Team Number 6: E-Commerce System using Javalin

Alexandru, Dumitru
Andrejs, Kārkliņš
Xinrui, Xu
Zenan, Guan

…to serve and protect data.

Technology

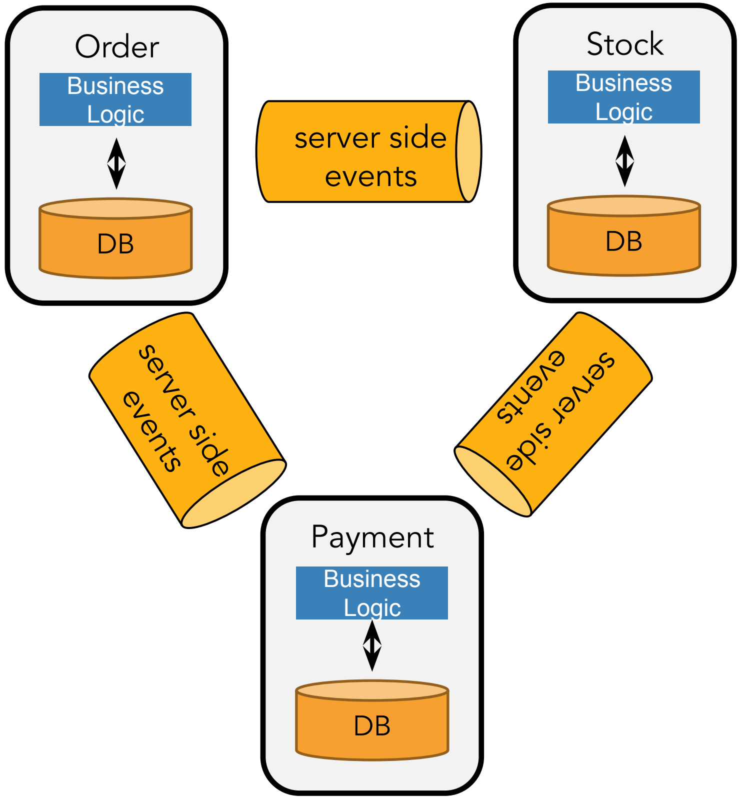
Javalin (Java)

