

Ticket Distribution, Benchmarking and Revenue Computation for a Public Transport Company

Gustavo Martins - 2010131414

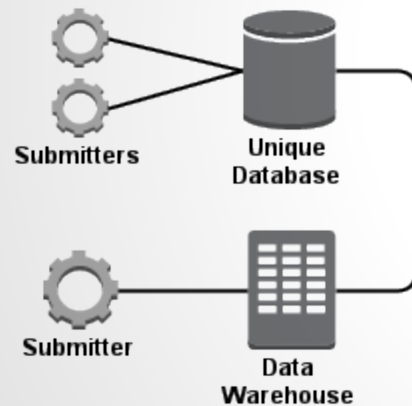
João Valença - 2010130607

Objectives

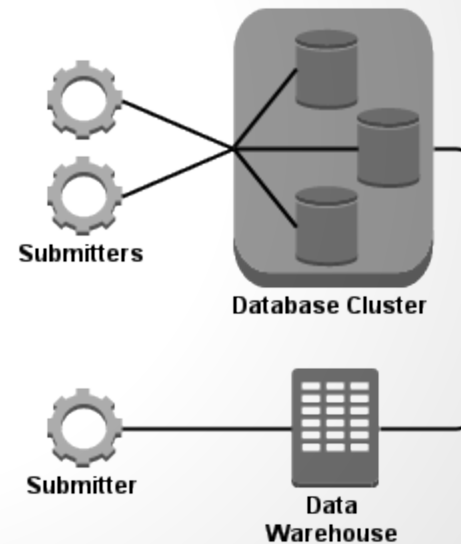
- Create a Benchmark for Ticketing Systems
- Suggest an approach to compute the Revenue distribution
- Compare different storage devices
- Compare Standalone and Scalable architectures
- Prove that both architectures can be used
- Prove that the system can be scalable

System Architecture

- Standalone

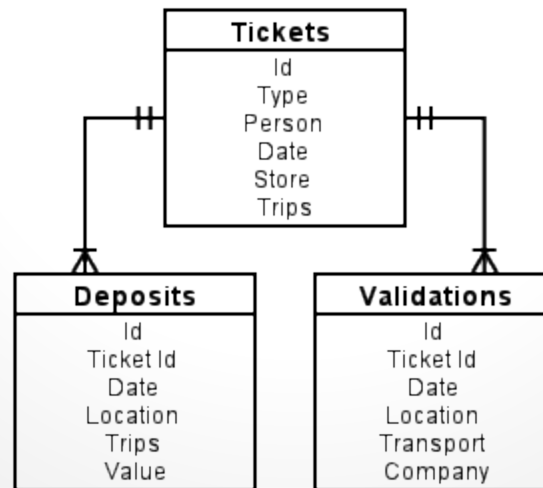


- Scalable



Database Structure

- **Tickets** → Store information about the ticket and its sale
- **Deposits** → Store trip deposits made on a ticket
- **Validations** → Store validations made with a ticket on public transports



Case Study

- Generation

- Ticket Generation
 - Create random information to insert into the Tickets table
- Stress Test
 - Multiple processes inserting previously generated information into the Tickets table

- Benchmark

- Queries
 - Execute business context queries using different table combinations to the database
- Revenue Computation
 - Algorithm retrieving validations from the database and processing them

Experimental Setup

- Standalone

	gmlt	jvub
OS	Arch Linux 3.14.4	Arch Linux 3.14.4
	64 bits	64 bits
Disk	HDD	SSD
	5400 rpm	-
	75 MB/s (read)	300 MB/s (read)
RAM	4 GB	8 GB
	1333 MHz	1600 MHz
CPU	i7 Quad-Core	i7 Dual-Core
	1.6 GHz	2 GHz

