PAUL SCHERRER INSTITUT

PSI

brightness

Mark Könnecke, Michele Brambilla, Dominik Werder

Status Report
WP5 Second Integration Meeting

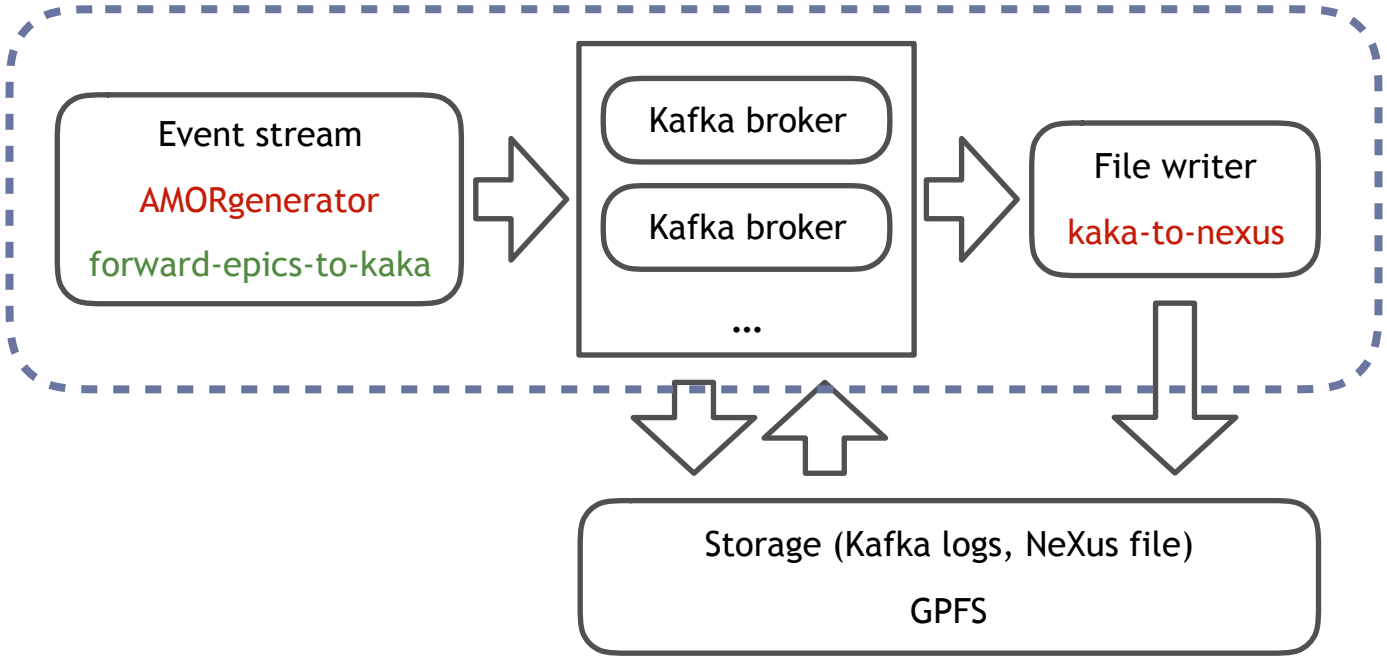
PAUL SCHERRER INSTITUT

Overview

- Performance Tests
- Investigating cluster access
- Integration with NICOS
- File Writer Code Review
- PSI Topics

Testing the toolchain

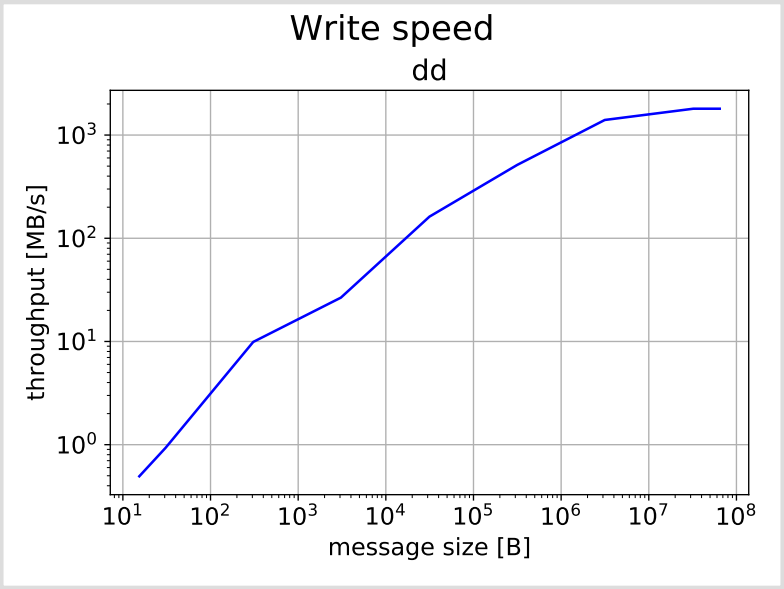
We want to test the performances of the whole toolchain from the **production** of an event stream to the **storage** of the nexus file on the disk



The testing environment

- Processor: 2 x Intel Xeon E5-2690 @ 2.60GHz, 14 cores (no hyperthreading)
- Memory: 256GB
- File system: GPFS via 4x Infiniband FDR

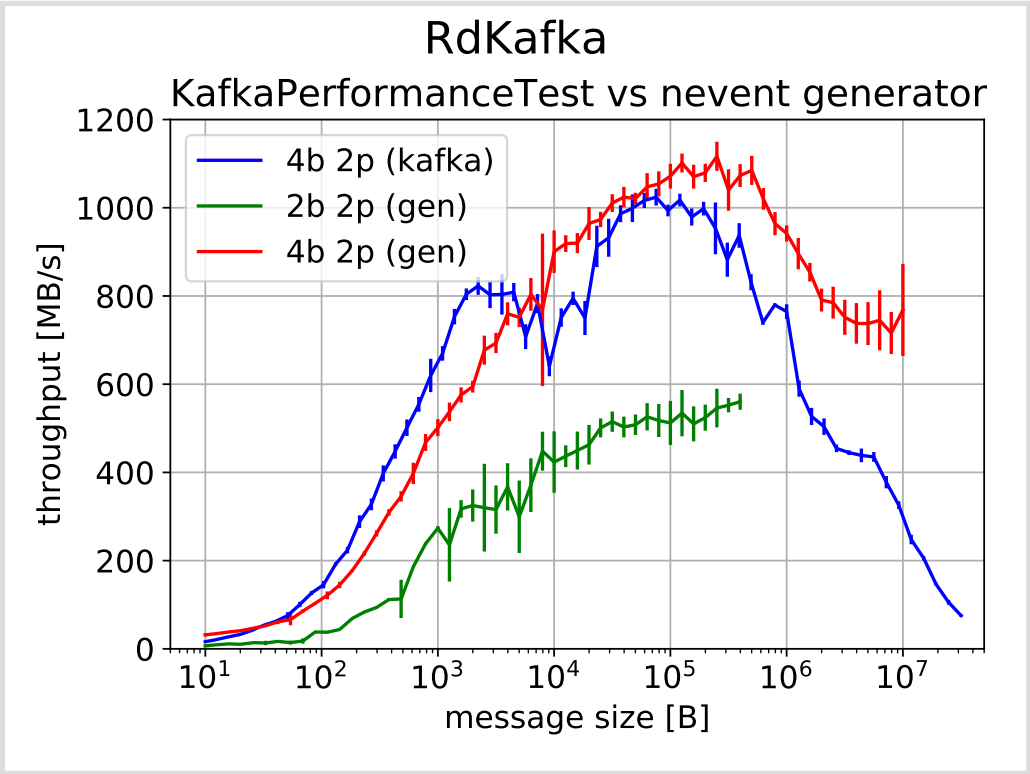
Same machine for **producer**, **broker** and **consumer**
File system shared with other users



Max: 4.5 GB/sec Message size dependent!

