Couchbase stress testing application

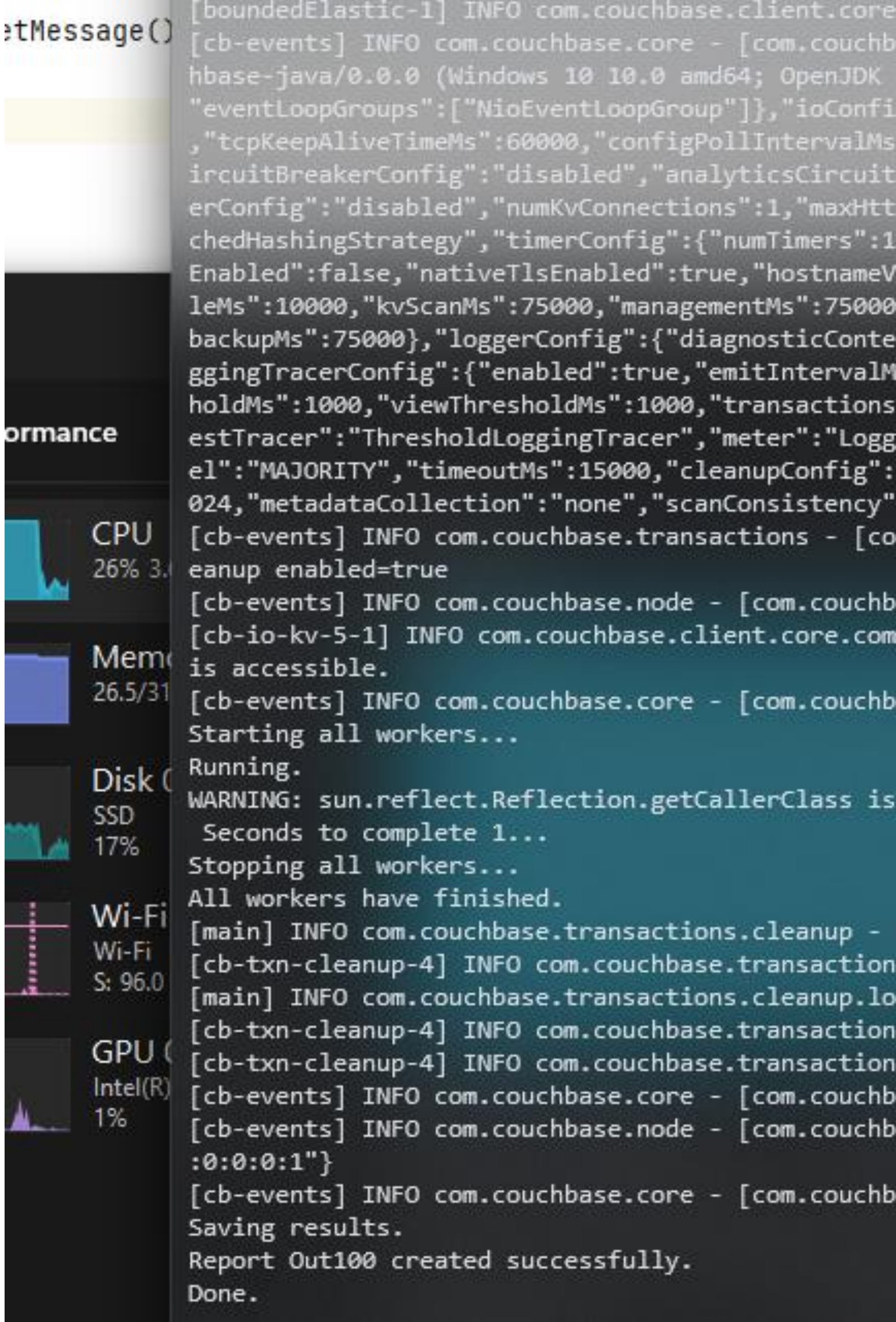
BeerFast

BeerFast

- Why BeerFast ?
- Code repository
  - Tinariven/BeerFast
- Author
  - Michał Joachimski & Copilot “pair programming”

