

PAUL SCHERRER INSTITUT



brightness

WIR SCHAFFEN WISSEN - HEUTE FÜR MORGEN



Michele Brambilla :: Scientific Software Developer :: Paul Scherrer Institut
with Dominik Werder and Mark Koennecke

Performance Testing the BrightnESS toolchain

6th ECP Workshop

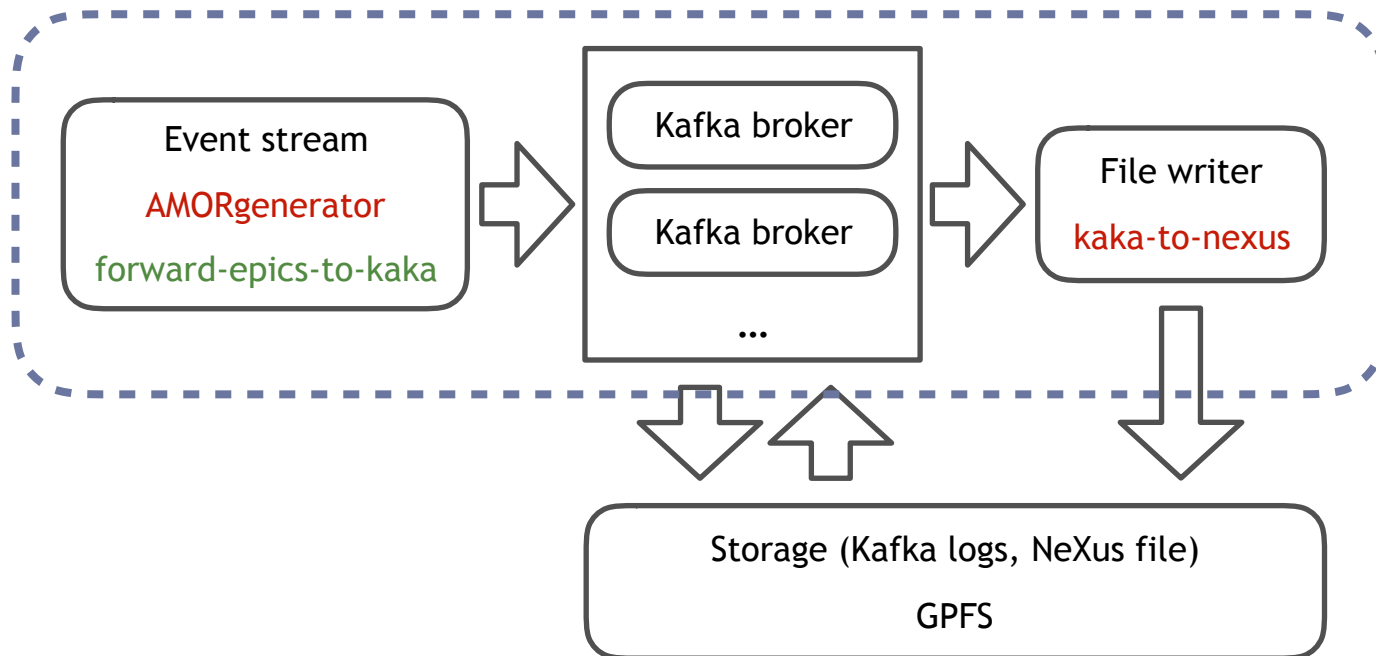
Outline

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

Testing the toolchain

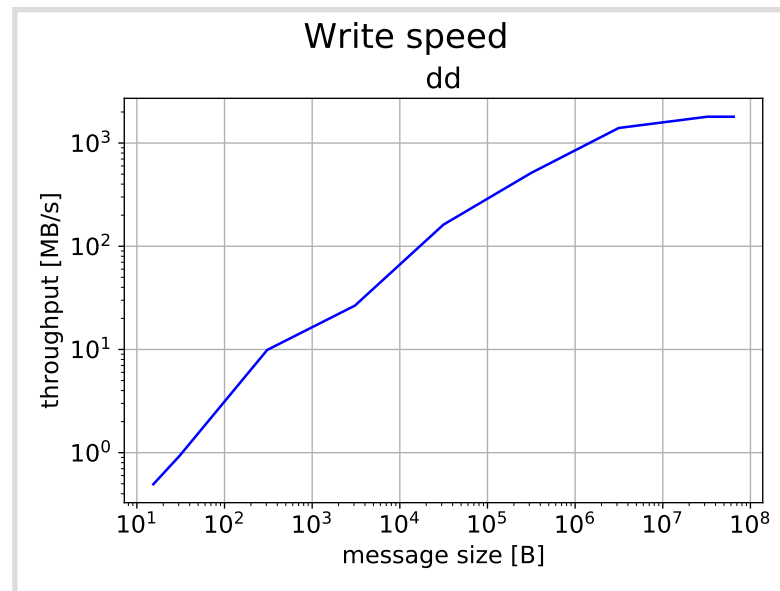
We want to test the performances of the whole toolchain from the **production** of an event stream to the **storage** of the nexus file on the disk



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

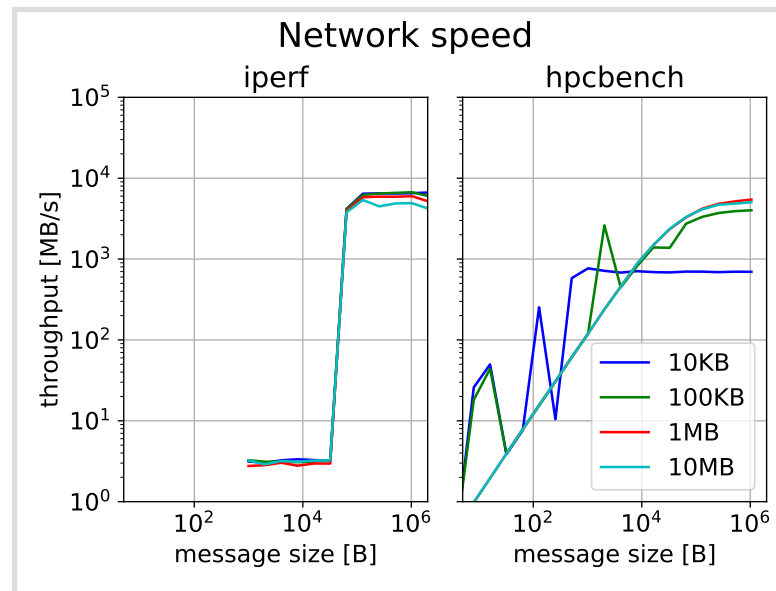


```
dd if=/dev/zero of=testfile bs=<msg size> count=100 oflag=direct
```

