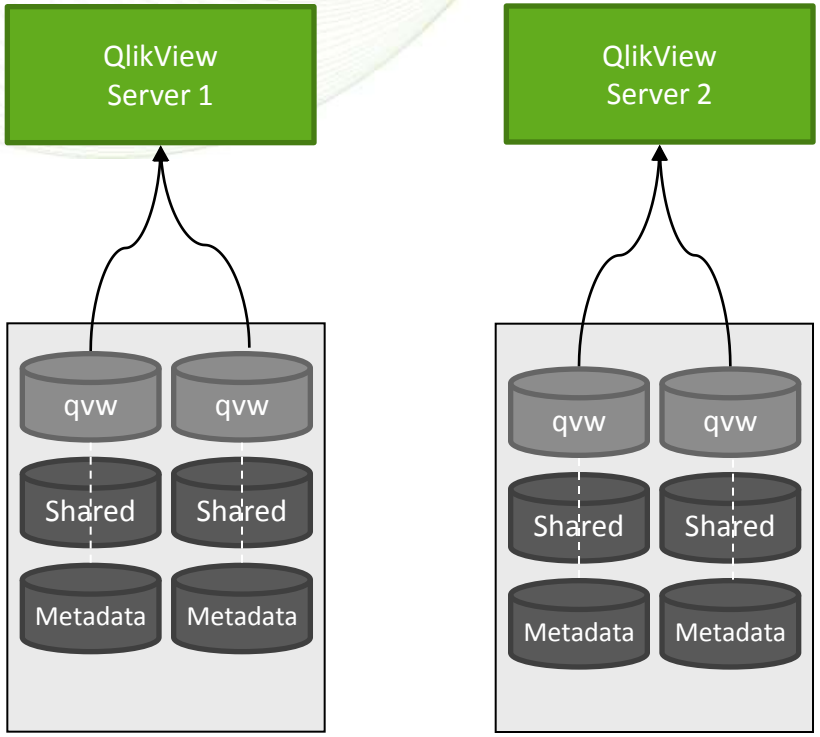
Biggest configuration at a customer early 2009 is:

8 x QuadCore (AMD-CPUs)