Test conditions

- Coachbase installed as Docker container inside WSL Ubuntu (Windows Subsystem Linux)
- Run on
  - 13th Gen Intel(R) Core(TM) i7-1370P   1.90 GHz
  - 32GB RAM
  - Java 11 OpenJDK build 11.0.8



```
[boundedElastic-1] INFO com.couchbase.client.core
[cb-events] INFO com.couchbase.core - [com.couchb
hbase-java/0.0.0 (Windows 10 10.0 amd64; OpenJDK
"eventLoopGroups":["NioEventLoopGroup"]},"ioConfi
,"tcpKeepAliveTimeMs":60000,"configPollIntervalMs
ircuitBreakerConfig":"disabled","analyticsCircuit
erConfig":"disabled","numKvConnections":1,"maxHtt
chedHashingStrategy","timerConfig":{"numTimers":1
Enabled":false,"nativeTlsEnabled":true,"hostnameV
leMs":10000,"kvScanMs":75000,"managementMs":75000
backupMs":75000},"loggerConfig":{"diagnosticConte
ggingTracerConfig":{"enabled":true,"emitIntervalM
holdMs":1000,"viewThresholdMs":1000,"transactions
estTracer":"ThresholdLoggingTracer","meter":"Logg
el":"MAJORITY","timeoutMs":15000,"cleanupConfig":
024,"metadataCollection":"none","scanConsistency"
[cb-events] INFO com.couchbase.transactions - [co
eanup enabled=true
[cb-events] INFO com.couchbase.node - [com.couchb
[cb-io-kv-5-1] INFO com.couchbase.client.core.com
is accessible.
[cb-events] INFO com.couchbase.core - [com.couchb
Starting all workers...
Running.
WARNING: sun.reflect.Reflection.getCallerClass is
 Seconds to complete 1...
Stopping all workers...
All workers have finished.
[main] INFO com.couchbase.transactions.cleanup -
[cb-txn-cleanup-4] INFO com.couchbase.transaction
[main] INFO com.couchbase.transactions.cleanup.lo
[cb-txn-cleanup-4] INFO com.couchbase.transaction
[cb-txn-cleanup-4] INFO com.couchbase.transaction
[cb-events] INFO com.couchbase.core - [com.couchb
[cb-events] INFO com.couchbase.node - [com.couchb
:0:0:0:1"}
[cb-events] INFO com.couchbase.core - [com.couchb
Saving results.
Report Out100 created successfully.
Done.
```