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

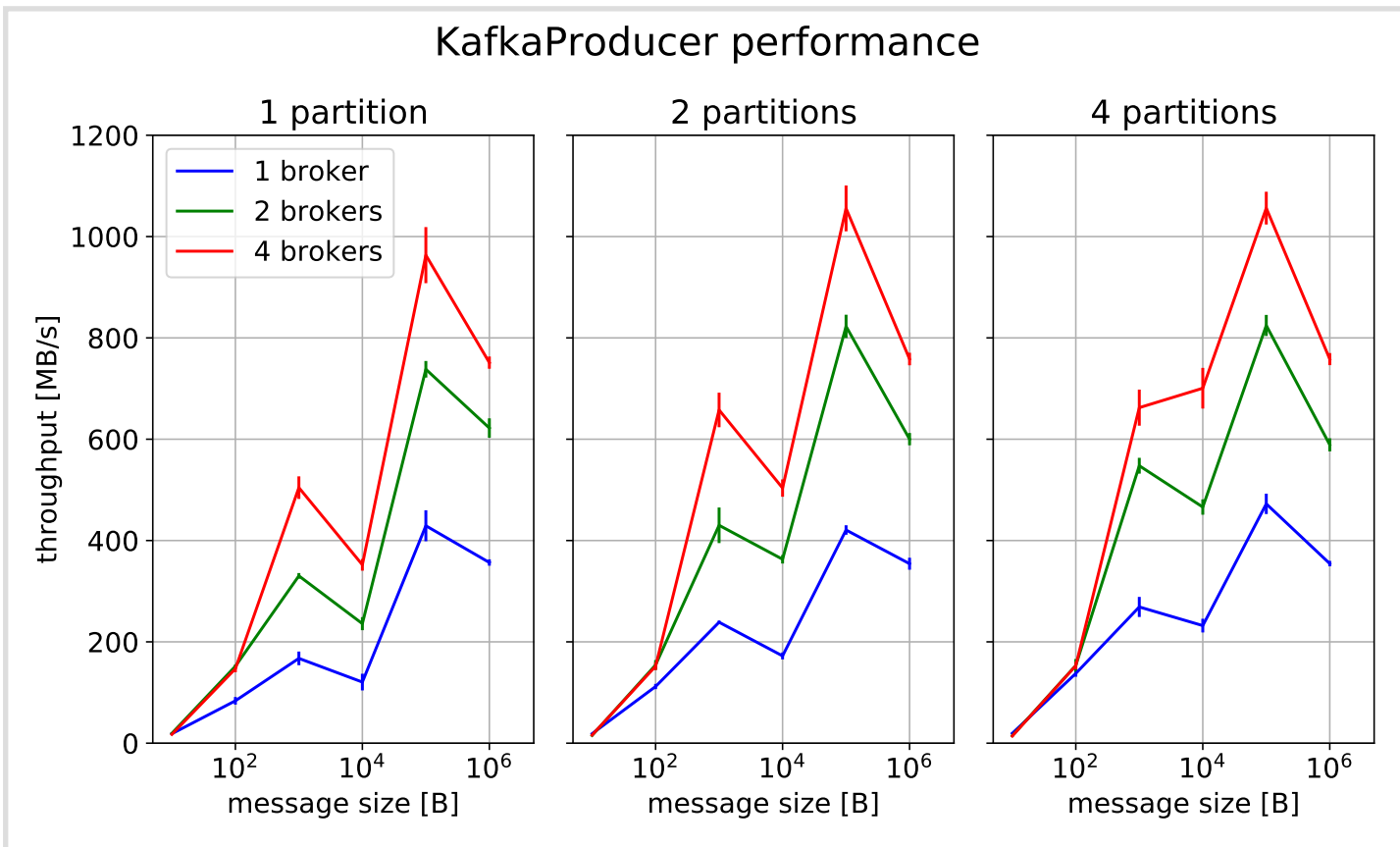
Same machine for **producer**, **broker** and **consumer**
File system shared with other users



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

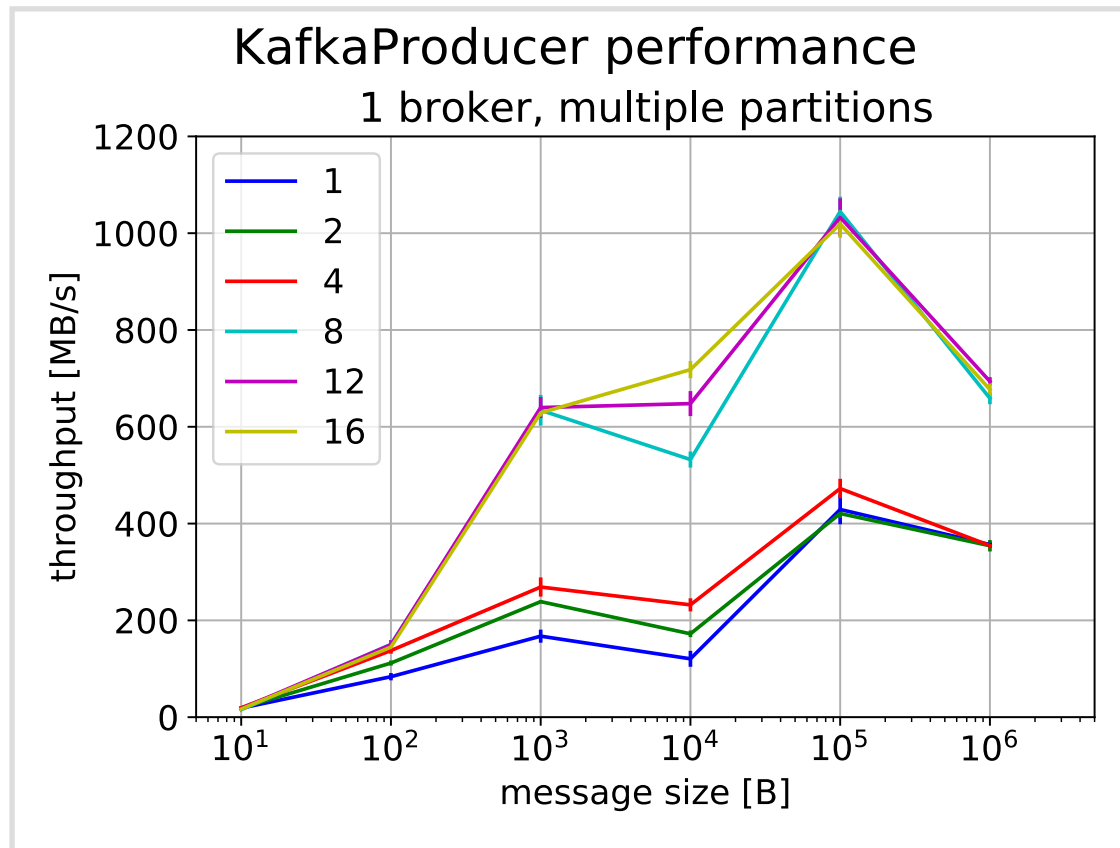
Kafka builtin tools

```
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]
```