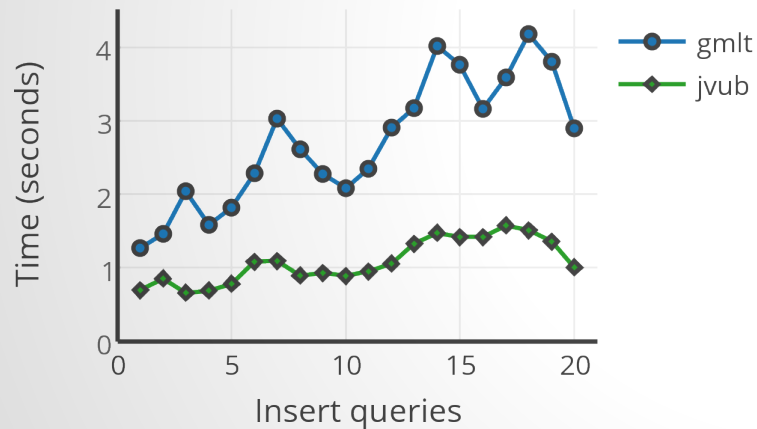
- Scalable

	Manag. Node	Storage Node
OS	CentOS 2.6.32	CentOS 2.6.32
	32 bits	32 bits
Disk	HDD	HDD
	7200 rpm	7200 rpm
	75 MB/s (read)	75 MB/s (read)
RAM	1.5 GB	1.8 GB
	667 MHz	667 MHz
CPU	Intel Celeron	Intel Celeron
	2 GHz	2 GHz

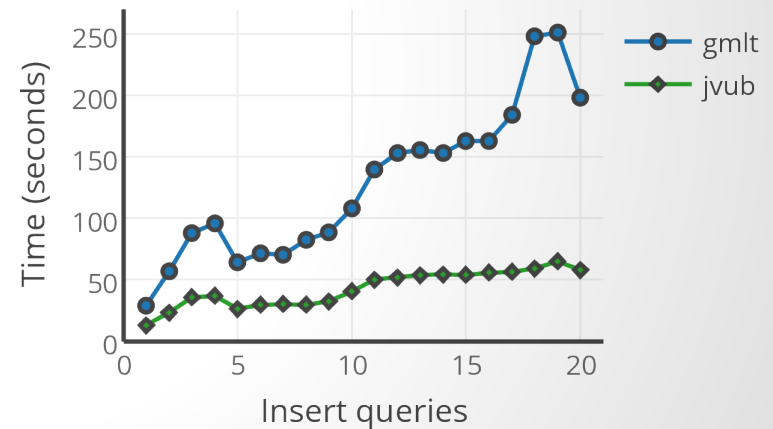
Experimental Results (Standalone)