Kafka producer & neutron event generator

SINQ-AMORsim (neutron event generator)



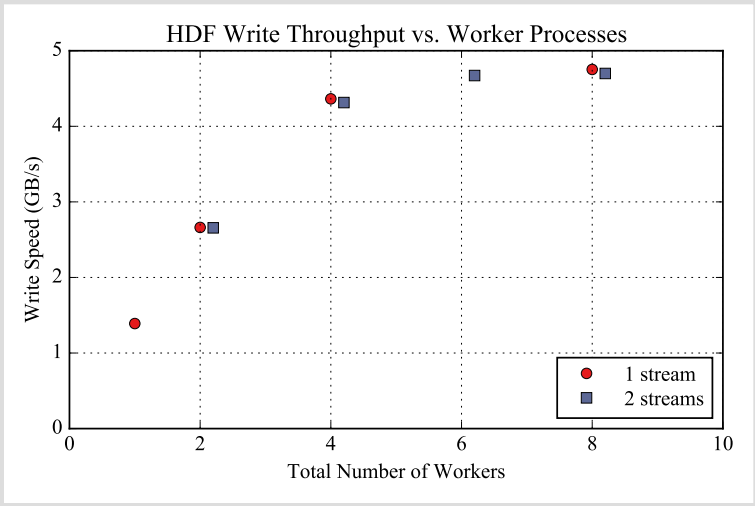
- Comparable results with KafkaPerformance test
- Slightly better results for large messages
- Dependence on number of brokers
- Performances are affected by compressive load of the system
- >1.2 GB/s, peaks 1.8 GB/s

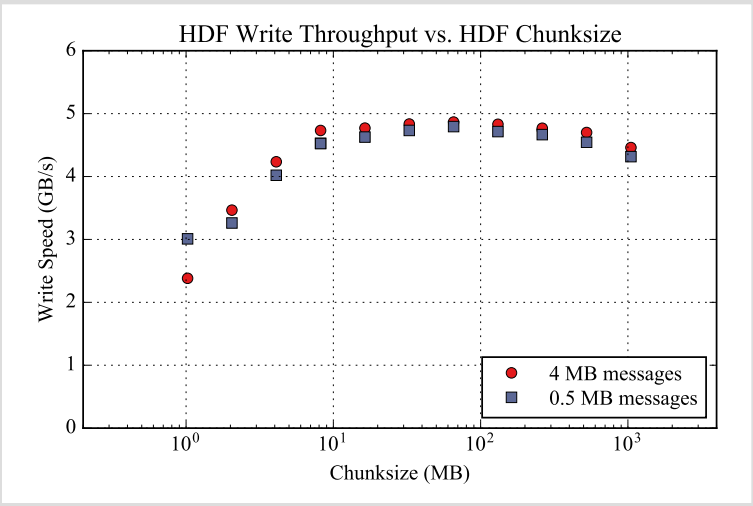
Kafka & Filewriter (write on disk)

memory to HDF

Results presented by D. Werder at ICALEPICS 2017

- Pre-generate messages in memory
- Queue messages and feed the writer





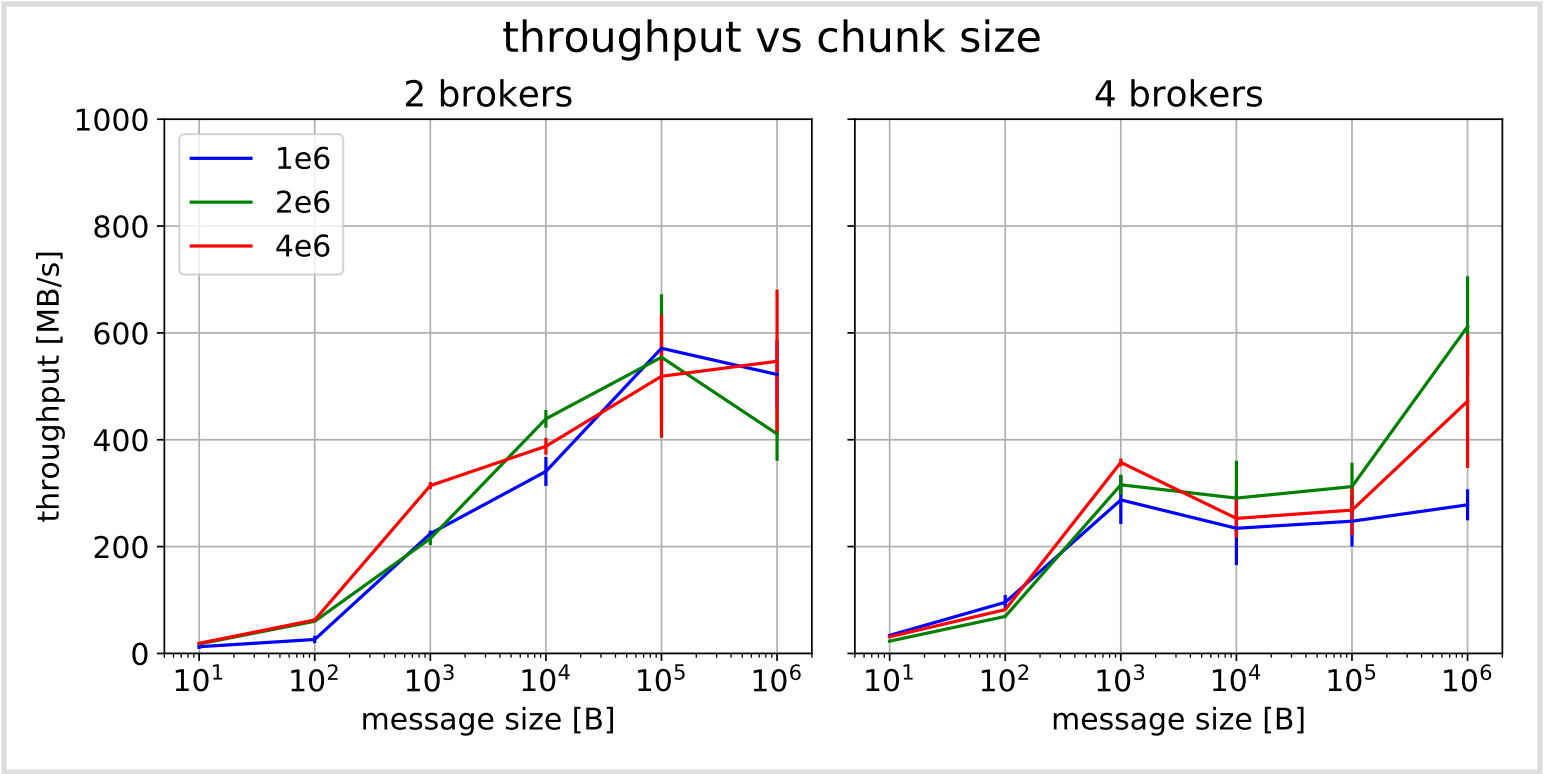
- 6 processes reach a maximum performance of 4.8GB/s
- The optimal throughput can be achieved for a range of chunk size

PAUL SCHERRER INSTITUT

event stream to Kafka to HDF

To achieve good performances some tuning is required

```
"nexus": {
    "indices": {
        "index_every_kb": 4096
    },
    "chunk": {
        "chunk_n_elements": 2000000,
    },
    "buffer": {
        "size_kb": 4096,
        "packet_max_kb": 16
    }
}
```



Performance Test Conclusions

Current status

- On a single machine the producer and the consumer can reach 1.4 GB/s
- Single process writes up to 1.4 GB/s
- The full toolchain tested on a single machine with GPFS reaches 600 MB/s
- Dependence on the number of brokers

Ongoing development

- Parallel writer
- Parallel consuming
- Nikos integration

Proper performance tests requires separate hardware for brokers, producers and consumers

Network throughput on 127.0.0.1

- iperf3 -c localhost: 8.6GB/sec
- iperf3 -P 4 -c localhost: 1.4GB/sec, sum: 5.8GB/sec
- iperf3 -P 6 -c localhost: 1.59GB/sec, sum: 9.52 GB/sec