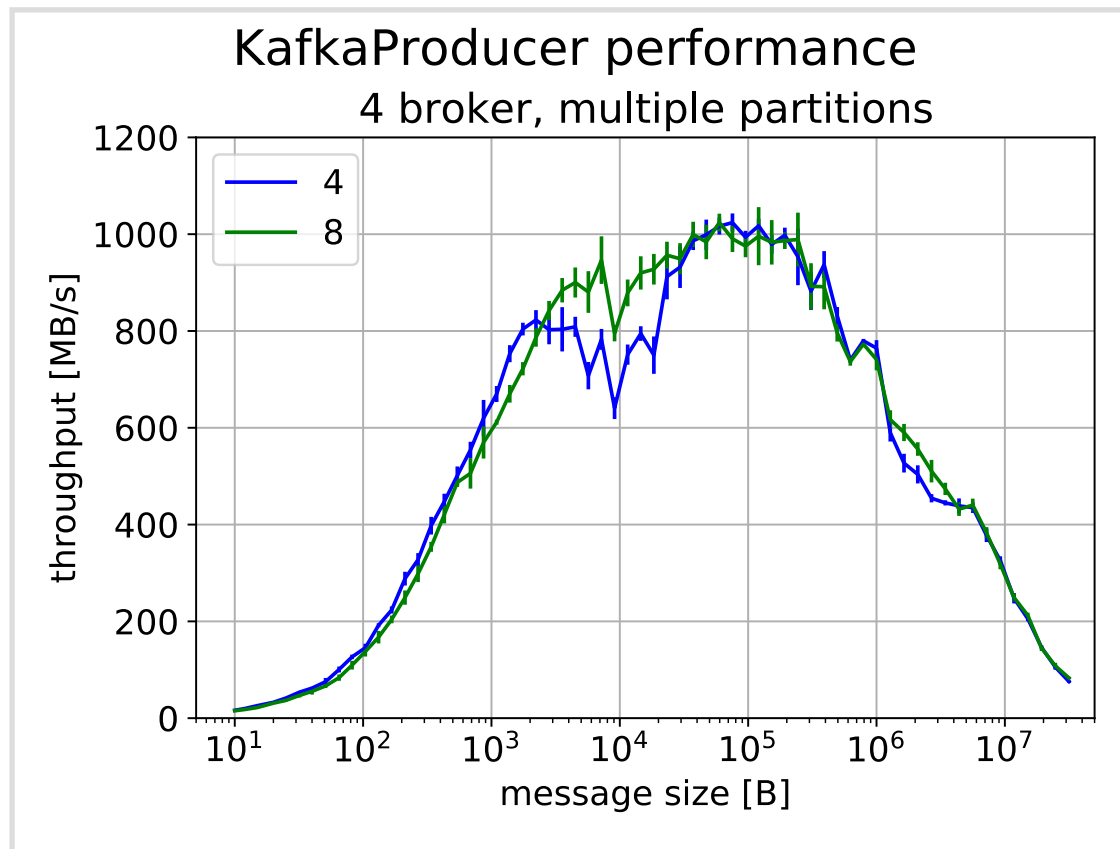
Kafka producer & neutron event generator

```
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]
```

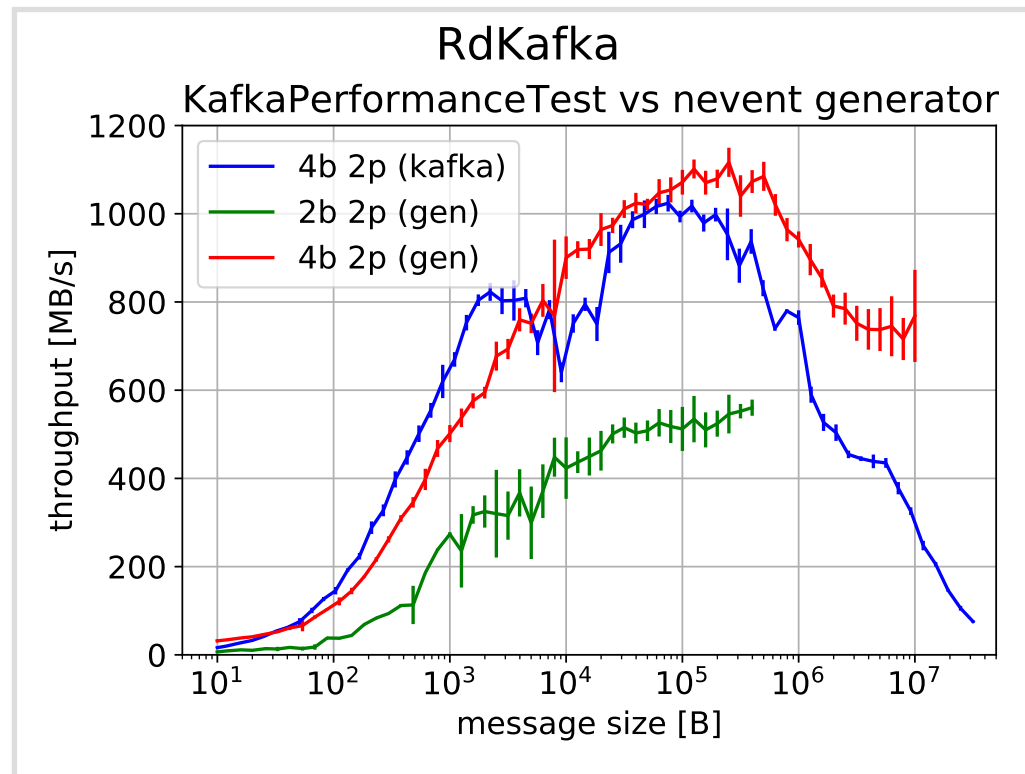


Kafka producer & neutron event generator

```
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]
```