- Stress Test

Deposits

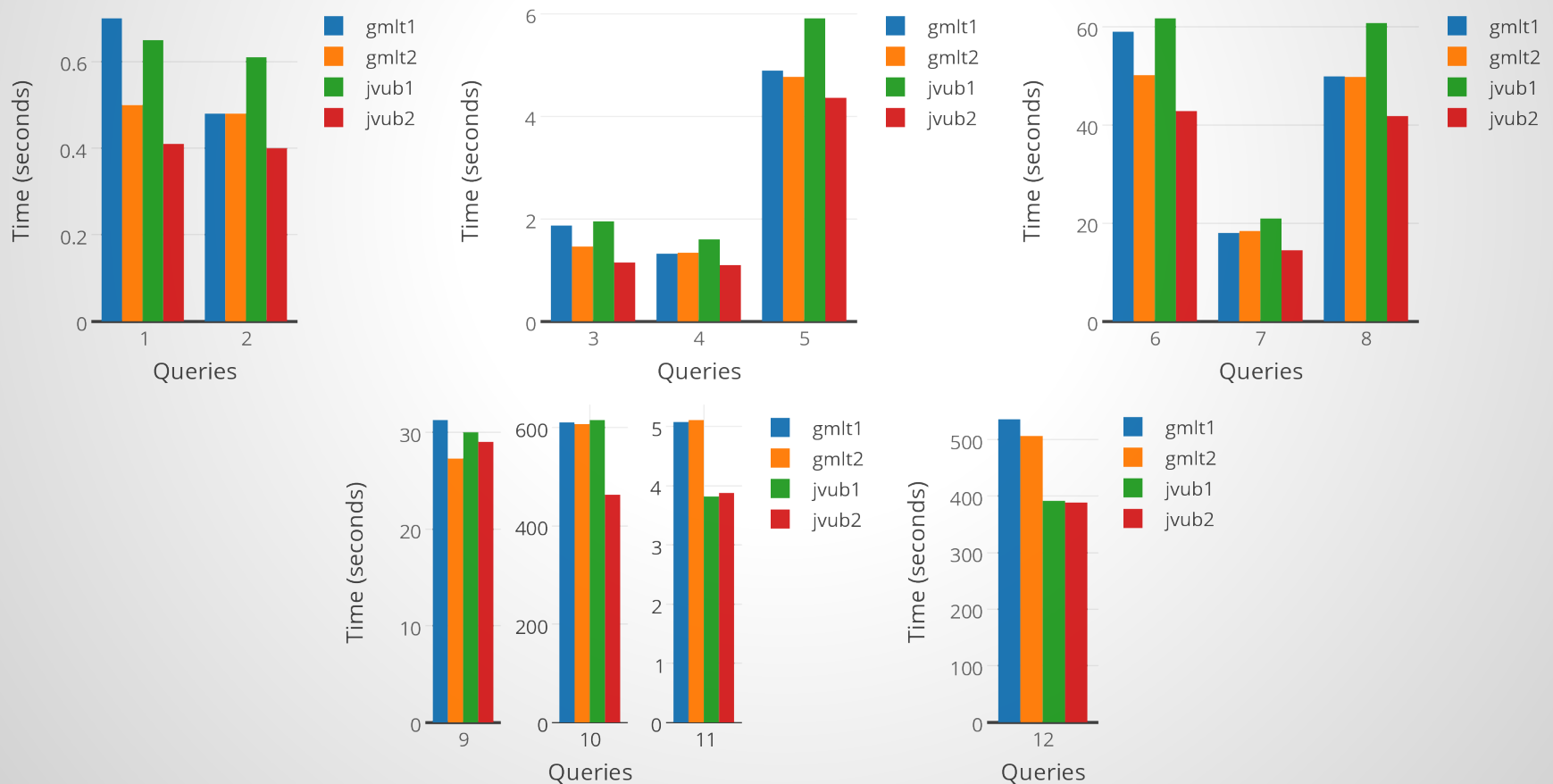


Validations



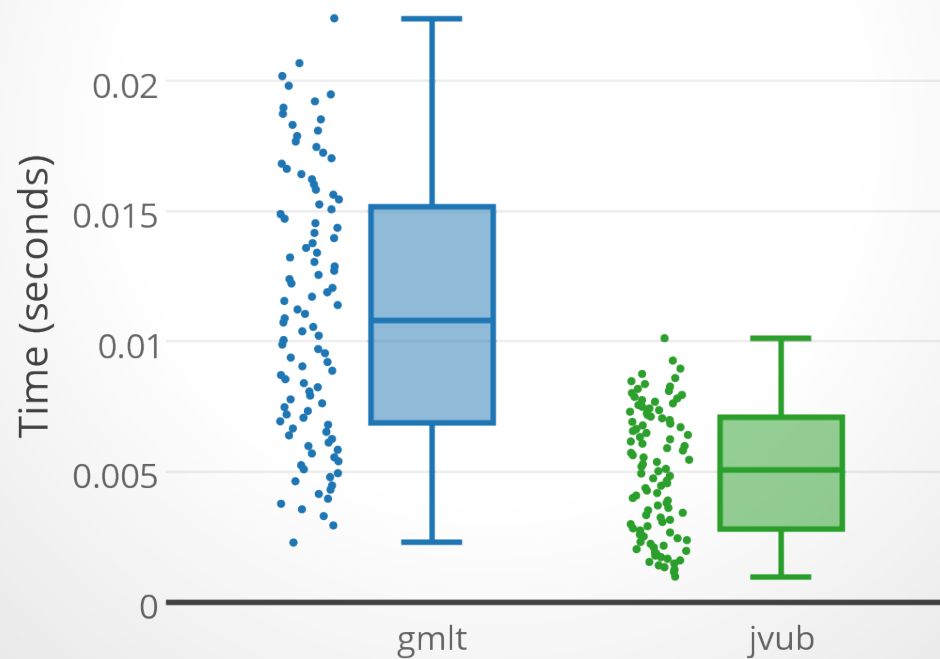
Experimental Results (standalone)

● Queries



Experimental Results (Standalone)