- Localhost throughput is shared between different processes…
- Explains observed limitation to a degree…

Investigating Cluster Access

- Google: did not come back on me, no fast disk writing on WWW
- Amazon: did come back, max 500MB/sec
- MS Azure: still waiting, in principle up to 2GB/sec from WWW
- EULER (ETHZ), SCSC (Lugano): batch operation, we will never get exclusive access to nodes as required for undisturbed performance tests
- NUM clusters: optimized for McStas: high CPU, little I/O
- DMSC, Missing In Action :-(
- Under investigation: Fast SSD plus normal workstations
  - Intel Optane 900P ~2GB/sec sequential R/W, ~600 CHF
  - Intel SSD P4600, sequential R: 3.2 GB/sec, W: 1.65GB/sec, ~1300 CHF
  - Samsung 960 Pro, M.2, 1TB, 3.5GB/sec read, 2.1GB/sec write, both sequential, 523EU
- PSI: starved….
  - UPDATE: we get more nodes in the SLS cluster
  - I want a prioritized testing schedule from this meeting

ESS will definitely need a dedicated cluster!!

Integration with NICOS

- **SUCCESS!!**
- NICOS can now configure the Forwarder and write files via NeXus File Writer
- We have the complete chain running from event generator to NeXus file
- https://github.com/ess-dmsc/dm-sinq-amor
- Changes were required:
  - NeXus File Writer: Michele
  - Forwarder: Dominik
  - System management issues: Dominik

NeXus File Writer Code Review

- Code reviewed
- Still a number of things to be addressed
  - Documentation
  - Error handling
  - Integration of parallel writing
- We are working down the list of issues now

Tests at Real AMOR

- I arranged with Jochen Stahn that we can test AMOR integration at the real thing in the second half of 2018
- 3-4 slots of a few days distributed across that time
- Mode of operation:
  - We test for a few days until instrument scientists is fed up
  - Write down what bothers us
  - Switch back to SICS
  - Fix issues offline
  - Rinse and repeat
- Issue: AMOR as of now only does histogramming
  - See my new project for event streaming at SINQ

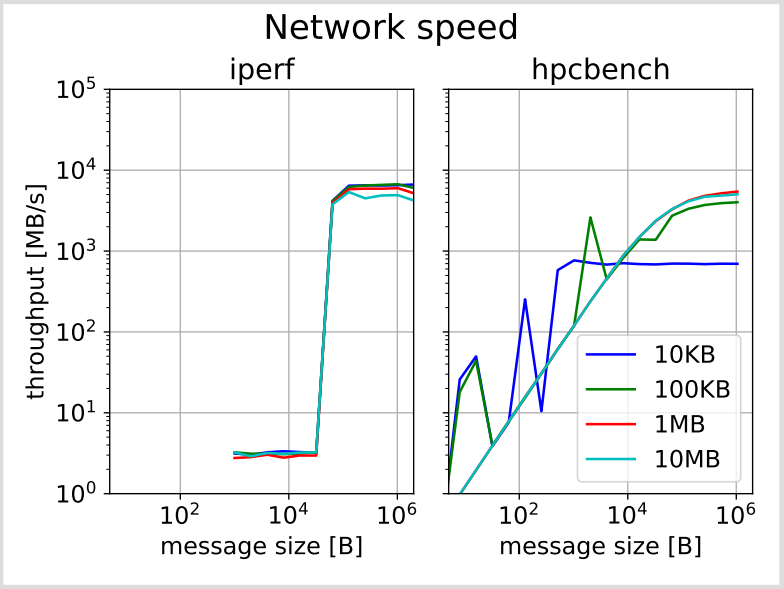
PSI Topics for this Meeting

- Prioritized testing schedule
  - High performance
  - VM cluster
- Review of ansible installers
  - Docker?
  - Plan for starting and stoping dependencies
  - —> Dominiks presentation
- Timepix 2-3 detector coming to PSI for imaging
  - Same as for ODIN
  - May I offer BrightnESS support?
- BrightnESS publication
- Additional projects

The testing environment

- Processor: 2 x Intel Xeon E5-2690 @ 2.60GHz, 14 cores (no hyperthreading)
- Memory: 256GB
- File system: GPFS via 4x Infiniband FDR

Same machine for **producer**, **broker** and **consumer**
File system shared with other users

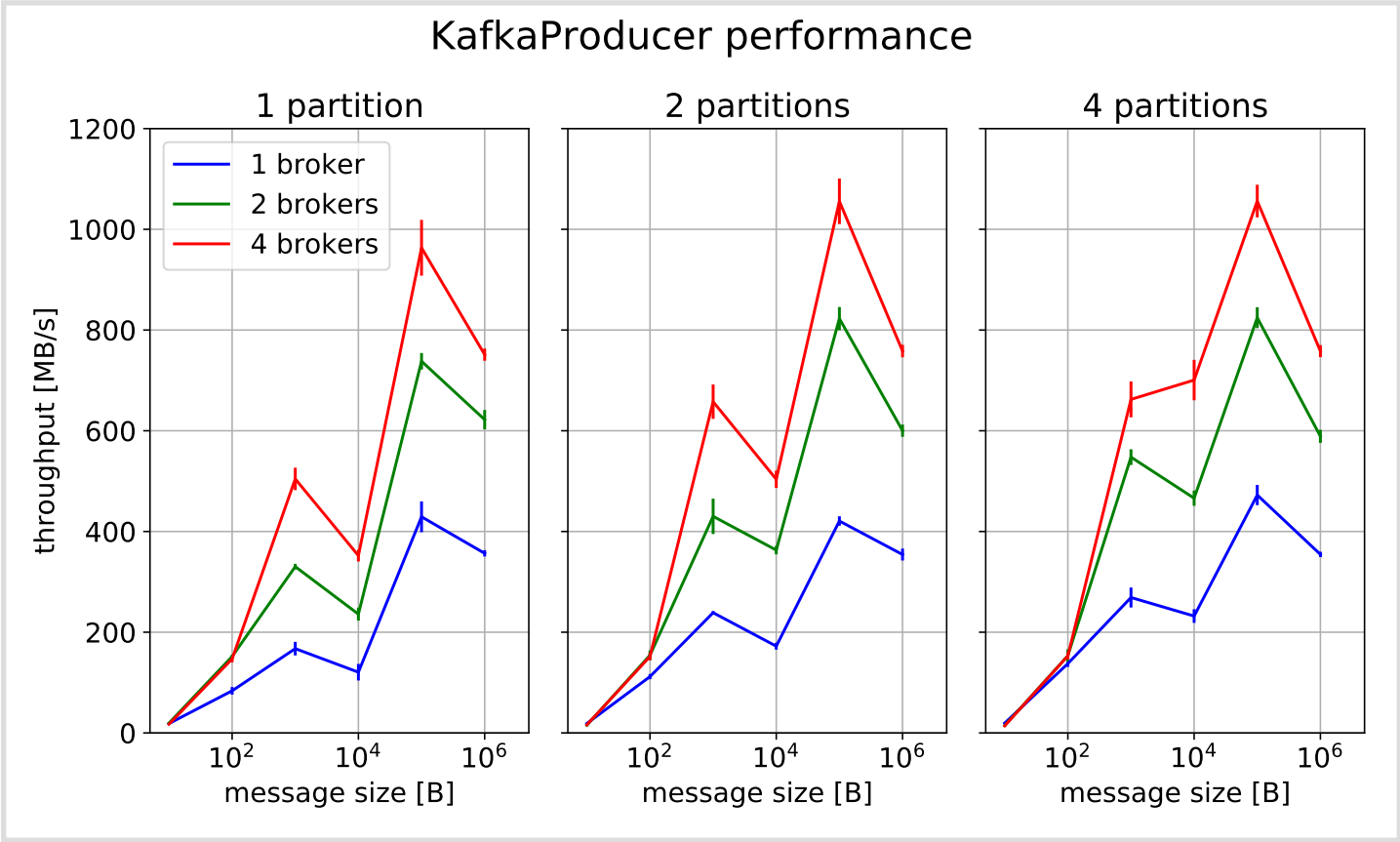



```
iperf3 -c localhost -P <nproc> -f M -w <wsize> -t 10 -i 1
```