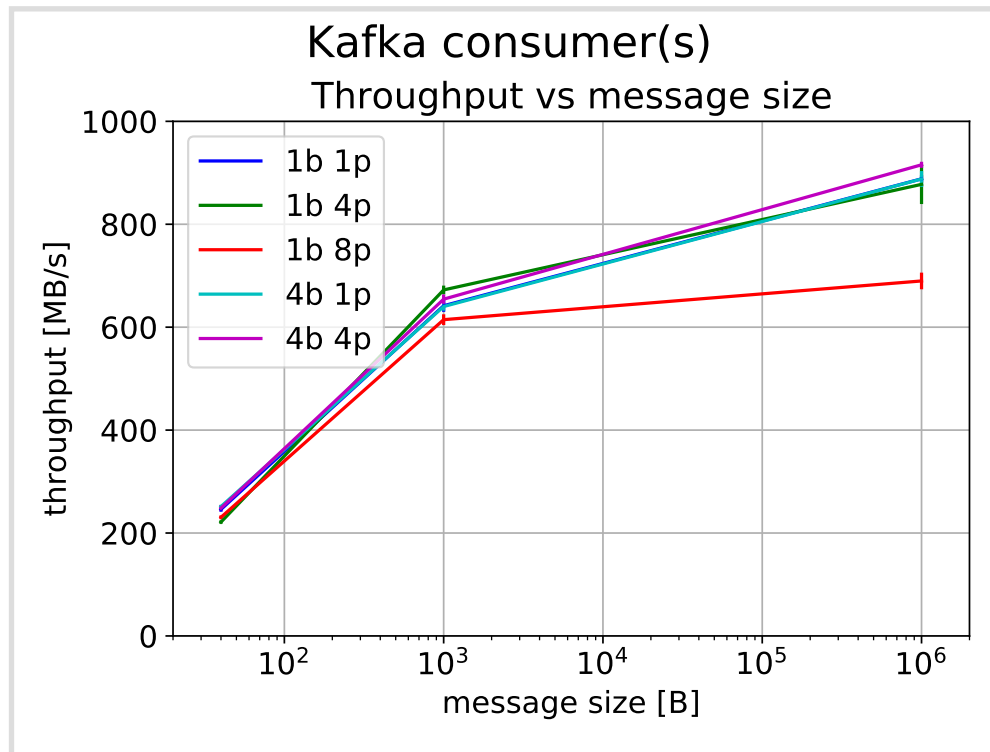
SINQ-AMORsim (neutron event generator)