- Light frame
- Restful API

Cassandra

- Nosql database
- Scalability
- Distributed database

Services Architecture: Event Sourcing

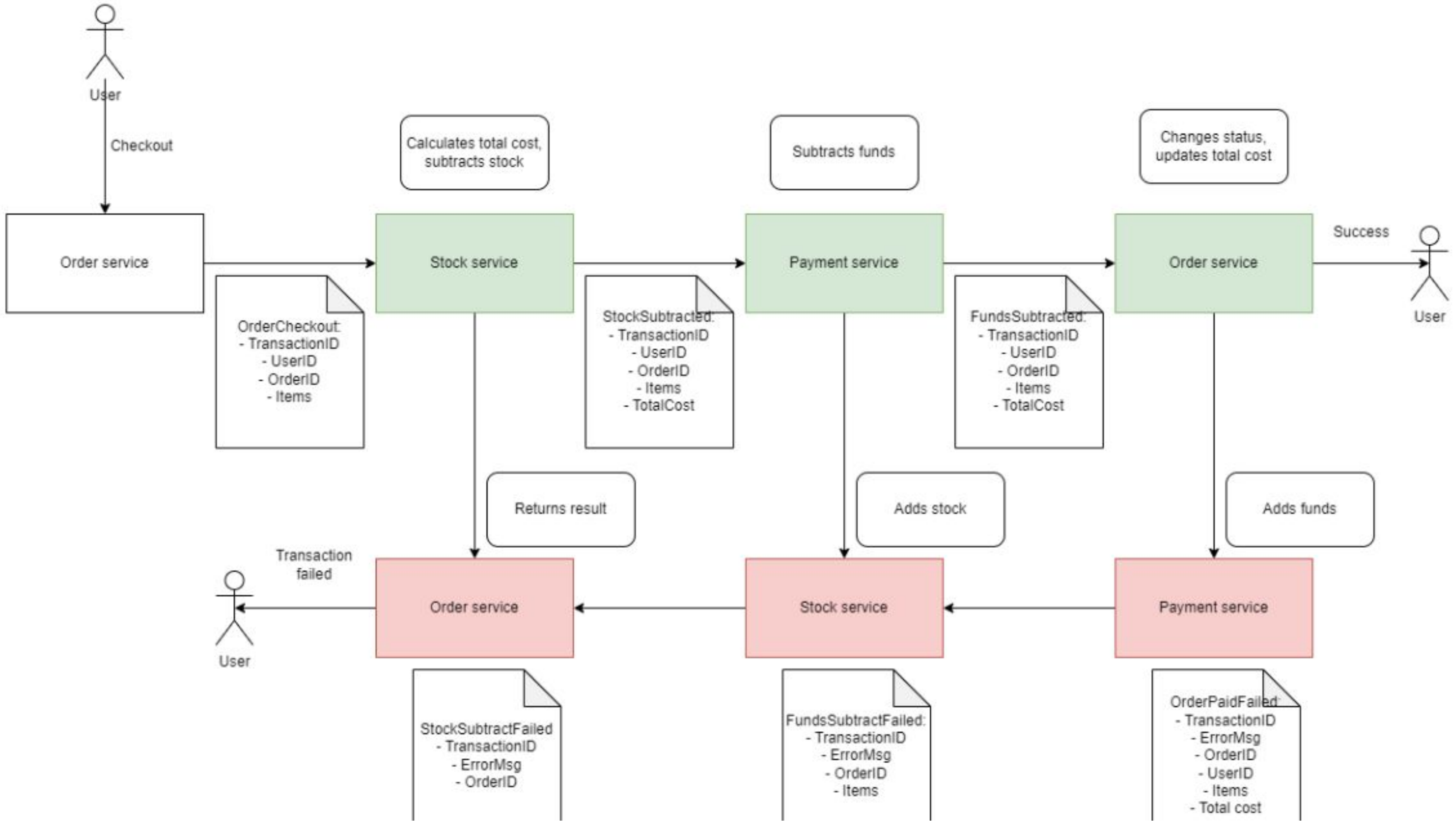


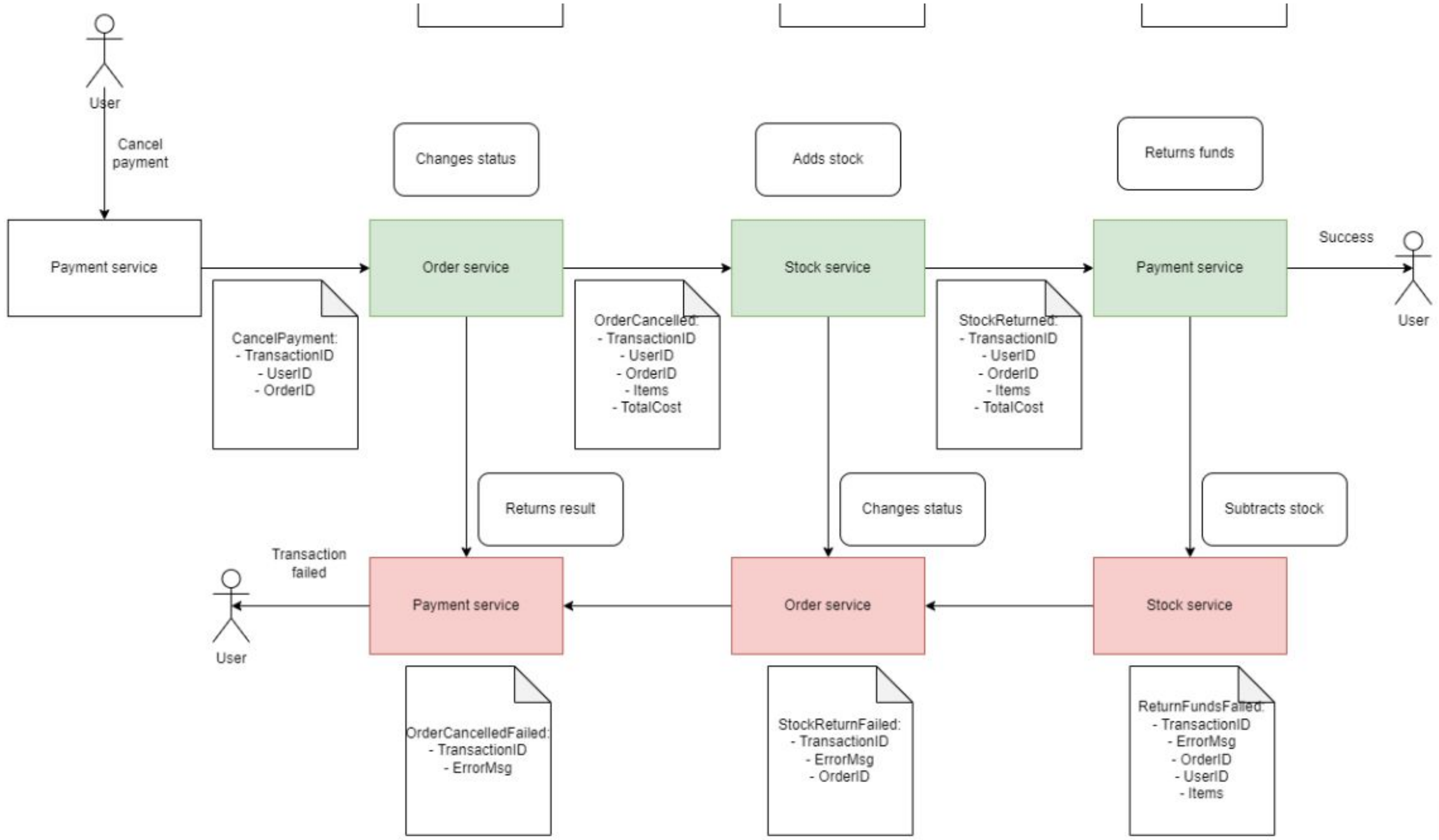
- SSE (Server-side events) communication between microservice.
- 2 Threads reserved for keep-alive communication