Kafka producer & neutron event generator

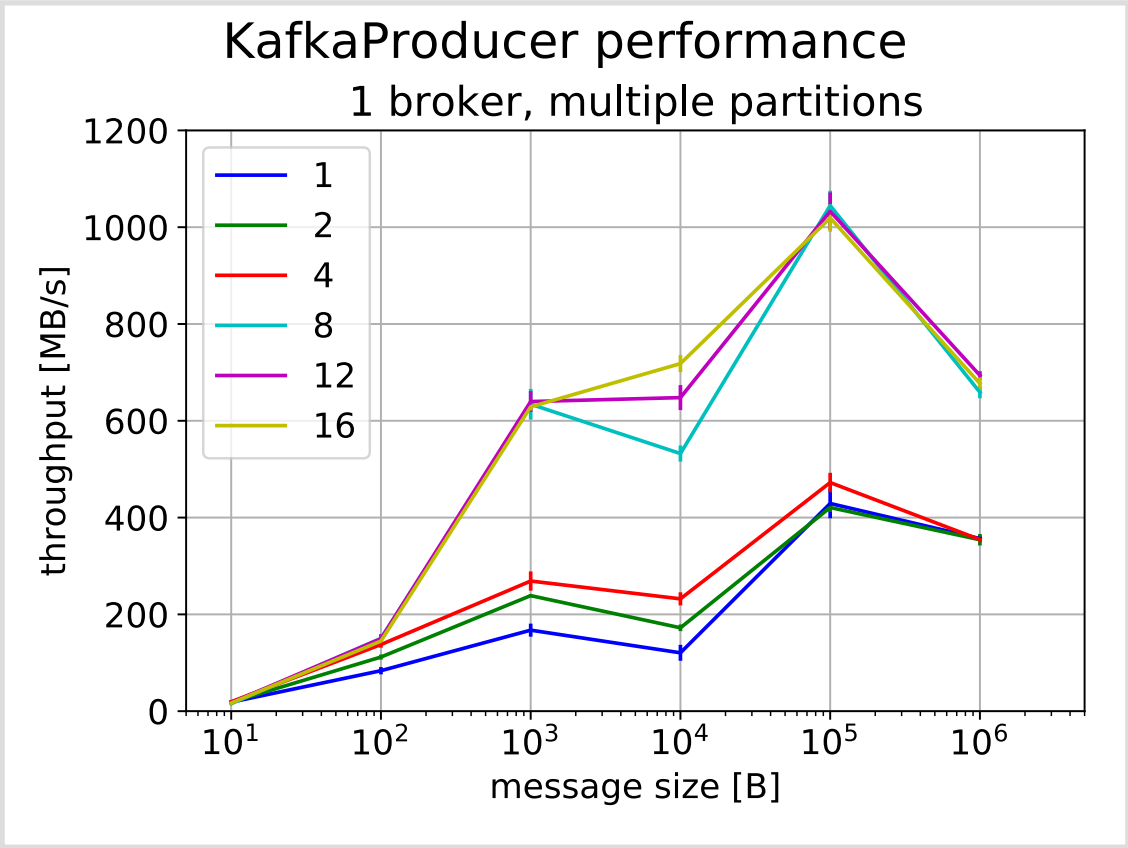
Kafka builtin tools

```
bin/kafka-run-class.sh org.apache.kafka.clients.tools.ProducerPerformance
<topic> <msg-size> 100 -1 acks=1 bootstrap.servers=localhost
[buffer.memory=67108864 batch.size=8196]
```



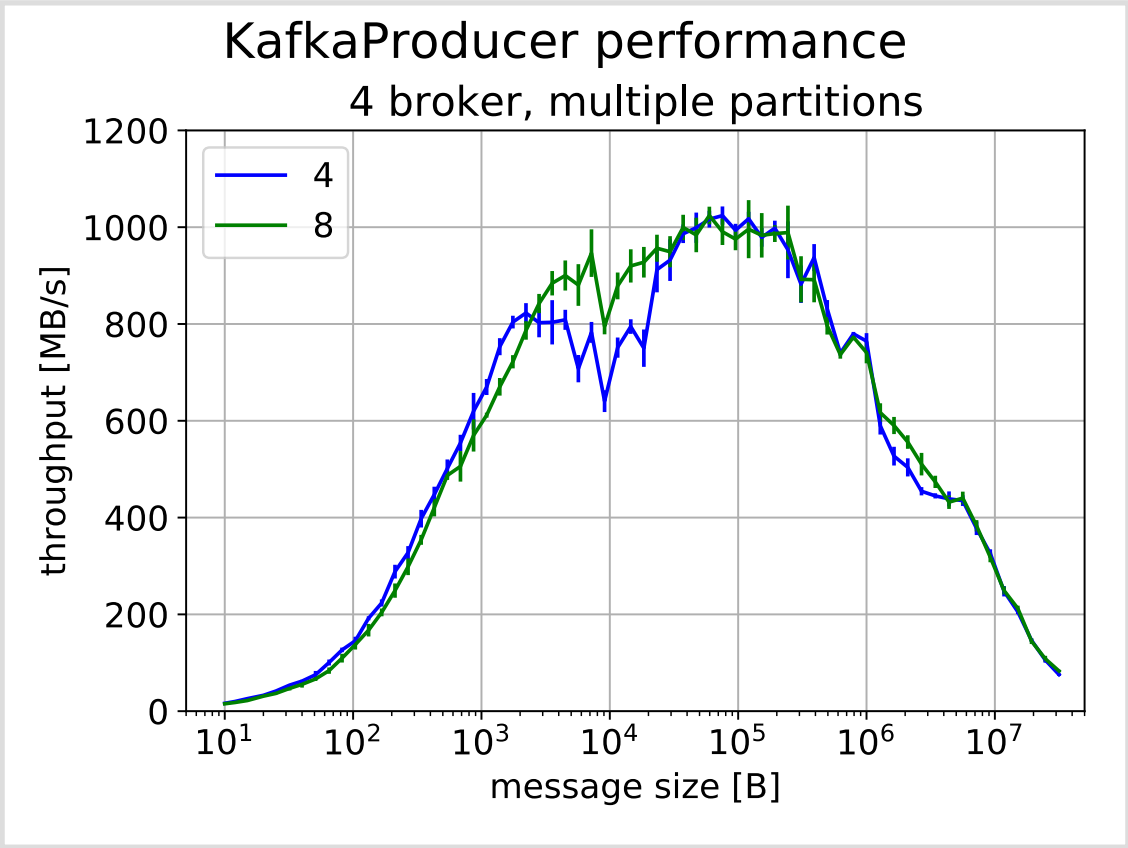
Kafka producer & neutron event generator

bin/kafka-run-class.sh org.apache.kafka.clients.tools.ProducerPerformance <topic> <msg-size> 100 -1 acks=1 bootstrap.servers=localhost [buffer.memory=67108864 batch.size=8196]