- 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 builtin tools

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

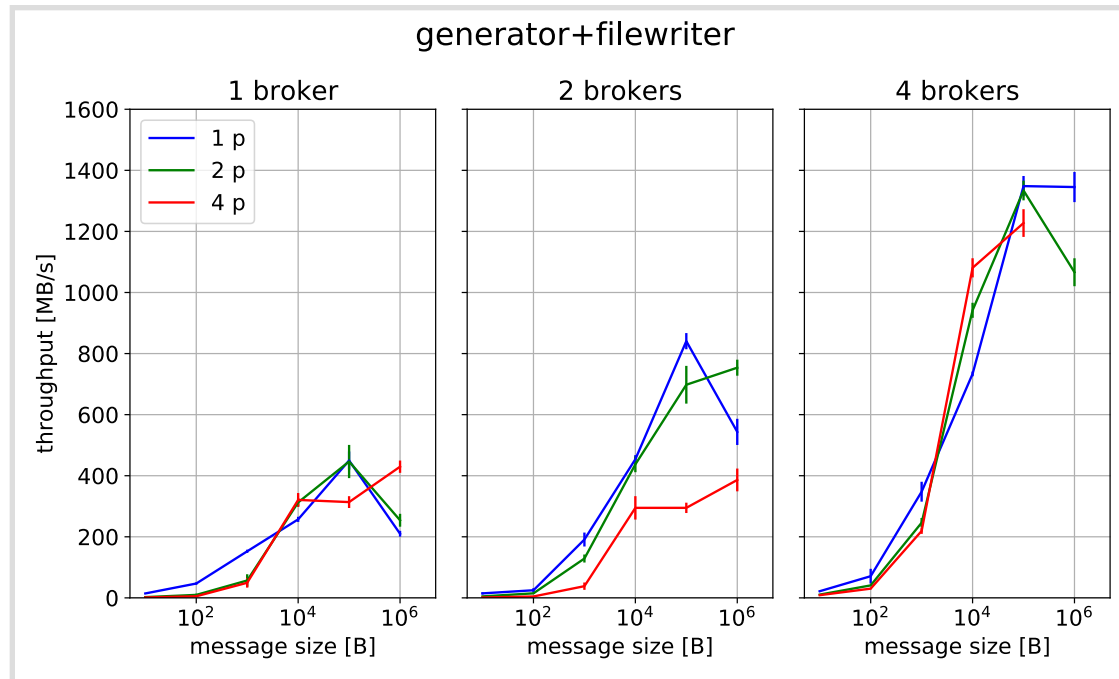


- Apparently independence on the number of brokers

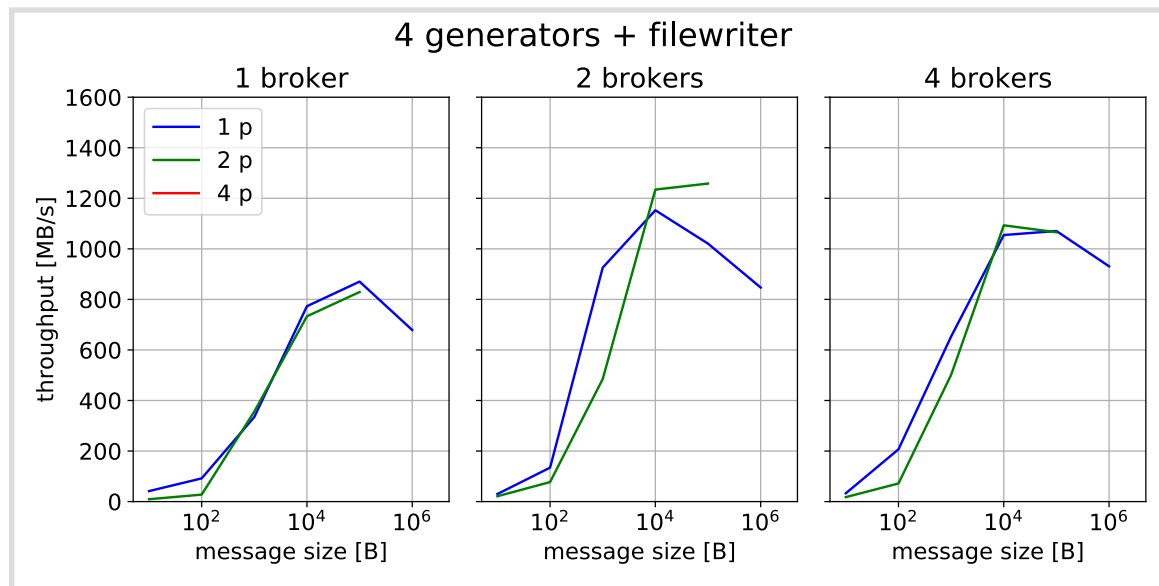
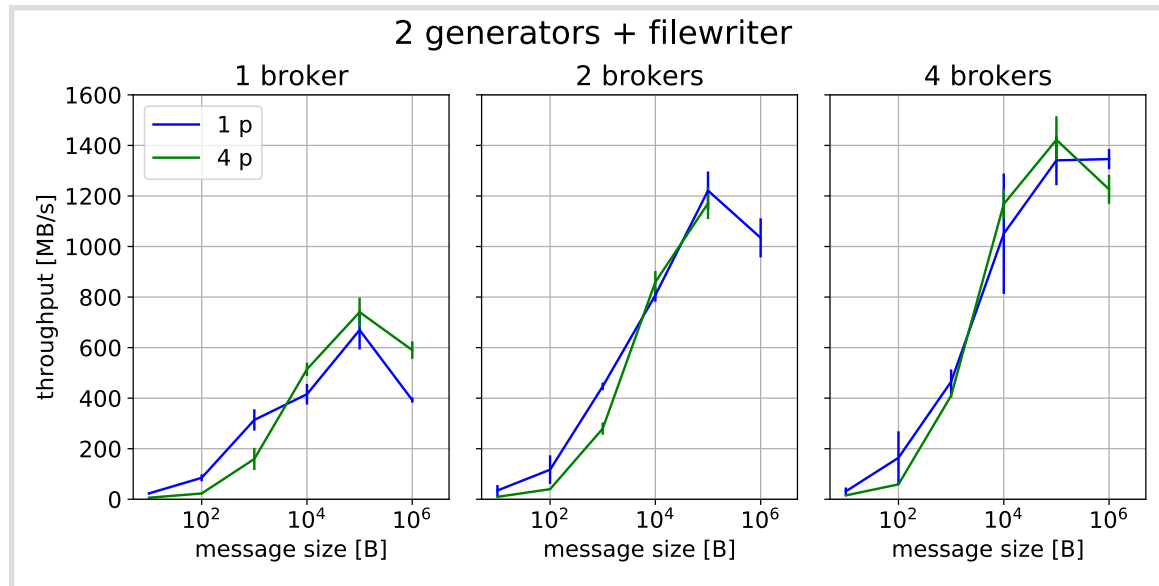
neutron event generator + file writer

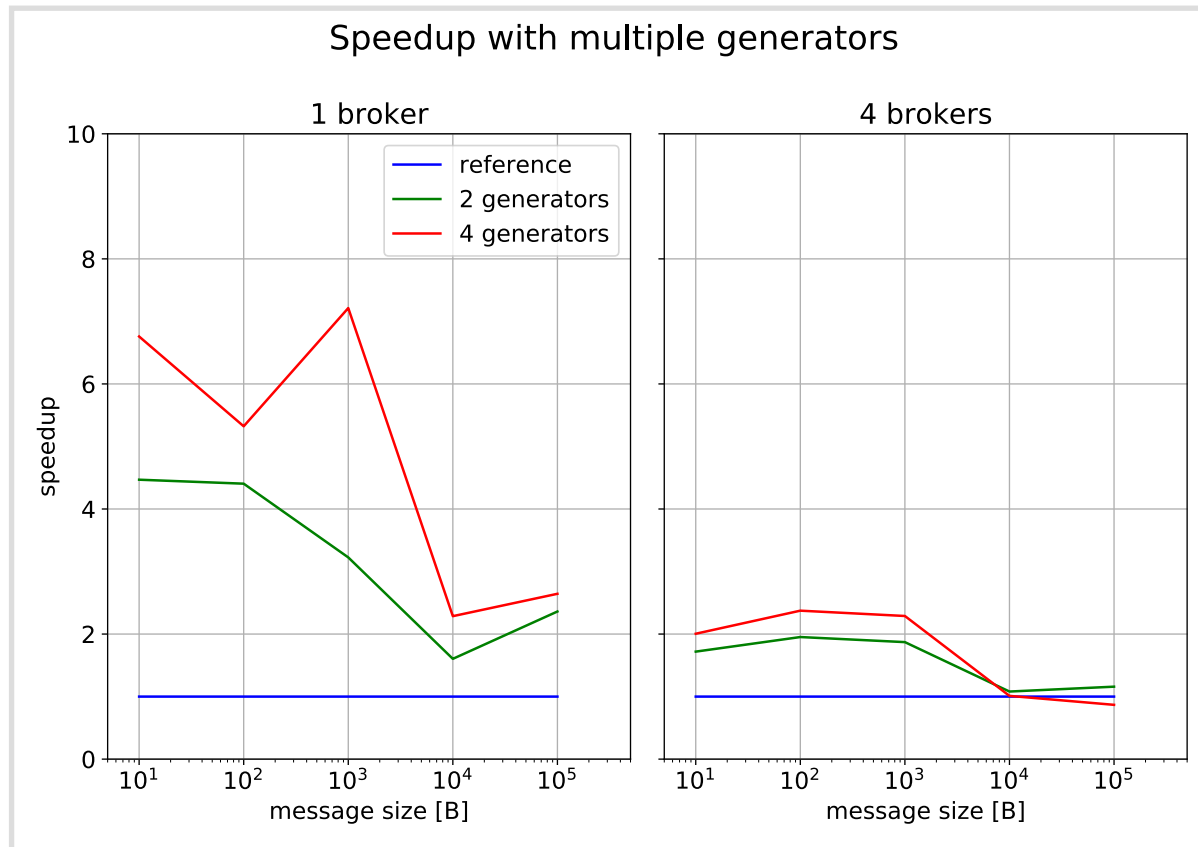
The file writer provides statistics about the consumed messages as Kafka logs. In particular:

- 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)