- Stock-service queue based writes. Consistency > speed.
- Keep total-cost lazy. Consistency < speed.

SAGA (Choreography)

Global transactions

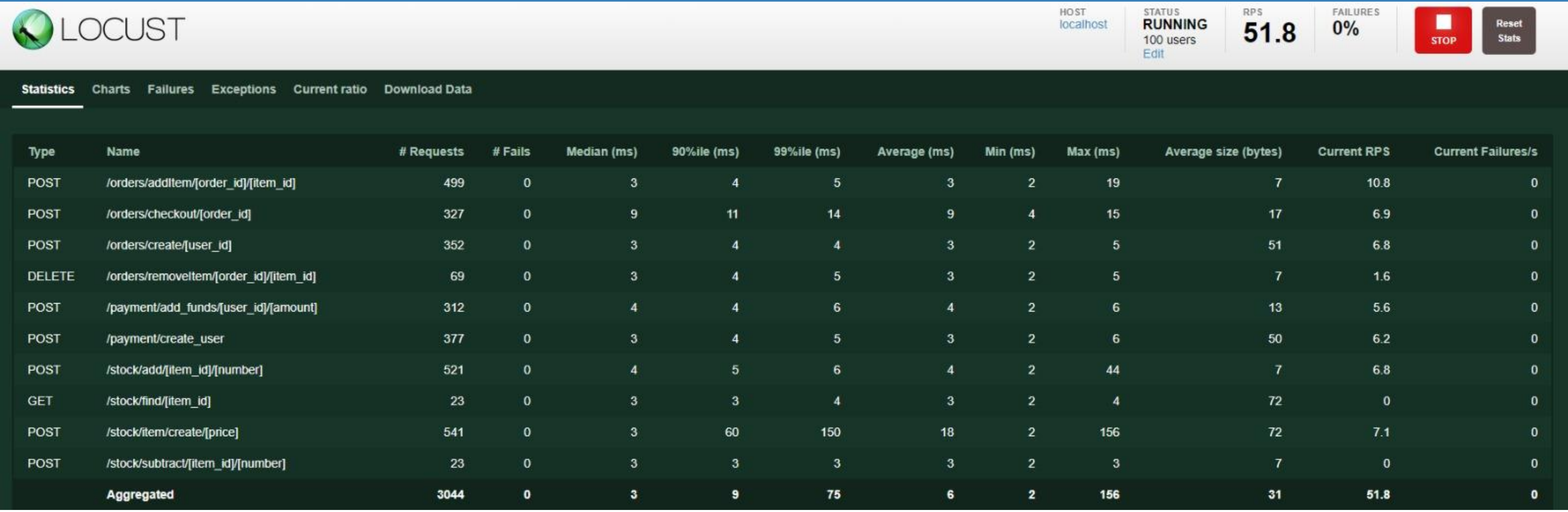
- **Checkout Transaction:**
  - Subtract stock
  - Subtract balance
  - Change status
- **Cancel Transaction:**
  - Return balance
  - Return stock
  - Change status