512 GB RAM

QlikView

Hardware: Scalability – Multiple Server

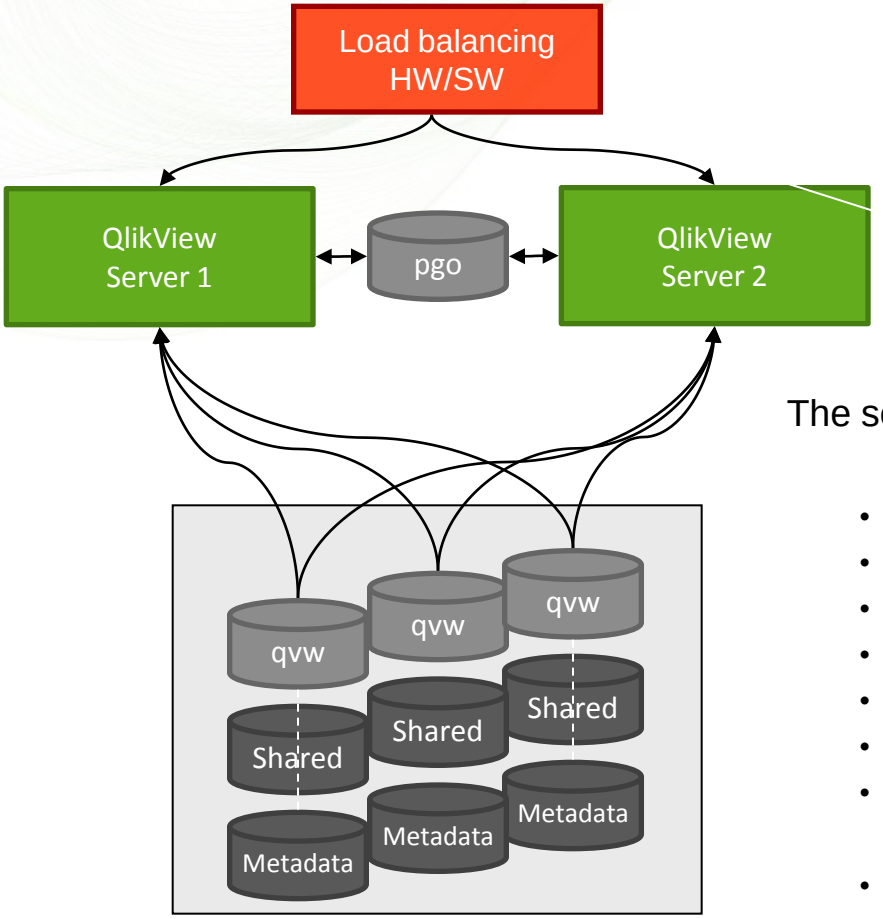


The servers:

- Request must be targeted to a specific server
- Have different IP-addresses
- Have different license numbers
- Have different share-files
- No communication between
- Load different QVWs (normally)
- Can be administered by the same QVP-instance
- Can have the same QV-accesspoint

QlikView

Hardware: Scalability – Clustering



Different Methods:

- Simple IP-address sharing via a special router
- IP address sharing per software settings (IP forwarding)
- Load balancing software
- Intelligent HW devices (e.g. BIG-IP load balancing)
- Intelligent QVP-Accesspoint (V9)

The servers:

- Request can't be targeted to a specific server
- Have different IP-addresses
- Have the same license number
- Use the same share-file
- Synchronise the settings
- Share the QVWs and shared objects
- Store information (CAL's, etc) in a pgo-file
- Amount of servers is unlimited

QlikView

Hardware: Scalability – Comparing Scalability Scenarios

QlikView
1 big Server

Advantage:

Can host very big QV model(s)
Seamless performance
Easy configuration/administration
Low SW costs (QV license)

Disadvantage:

No failover (complete QV stops)
Limited scalability
(max. CPU/RAM per box)

- **Big or very big QV models**
- **Small / medium amount of users**

QlikView Cluster | **QlikView Cluster**

Advantage:

Scalable
Multiple inexpensiv 2-way servers
Seamless performance (dep. on LB HW)
Failover (dep. on LB HW)

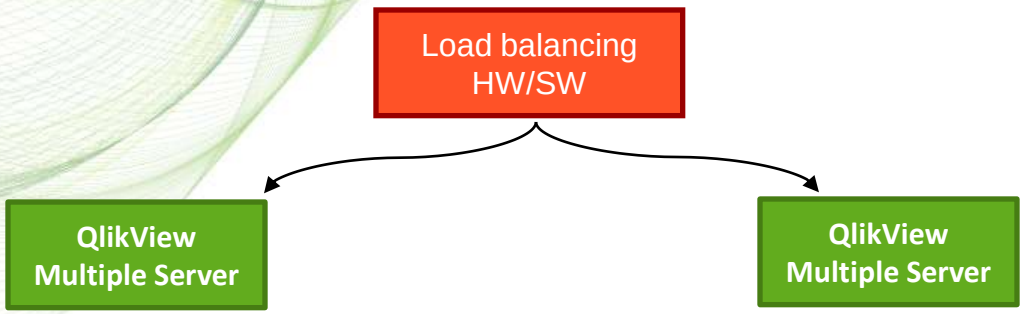
Disadvantage:

Higher SW cost (several QVS/QVP licenses)
Can't host very big QV models
Redundant models in server RAM