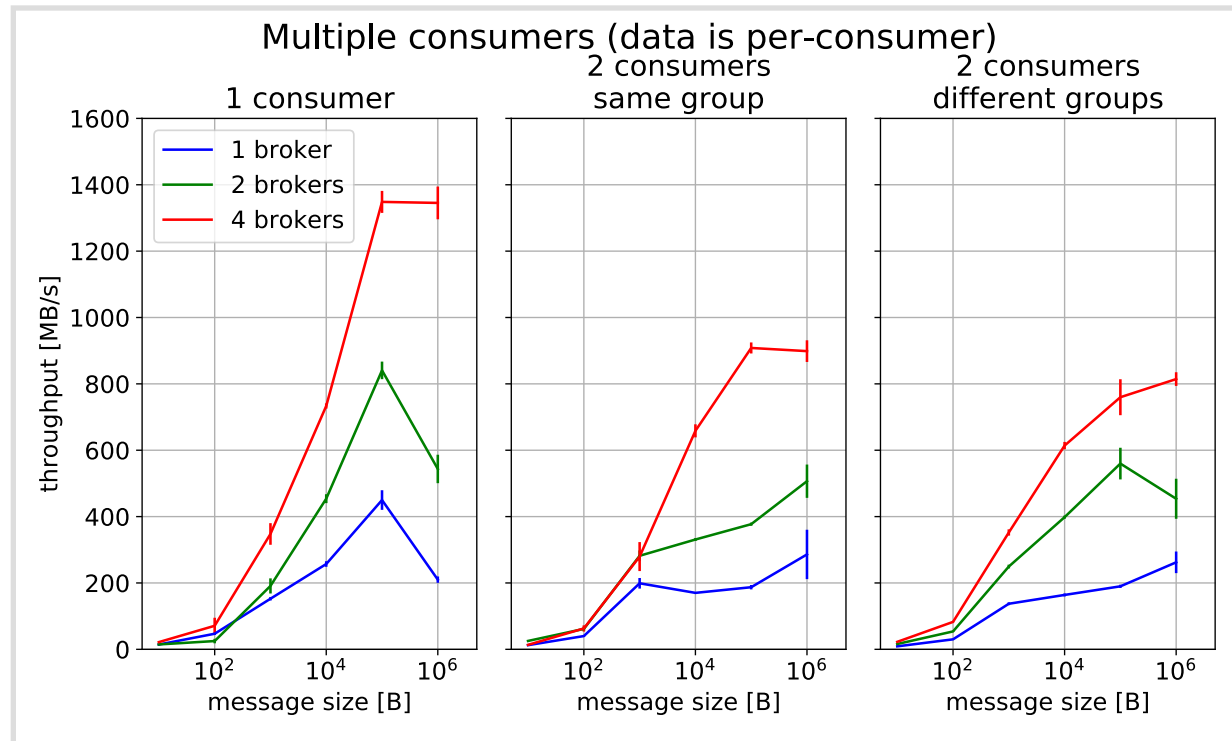
- Performance scales with number of brokers
- Independence on the number of partitions
- Better results *wrt* Kafka performance tools





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

multiple consumers

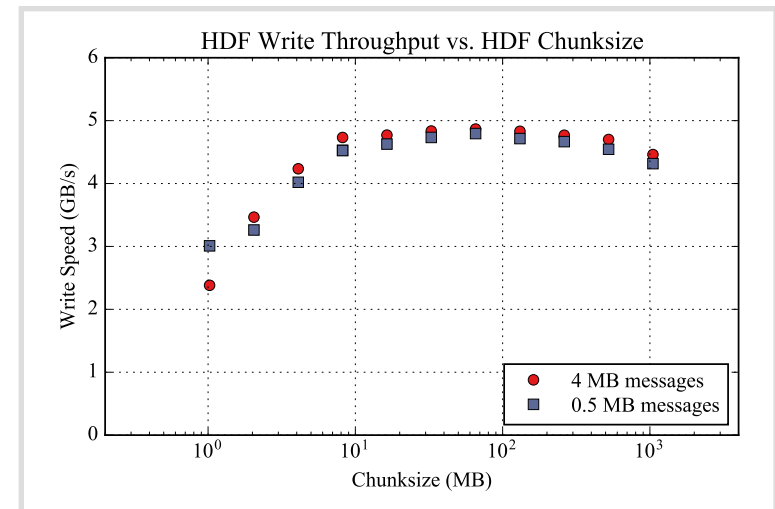
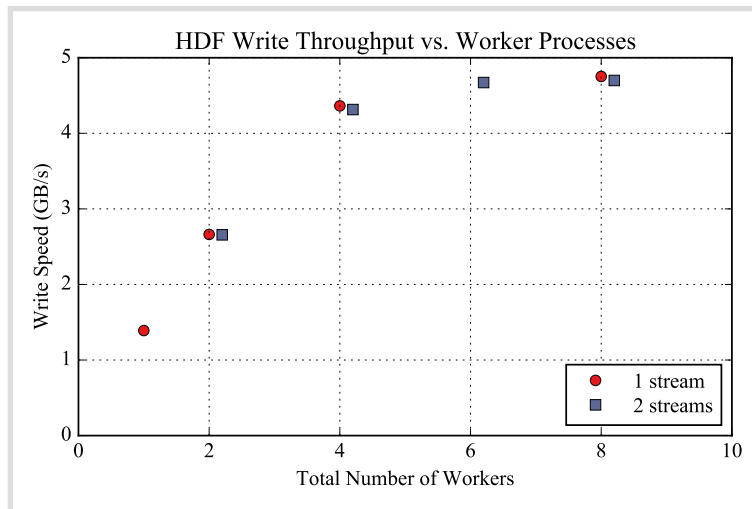


- Performance of the single is reduced
- Aggregate throughput larger than single consumer