User
Checkout
Order service
Calculates total cost, subtracts stock
Stock service
Subtracts funds
Payment service
Changes status, updates total cost
Order service
Success
User
OrderCheckout:
- TransactionID
- UserID
- OrderID
- Items
StockSubtracted:
- TransactionID
- UserID
- OrderID
- Items
- TotalCost
FundsSubtracted:
- TransactionID
- UserID
- OrderID
- Items
- TotalCost
Returns result
Adds stock
Adds funds
Transaction failed
User
Order service
Stock service
Payment service
StockSubtractFailed
- TransactionID
- ErrorMsg
- OrderID
FundsSubtractFailed:
- TransactionID
- ErrorMsg
- OrderID
- Items
OrderPaidFailed:
- TransactionID
- ErrorMsg
- OrderID
- UserID
- Items
- Total cost


User
Cancel payment
Payment service
CancelPayment: - TransactionID - UserID - OrderID
Changes status
Order service
OrderCancelled: - TransactionID - UserID - OrderID - Items - TotalCost
Adds stock
Stock service
StockReturned: - TransactionID - UserID - OrderID - Items - TotalCost
Returns funds
Payment service
Success
User
Returns result
Changes status
Subtracts stock
Transaction failed
User
Payment service
Order service
Stock service
OrderCancelledFailed: - TransactionID - ErrorMsg
StockReturnFailed: - TransactionID - ErrorMsg - OrderID
ReturnFundsFailed: - TransactionID - ErrorMsg - OrderID - UserID - Items