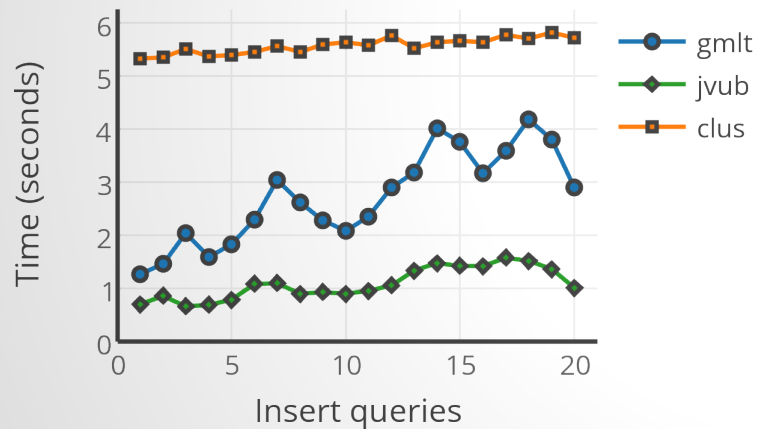
- Revenue Computation



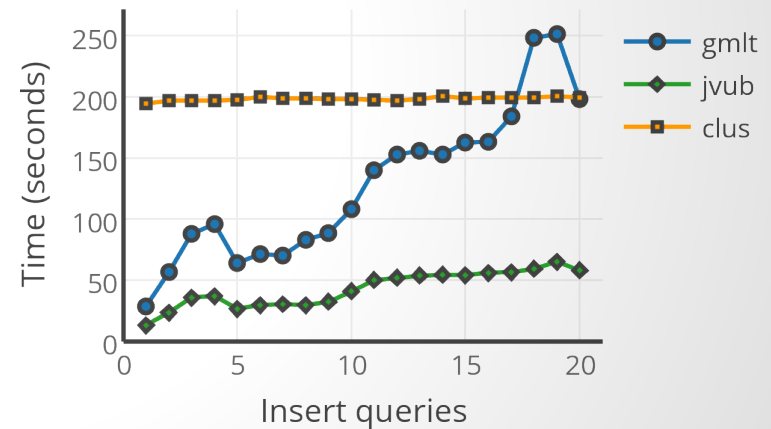
Experimental Results (Scalable)