Kafka producer & neutron event generator

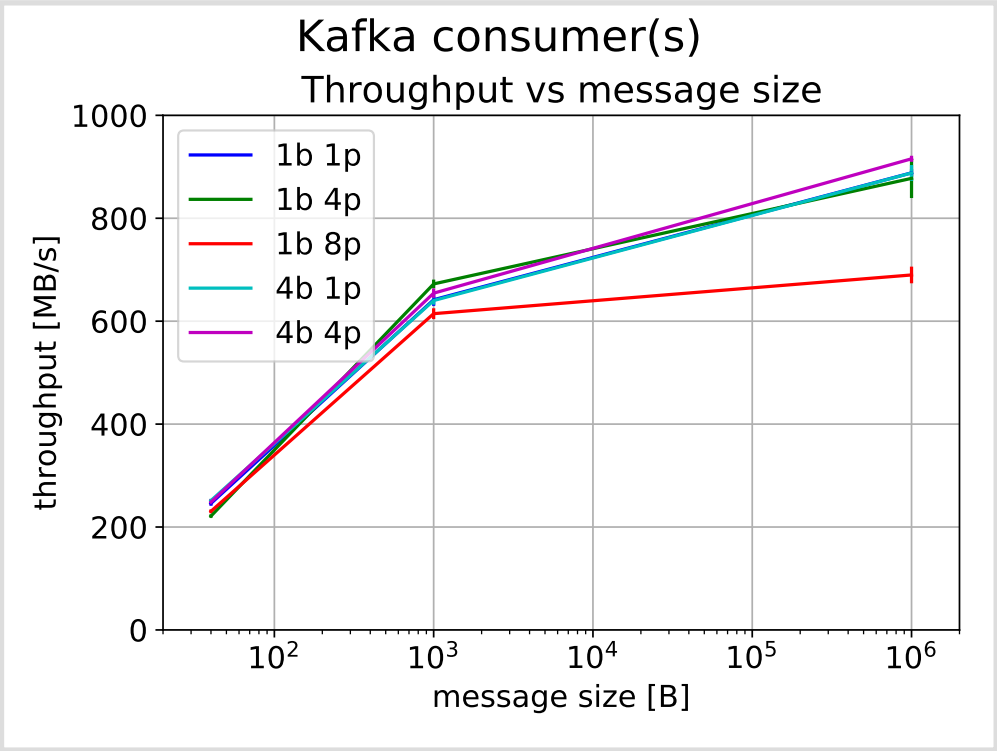
```
bin/kafka-run-class.sh org.apache.kafka.clients.tools.ProducerPerformance
<topic> <msg-size> 100 -1 acks=1 bootstrap.servers=localhost
[buffer.memory=67108864 batch.size=8196]
```



Kafka & Filewriter consumer (no disk)

Kafka builtin tools

```
bin/kafka-consumer-perf-test.sh --zookeeper localhost:2181 --messages <# messages> --topic <topic> --threads 1
```

