

CLARITY FROM CHAOS



Optimizing QlikView

Simon Mills

Optimizing QlikView

IoD, London City, Liverpool St

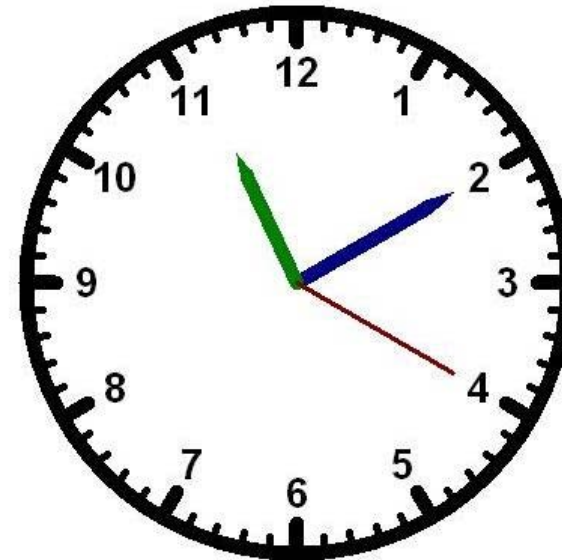
Wednesday 7th October 2009

Why Optimize?

Why Optimize?

Availability

Is the report available at the right time?



Why Optimize?

Slow 'Qlik' Response

**Is the user waiting
too long?**



Why Optimize?

Efficient use of resources

**Make the accountants
happy**



Availability – Time to ETL

Availability – Time to ETL

- Use QVD files : super quick performance

Availability – Time to ETL

- Use QVD files : super quick performance
- Create parallel processes

Availability – Time to ETL

- Use QVD files : super quick performance
- Create parallel processes
- Incremental loads : don't re-extract source data

Availability – Time to ETL

- Use QVD files : super quick performance
- Create parallel processes
- Incremental loads : don't re-extract source data
- Only extract & process data required in the final report

Availability – Time to ETL

- Use QVD files : super quick performance
- Create parallel processes
- Incremental loads : don't re-extract source data
- Only extract & process data required in the final report
- Optimize the script

'Qlik' Response Time

'Qlik' Response Time

- Add more CPUs : limit user contention

'Qlik' Response Time

- Add more CPUs : limit user contention
- Reduce the RAM required : remove complex keys

'Qlik' Response Time

- Add more CPUs : limit user contention
- Reduce the RAM required : remove complex keys
- Cut the number of expressions calculated with each 'Qlik'

'Qlik' Response Time

- Add more CPUs : limit user contention
- Reduce the RAM required : remove complex keys
- Cut the number of expressions calculated with each 'Qlik'
- Simplify the expressions

'Qlik' Response Time

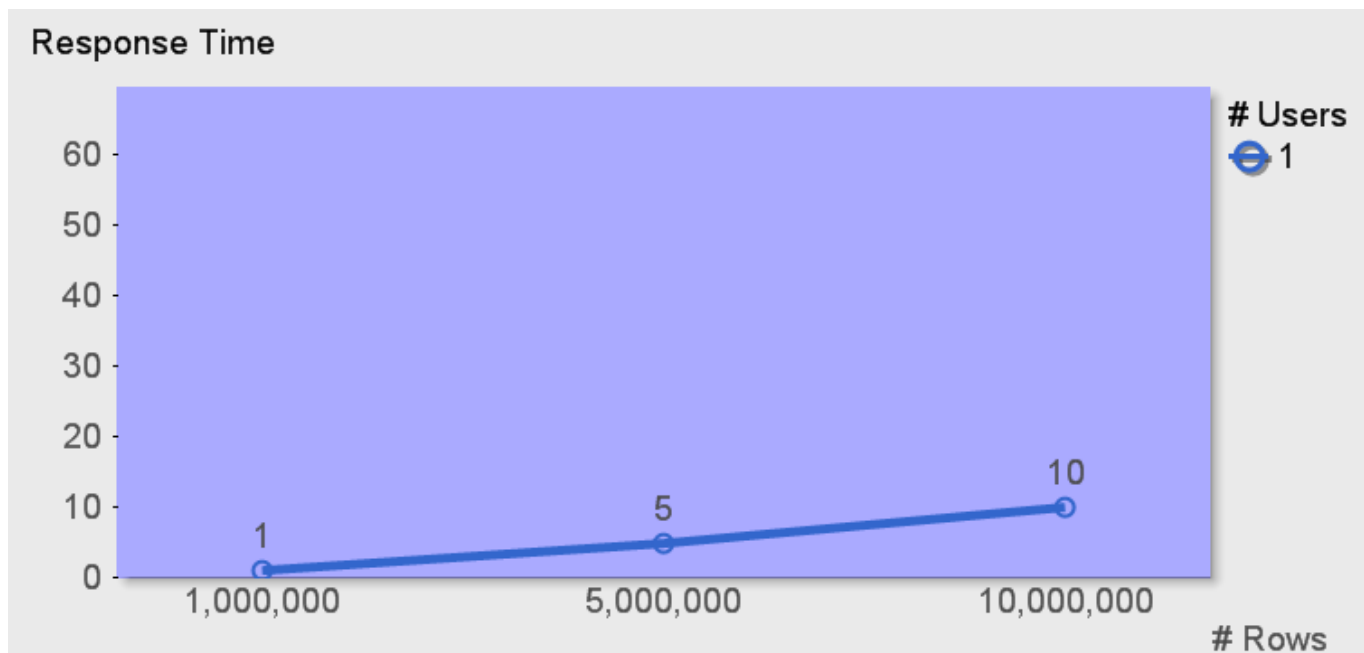
- Add more CPUs : limit user contention
- Reduce the RAM required : remove complex keys
- Cut the number of expressions calculated with each 'Qlik'
- Simplify the expressions
- Cache common chart results