- Stress Test

Deposits

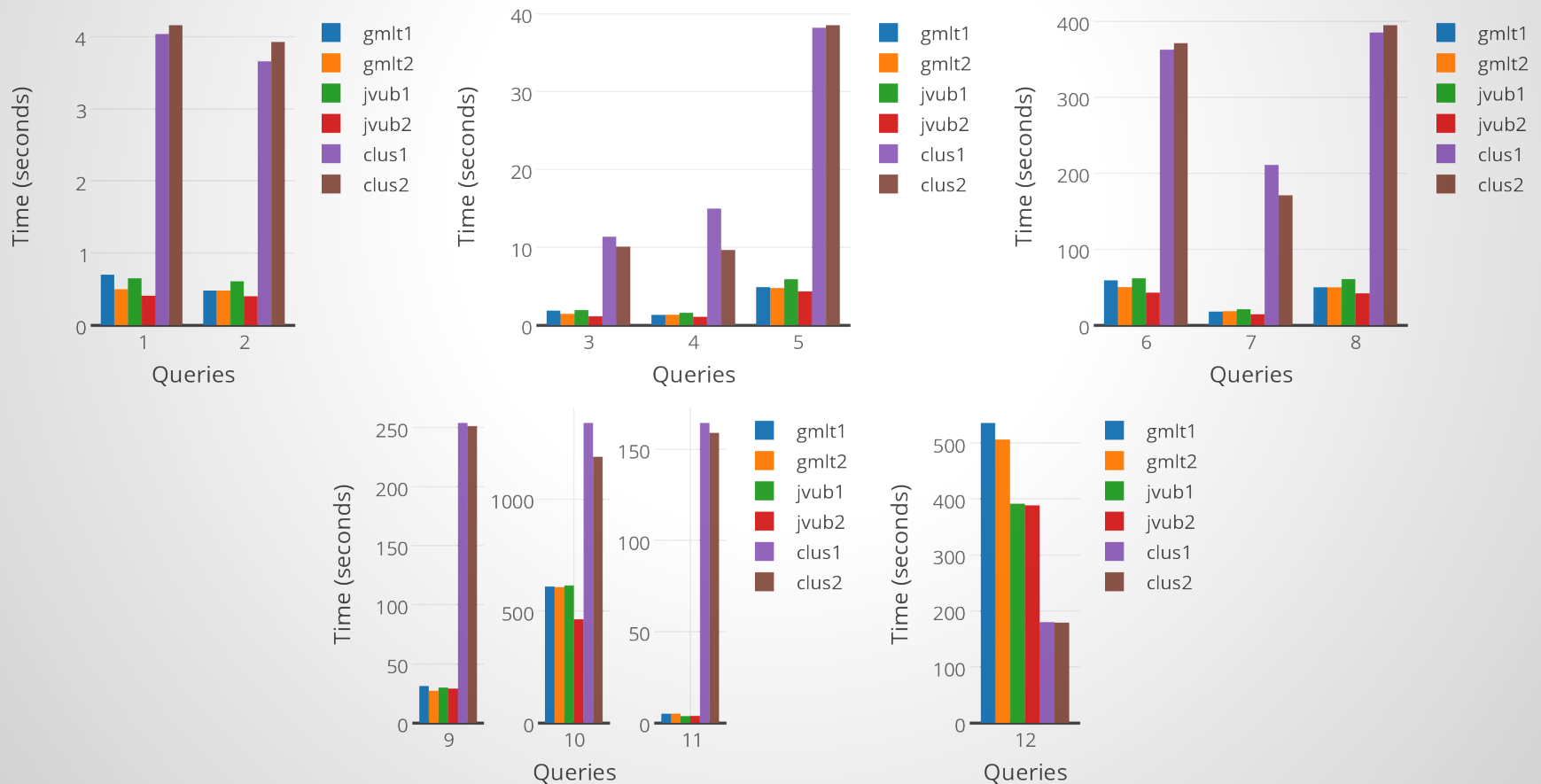


Validations



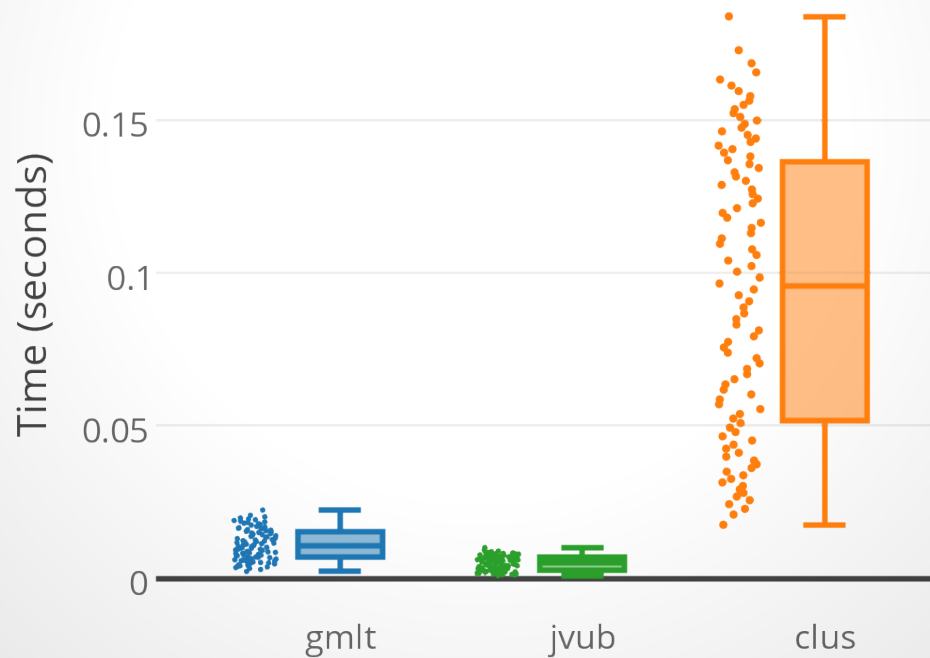
Experimental Results (scalable)

- Queries



Experimental Results (Scalable)

- Revenue Computation



Conclusions

- The created Benchmark covers all possible scenarios
- The storage device makes a difference when extensively used
- The system works well in a Standalone architecture
- The Scalable architecture must be on a high-speed network
- The queries shall be designed for use in a distributed system
- The system can be scalable