- **Small / medium QV models**
- **Large amount of users**

QlikView

Hardware: The CPU-time battle



Operational QlikView applications

- Quick response times
- Small apps
- Many users

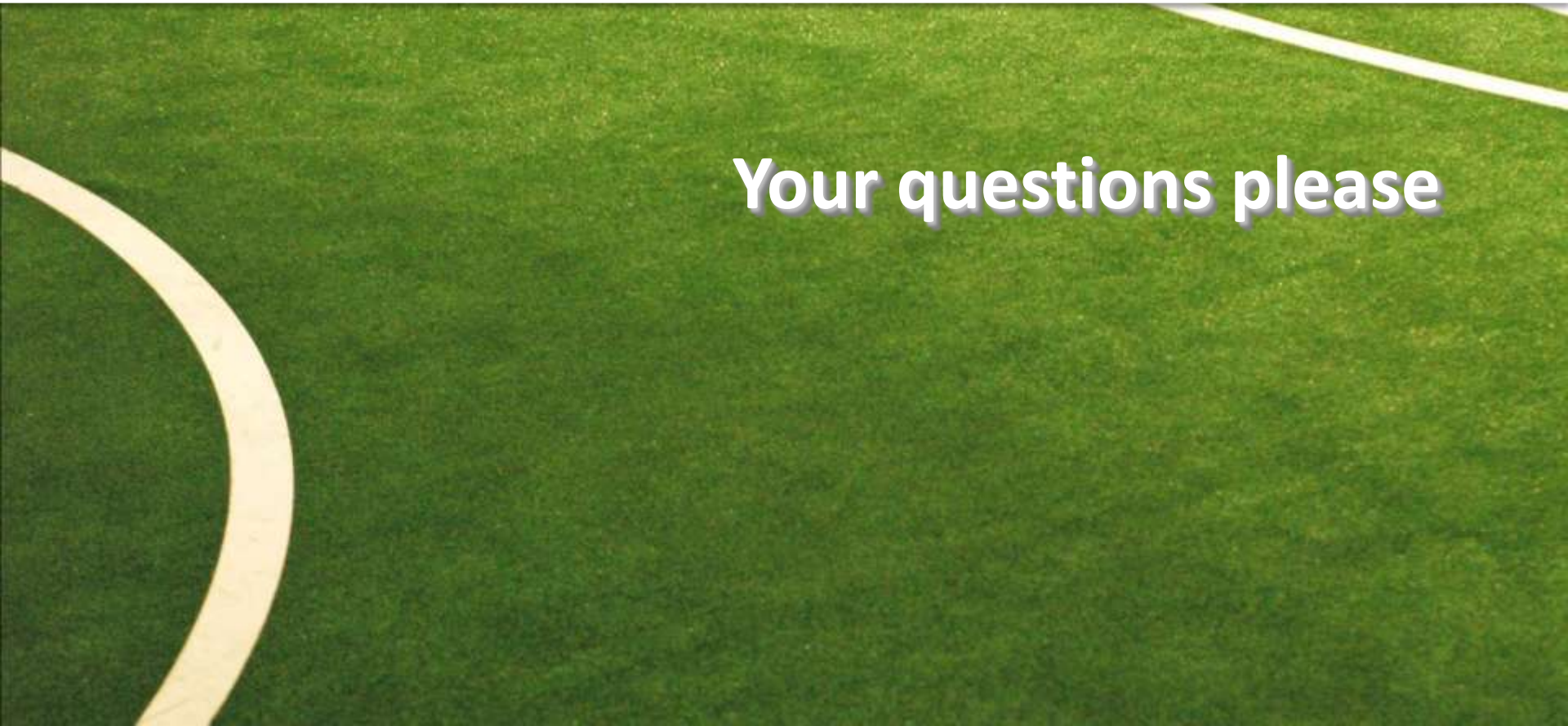
Analytical QlikView applications

- Longer response times
- Larger apps
- Less users

A heavy analytical application which take a lot of CPU-time can hog the system for the operational users.

QlikView

1º QlikDev Latino América



Thank You! Gracias! Obrigado!

1º QlikDev
Latino América

QlikView