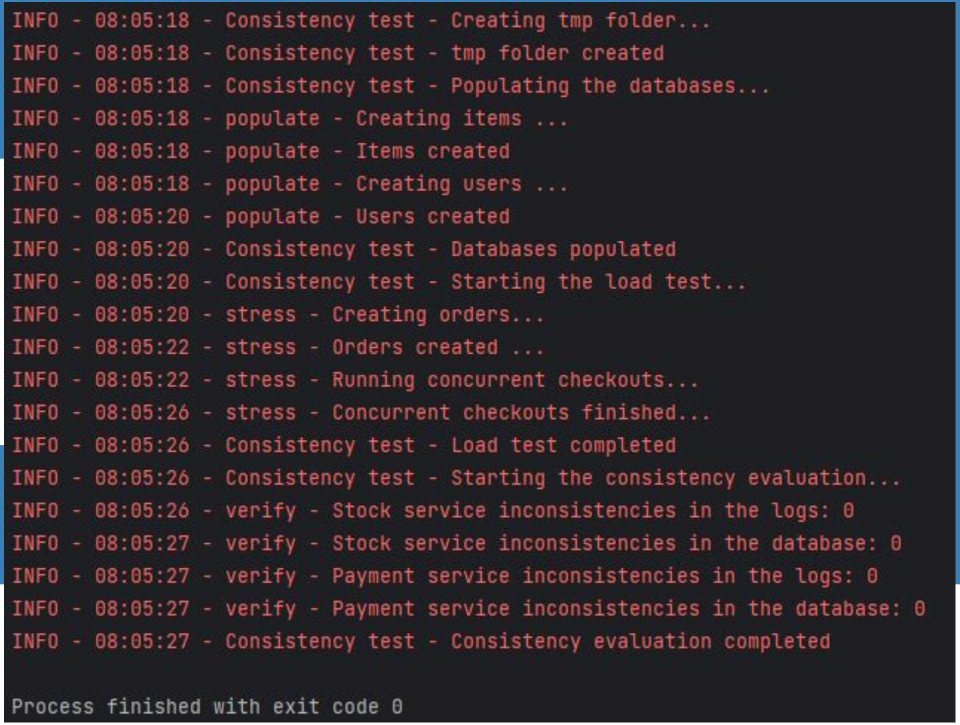
Results

Stress Test



Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
POST	/orders/addItem/[order_id]/[item_id]	499	0	3	4	5	3	2	19	7	10.8	0
POST	/orders/checkout/[order_id]	327	0	9	11	14	9	4	15	17	6.9	0
POST	/orders/create/[user_id]	352	0	3	4	4	3	2	5	51	6.8	0
DELETE	/orders/removeItem/[order_id]/[item_id]	69	0	3	4	5	3	2	5	7	1.6	0
POST	/payment/add_funds/[user_id]/[amount]	312	0	4	4	6	4	2	6	13	5.6	0
POST	/payment/create_user	377	0	3	4	5	3	2	6	50	6.2	0
POST	/stock/add/[item_id]/[number]	521	0	4	5	6	4	2	44	7	6.8	0
GET	/stock/find/[item_id]	23	0	3	3	4	3	2	4	72	0	0
POST	/stock/item/create/[price]	541	0	3	60	150	18	2	156	72	7.1	0
POST	/stock/subtract/[item_id]/[number]	23	0	3	3	3	3	2	3	7	0	0
	Aggregated	**3044**	**0**	**3**	**9**	**75**	**6**	**2**	**156**	**31**	**51.8**	**0**

Consistency Test



```
INFO - 08:05:18 - Consistency test - Creating tmp folder...
INFO - 08:05:18 - Consistency test - tmp folder created
INFO - 08:05:18 - Consistency test - Populating the databases...
INFO - 08:05:18 - populate - Creating items ...
INFO - 08:05:18 - populate - Items created
INFO - 08:05:18 - populate - Creating users ...
INFO - 08:05:20 - populate - Users created
INFO - 08:05:20 - Consistency test - Databases populated
INFO - 08:05:20 - Consistency test - Starting the load test...
INFO - 08:05:20 - stress - Creating orders...
INFO - 08:05:22 - stress - Orders created ...
INFO - 08:05:22 - stress - Running concurrent checkouts...
INFO - 08:05:26 - stress - Concurrent checkouts finished...
INFO - 08:05:26 - Consistency test - Load test completed
INFO - 08:05:26 - Consistency test - Starting the consistency evaluation...
INFO - 08:05:26 - verify - Stock service inconsistencies in the logs: 0
INFO - 08:05:27 - verify - Stock service inconsistencies in the database: 0
INFO - 08:05:27 - verify - Payment service inconsistencies in the logs: 0
INFO - 08:05:27 - verify - Payment service inconsistencies in the database: 0
INFO - 08:05:27 - Consistency test - Consistency evaluation completed

Process finished with exit code 0
```

What would we do better?

Main improvements

- Work on fault-tolerance
  - Logging
  - Redundancy
  - Recoverability
  - Account for unfinished transactions
- Use full potential of Cassandra:
  - Caching
  - Data replication
- Asynchronous non-blocking I/O for all HTTP request handling.

Thank You!