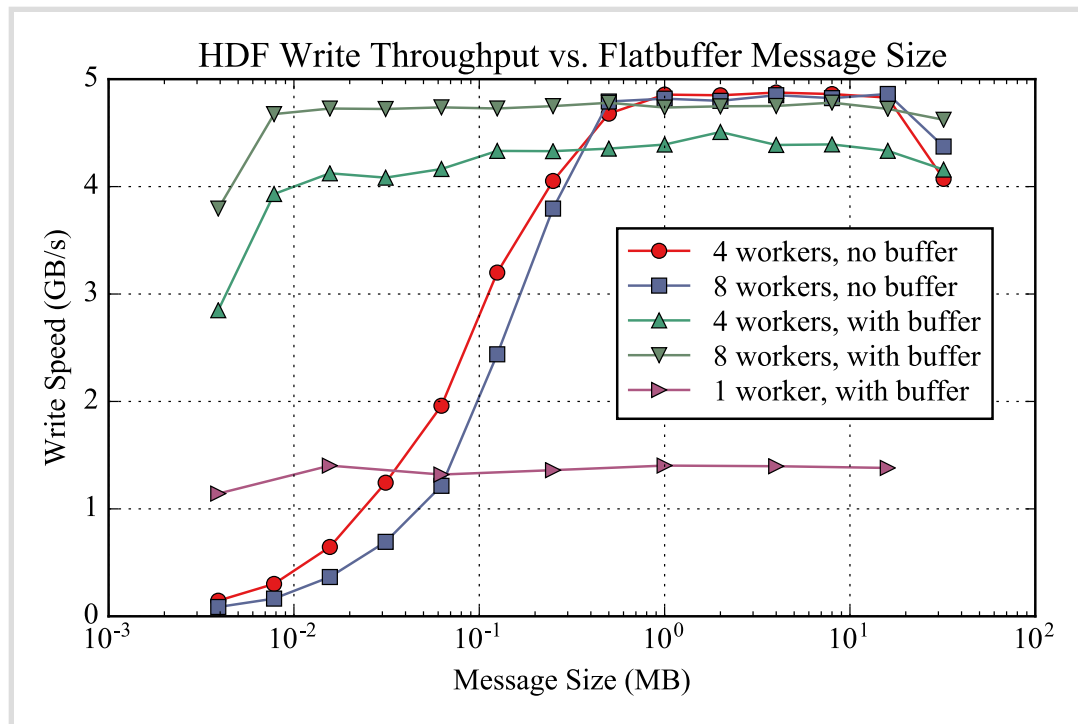
memory to HDF

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

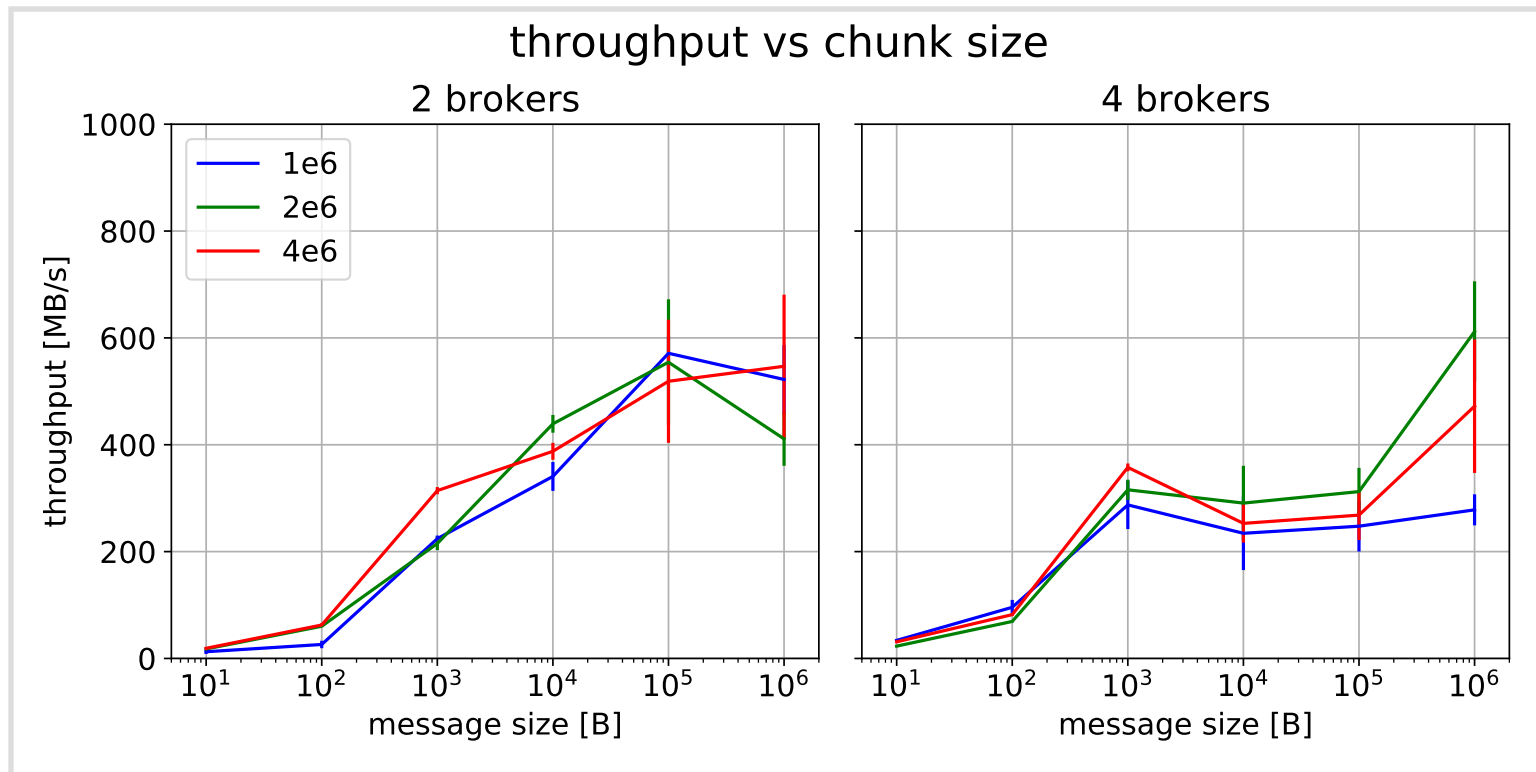


- Direct HDF5 writes: throughput decreases for small messages
- Buffered writes requires large number of workers to achieve the maximum throughput
- 1 worker data can be taken as a reference for the rest of this presentation