Test scenario

Run with 1,10,50,100,1000 number of threads

Loaded up 100 json files

Test over 180 seconds

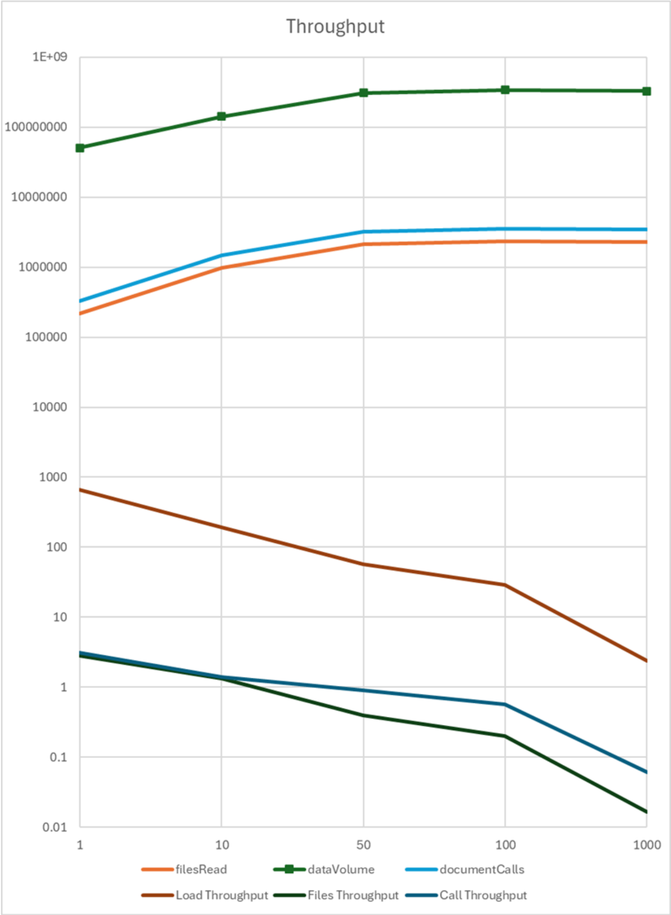
8 measures collected

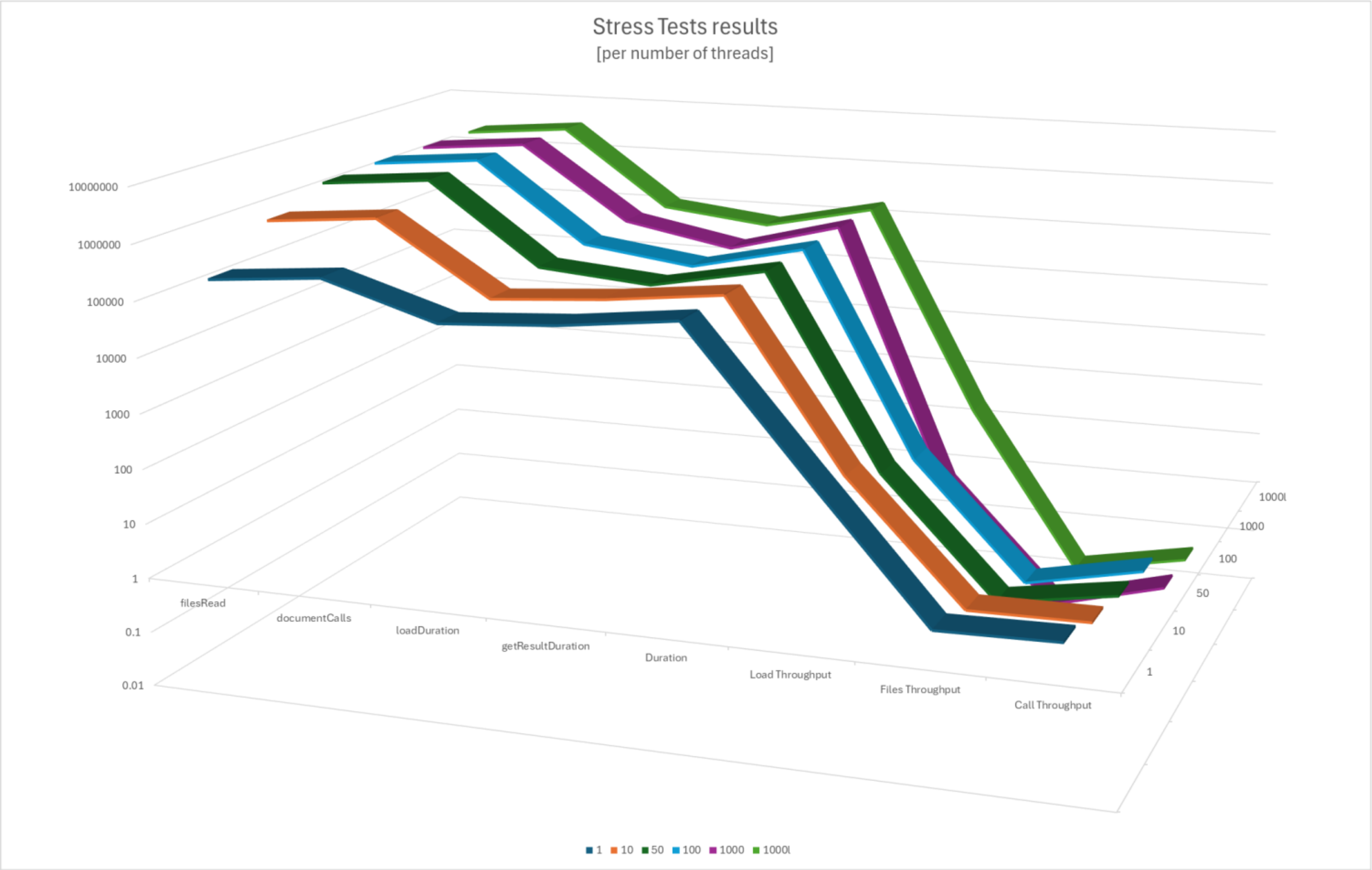
Output stored as csv files

Measures

- Thread start and end timestamp [ms]
- Number of loops executed
- Document loading duration (loadDuration) [ms]
- Document retrieving duration (getResultDuration) [ms]
- Number of document read & loaded (filesRead)
- Number of document retrieved (documentCalls)
- Document volume (dataVolume) [B]

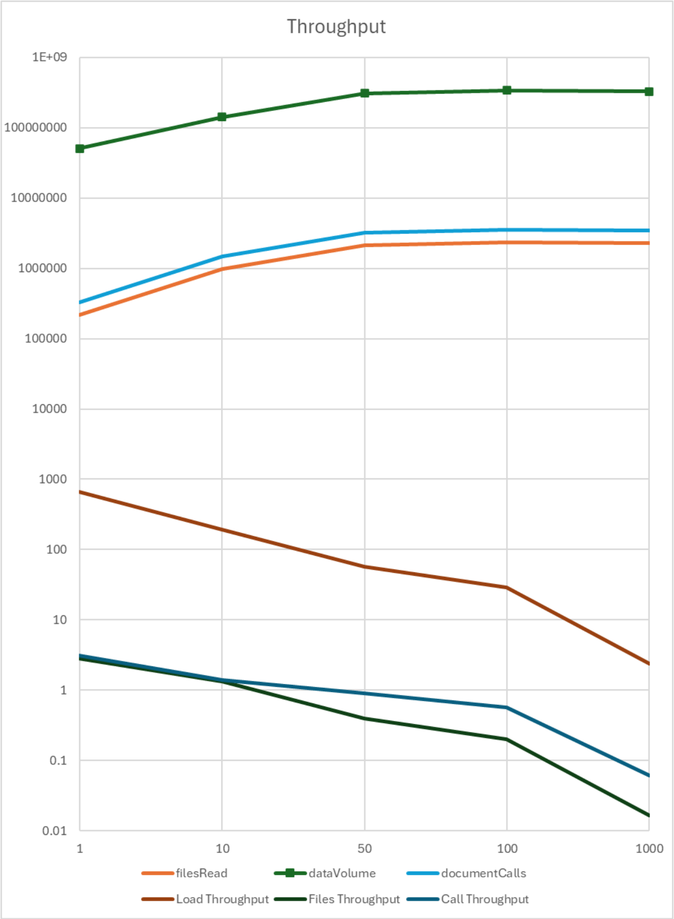
Test results





Conclusions

- Increasing number of threads
  - From 1 to 10 increases number of documents processed by 3.5x
  - Jump from 10 to 50 just doubles quantity
  - 100 gives only 10% increase over 50
  - 1000 threads does not bring value
- With large number of threads > 100
  - CPU, disk (IO) and Couchbase API limitation became a bottle neck
- Increasing the size of documents 10x does not affect performance



Final thoughts

- It was fun to do this exercise
- Good is enough
  - Each code can be improved, but do not need to
- Programming is like riding the bicycle
  - You can change bike but do not forget how to ride
- AI automates simple tasks
- I’ve recall/learned a lot ☺
- I like beer