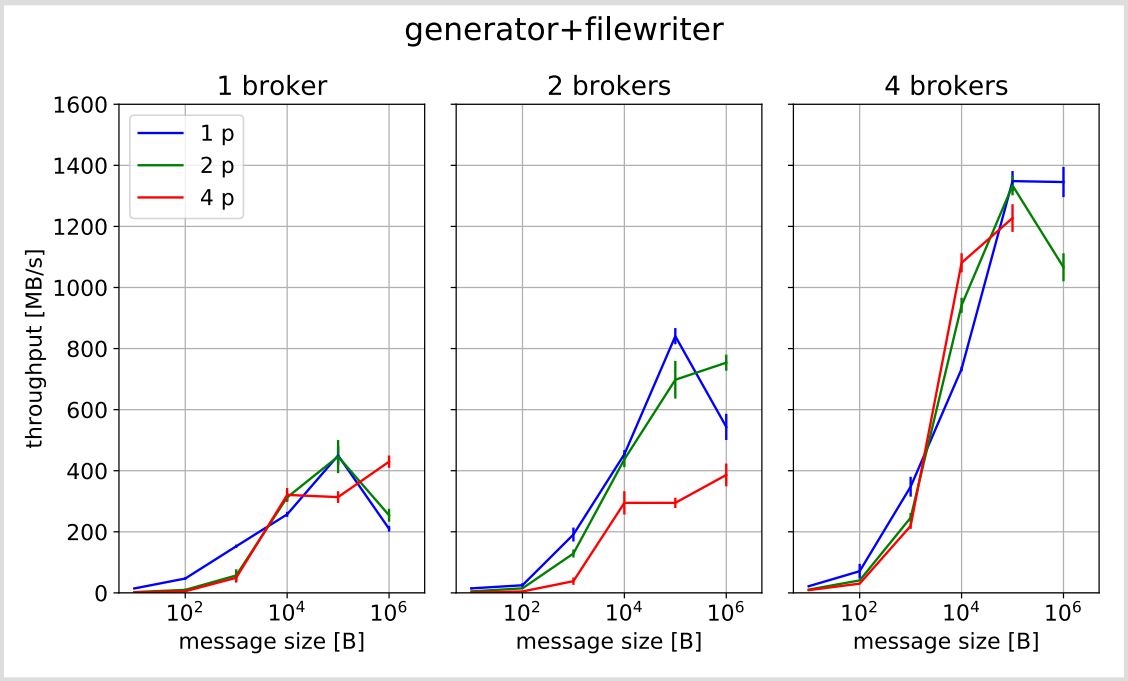


- Apparently independence on the number of brokers

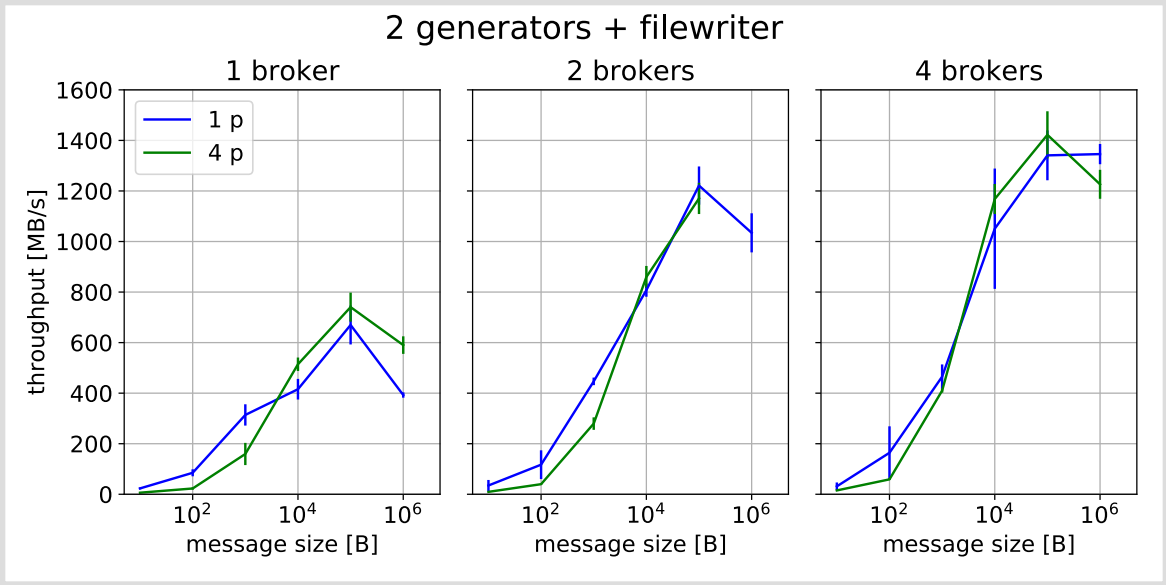
neutron event generator + file writer

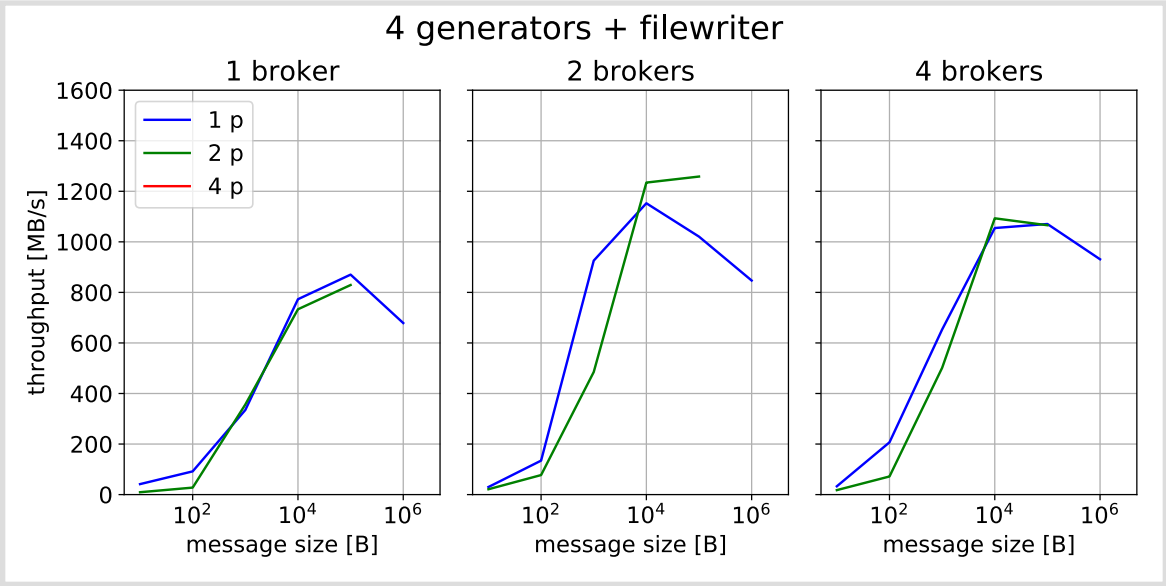
The file writer provides statistics about the consumed messages as Kafka logs. In particular:

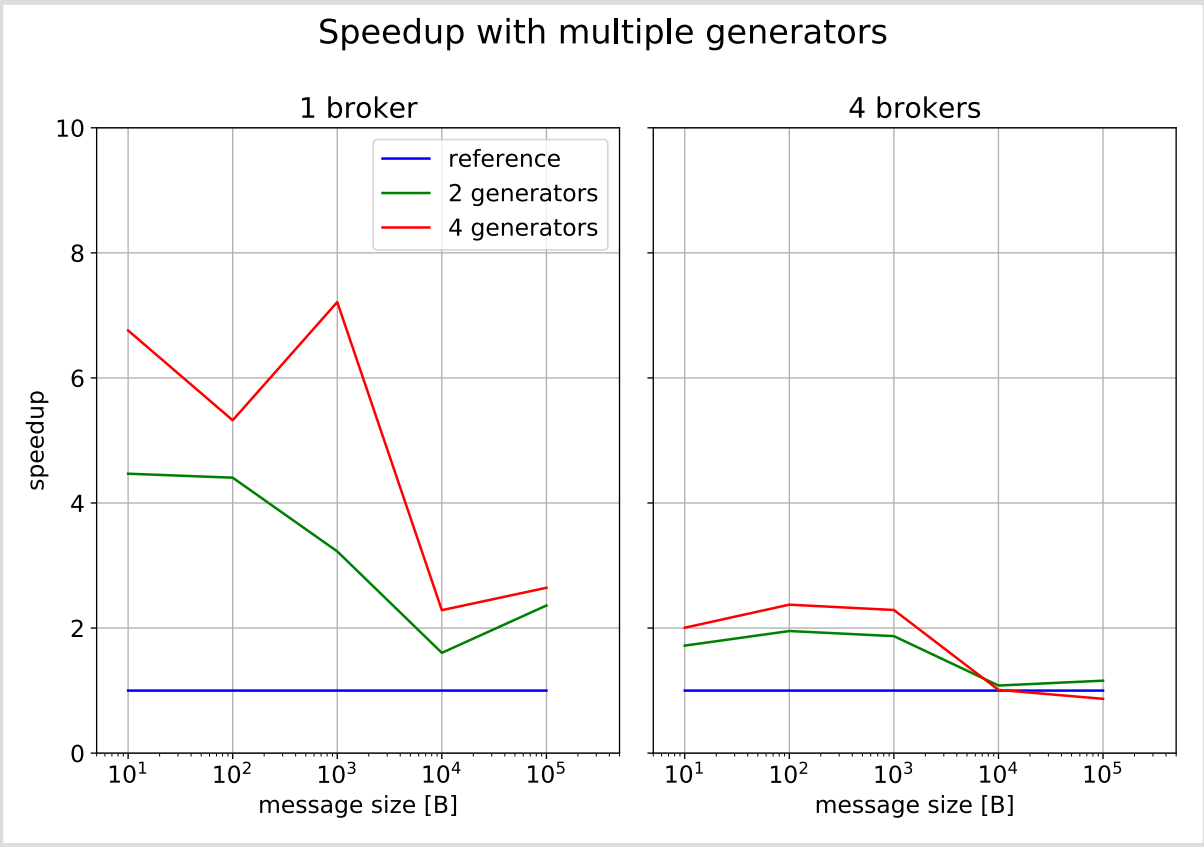
- number of consumed messages and bytes received, runtime (per file)
- throughput and messages/s (per topic)
- number of errors, run status (per file and topic)



- Performance scales with number of brokers
- Independence on the number of partitions
- Better results *wrt* Kafka performance tools


2 generators + filewriter
1 broker
2 brokers
4 brokers
1 p
4 p
throughput [MB/s]
message size [B]

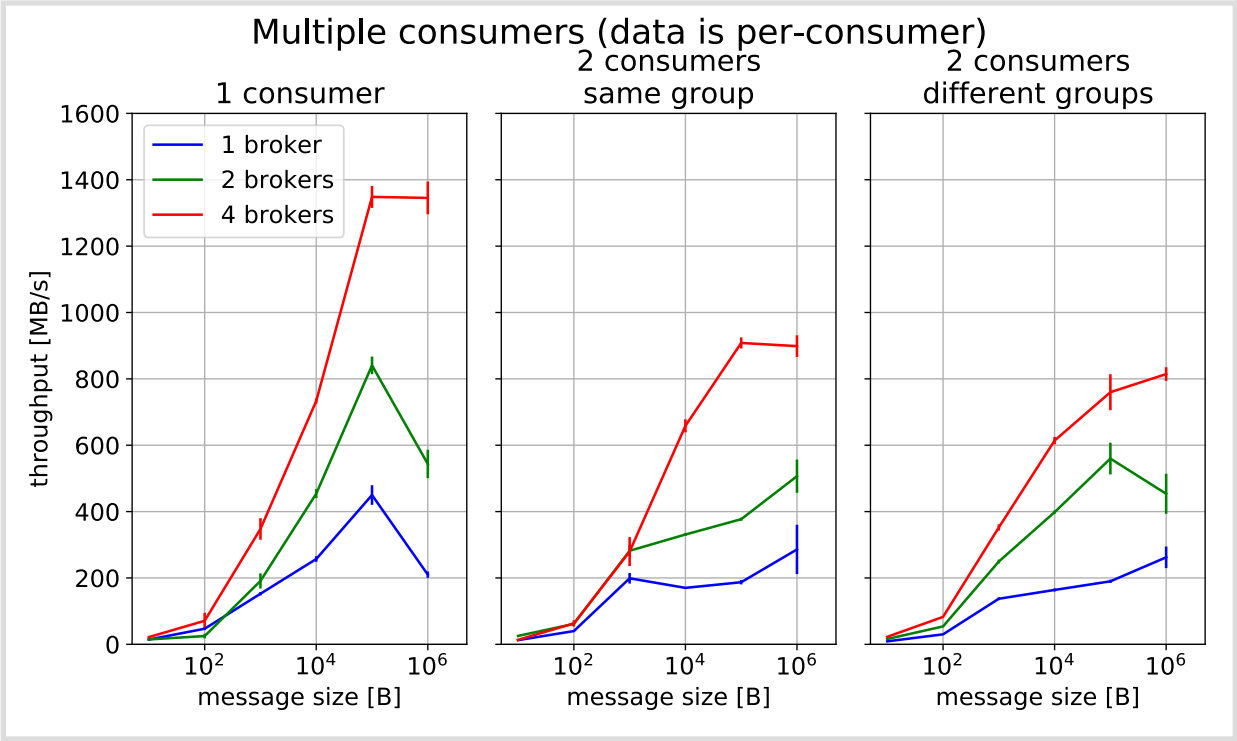

4 generators + filewriter
1 broker
2 brokers
4 brokers
1 p
2 p
4 p
throughput [MB/s]
message size [B]