event stream to Kafka to HDF

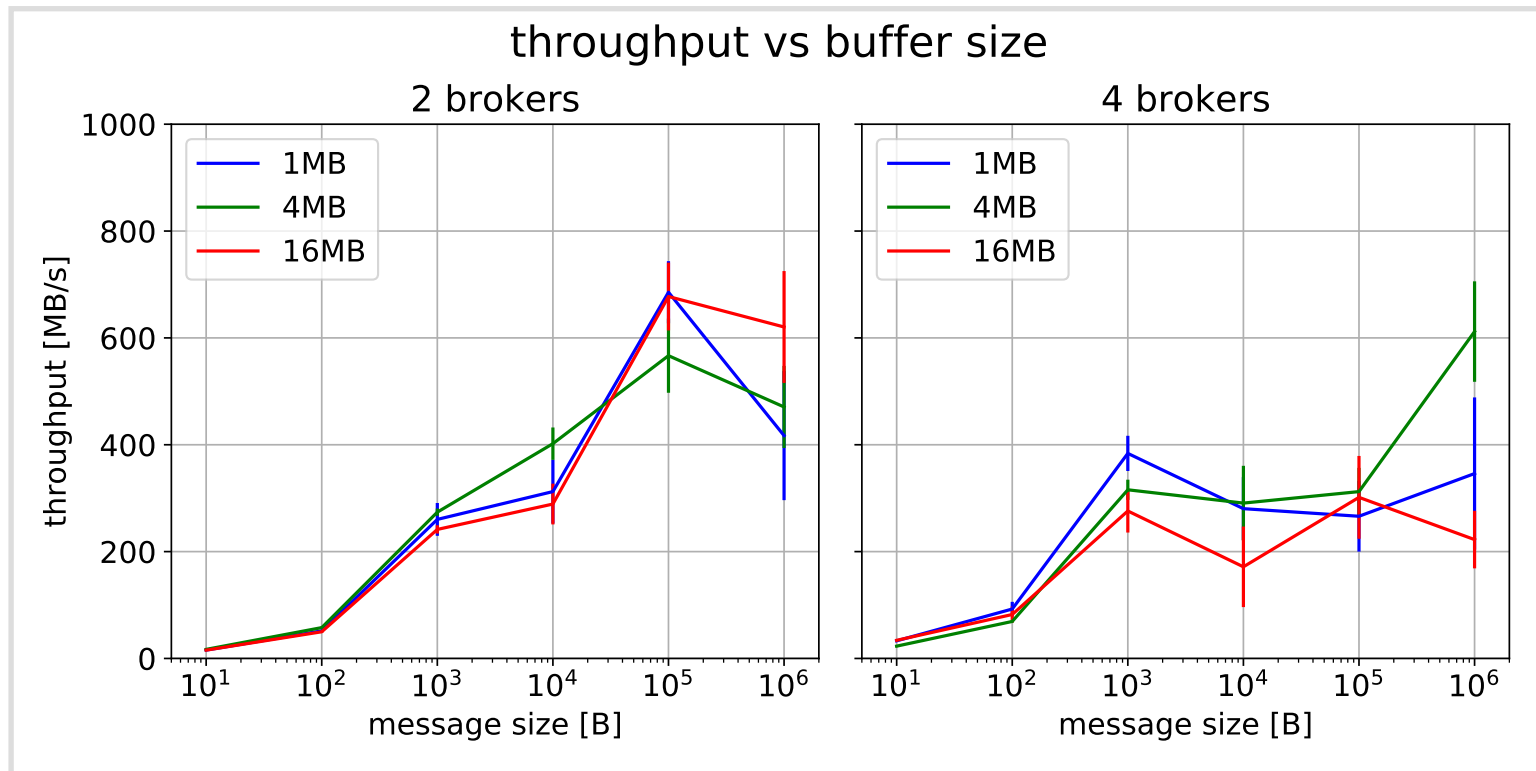
To achieve good performances some tuning is required

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



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
  }
}
```



Current status

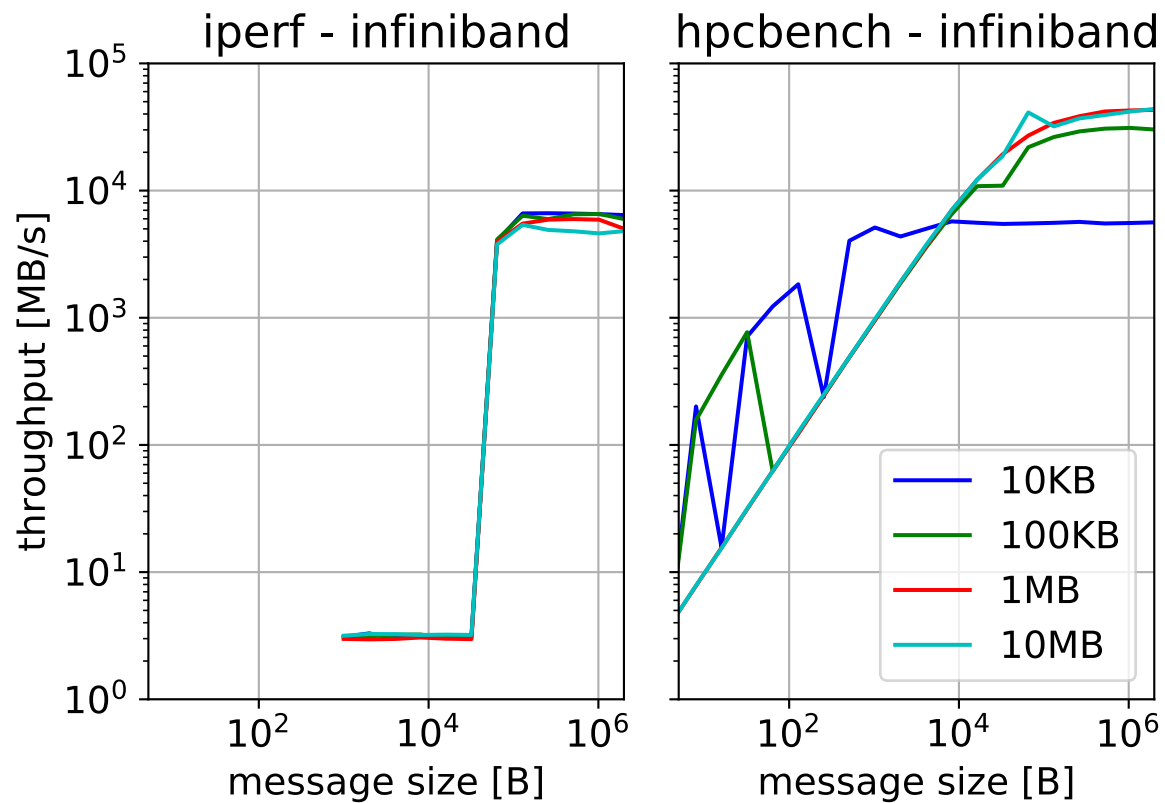
- On a single machine the producer and the consumer can reach 1.4 GB/s
- Single process writes up to 1.4 GB/s
- The full toolchain tested on a single machine with GPFS reaches 600 MB/s
- Dependence on the number of brokers

Ongoing development

- Parallel writer
- Parallel consuming
- Nikos integration

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

Network speed



KafkaProducer performance