'Qlik' Response Time

- Add more CPUs : limit user contention
- Reduce the RAM required : remove complex keys
- Cut the number of expressions calculated with each 'Qlik'
- Simplify the expressions
- Cache common chart results
- Reduce the number of rows of data

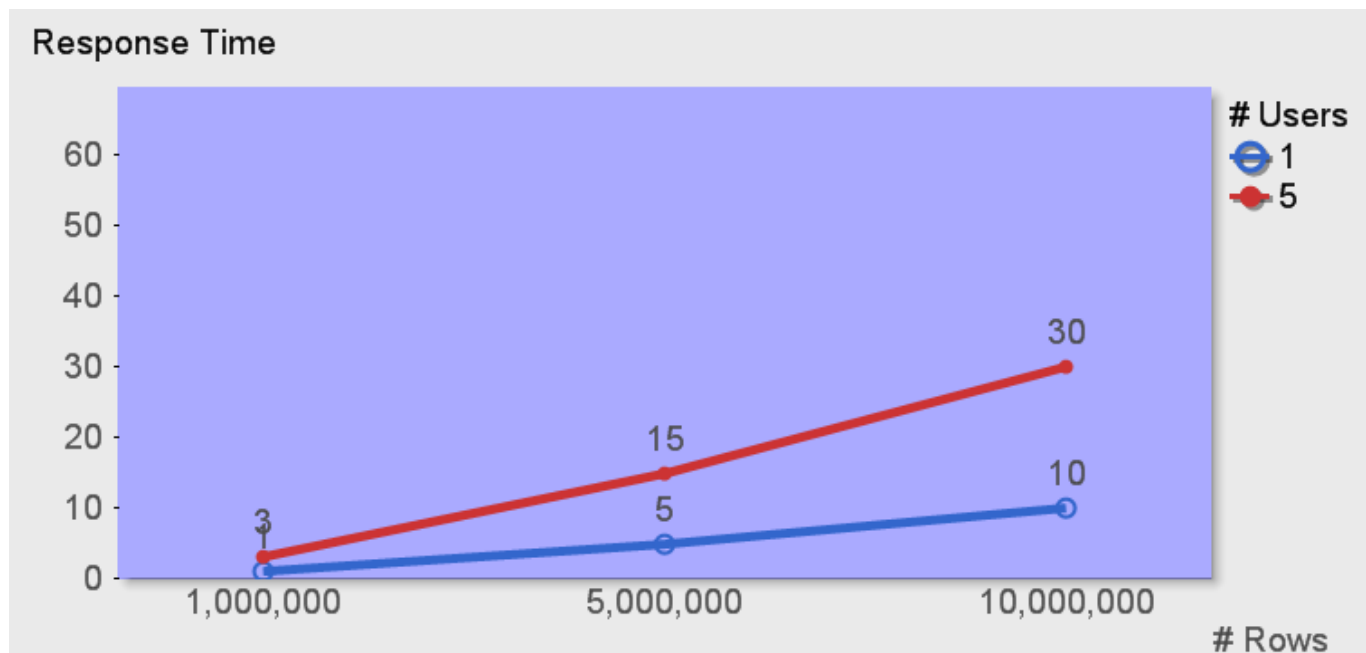
Optimizing QlikView

Rows & 'Qlik' Response Time



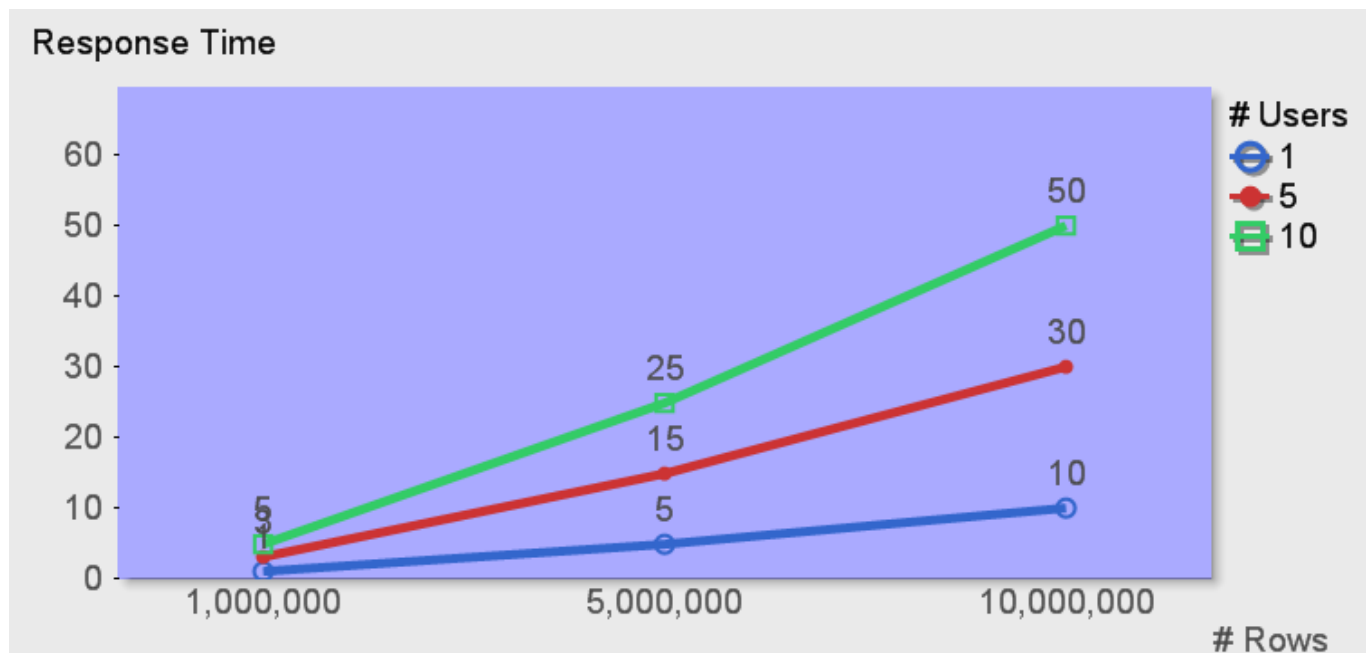
Optimizing QlikView

Rows & 'Qlik' Response Time



Optimizing QlikView

Rows & 'Qlik' Response Time



Reduce the # of Rows

- Create an additional report containing aggregated data

Reduce the # of Rows

- Create an additional report containing aggregated data
 - This can combine detail and aggregated data in the same report

Reduce the # of Rows

- Create an additional report containing aggregated data
 - This can combine detail and aggregated data in the same report
- Or use Dynamic Data Reduction

Conclusions

Conclusions

- Plan for parallel ETL processes from the start

Conclusions

- Plan for parallel ETL processes from the start
- Check your expressions work with all the data

Conclusions

- Plan for parallel ETL processes from the start
- Check your expressions work with all the data
- Understand your user base and the likelihood of contention

Conclusions

- Plan for parallel ETL processes from the start
- Check your expressions work with all the data
- Understand your user base and the likelihood of contention
- Budget for optimization in the project from the beginning

Conclusions

- Plan for parallel ETL processes from the start
- Check your expressions work with all the data
- Understand your user base and the likelihood of contention
- Budget for optimization in the project from the beginning
- Buy the biggest server you can

Optimizing QlikView

Thank you