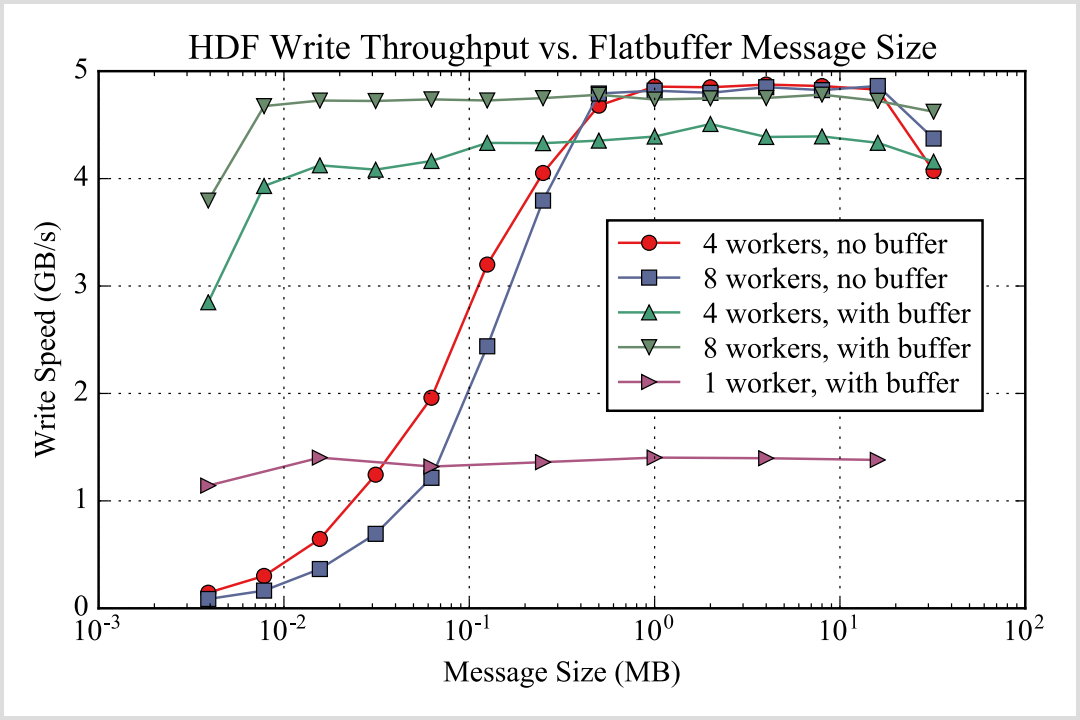
- Throughput tends to saturate around 1.4 GB/s
- The effect is more pronounced for small messages

PAUL SCHERRER INSTITUT

multiple consumers



- Performance of the single is reduced
- Aggregate throughput larger than single consumer

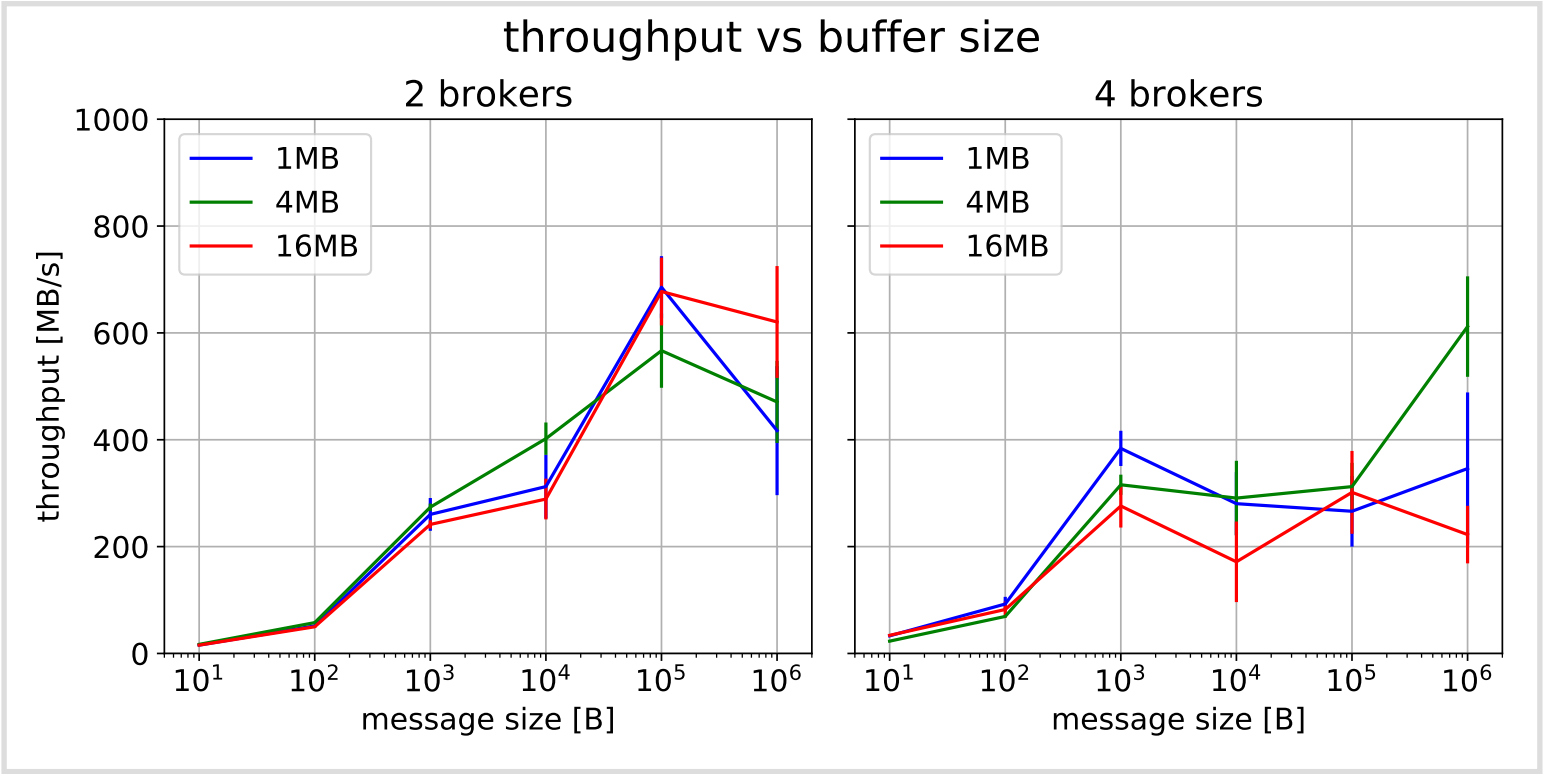


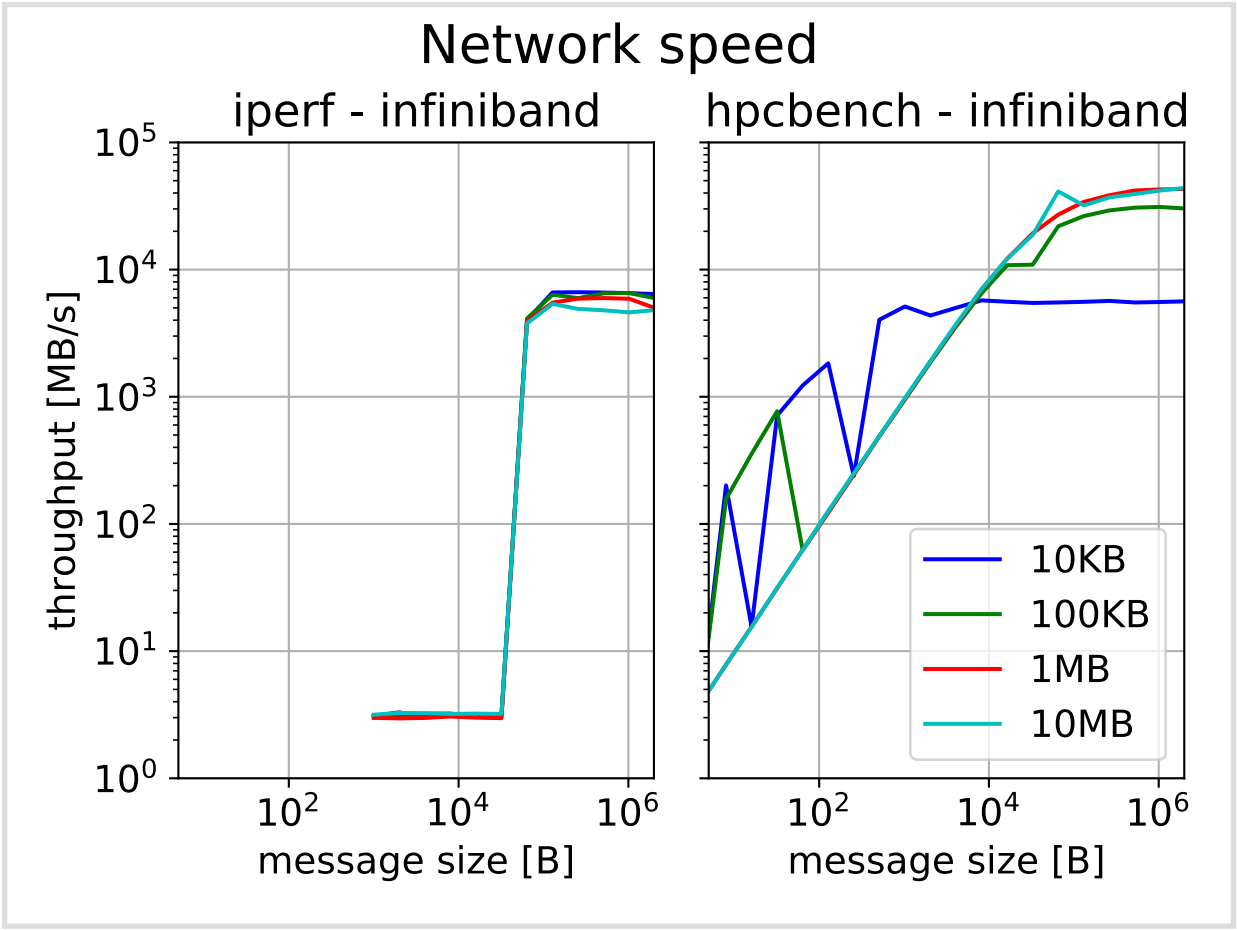
- Direct HDF5 writes: throughput decreases for small messages
- Buffered writes requires large number of workers to achieve the maximum throughput
- 1 worker data can be taken as a reference for the rest of this presentation

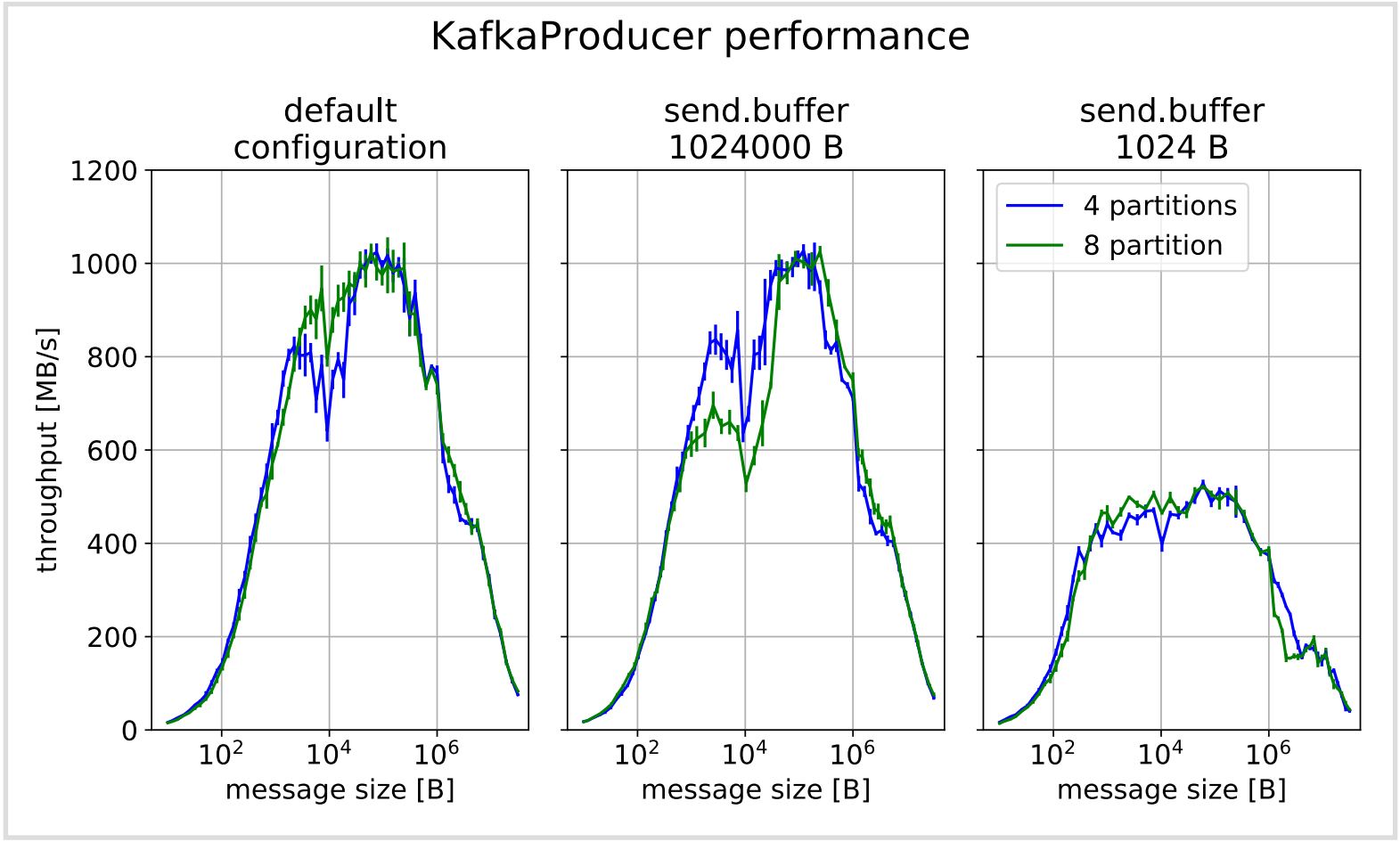
PAUL SCHERRER INSTITUT

To achieve good performances some tuning is required

```
"nexus": {
    "indices": {
        "index_every_kb": 4096
    },
    "chunk": {
        "chunk_n_elements": 2000000,
    },
    "buffer": {
        "size_kb": 4096,
        "packet_max_kb": 16
    }
}
```




Network speed
iperf - infiniband
hpcbench - infiniband
throughput [MB/s]
10^0
10^1
10^2
10^3
10^4
10^5
10^2
10^4
10^6
message size [B]
message size [B]
10KB
100KB
1MB
10MB


KafkaProducer performance
default configuration
send.buffer 1024000 B
send.buffer 1024 B
throughput [MB/s]
0
200
400
600
800
1000
1200
10^2
10^4
10^6
message size [B]
4 partitions
8 partition

Outline

The backbone of the data streaming chain is Apache Kafka
Provides a configurable number of persistent commit logs, scalability and redundancy.

- The data streaming toolchain
- Description of the system
- Producer performances
- Producer and consumer
- Producer, consumer and writer