Streaming Architecture using Kafka

Mark Richards
Independent Consultant
Hands-on Software Architect / Published Author
Founder, DeveloperToArchitect.com
www.wmrichards.com

Author of *Software Architecture Fundamentals Video Series* (O’Reilly)
Author of Microservices Pitfalls and AntiPatterns (O’Reilly)
Author of Microservices vs. Service-Oriented Architecture (O’Reilly)
Author of *Enterprise Messaging Video Series* (O’Reilly)
Author of *Java Message Service 2nd Edition* (O’Reilly)
agenda

streaming architecture patterns
kafka overview
kafka producers and consumers
kafka vs. messaging
real-world examples of streaming data
source code

https://github.com/wmr513/streaming
Streaming Architecture Patterns
streaming architecture patterns
streaming architecture patterns

- capture and store
- filter and store
- analyze and store

service
Kafka Overview
kafka overview

publish and subscribe hybrid messaging model

messages are always persisted in a partitioned file by topics

message throughput can be upwards to 1,000,000+/sec

![Graph showing throughput vs. record size (bytes)]
kafka overview

topic structure

msg 1  msg 2  msg 3  msg 4  msg 5  msg 6  msg 7

producer

partition 0

consumer 1
offset 0

consumer 2
offset 1

...
kafka overview

topic structure

partition 0

| msg 1 | msg 2 | msg 4 | msg 8 | msg 9 | msg 11 | msg 17 |

partition 1

| msg 3 | msg 7 | msg 10 | msg 19 |

partition 2

| msg 5 | msg 6 | msg 12 | msg 13 | msg 14 | msg 15 | msg 16 | msg 18 |

producer
kafka overview

partitions and consumers

- topic 1
  - partition 0
  - partition 1
  - partition 2

- consumer group 1
  - consumer 1
kafka overview

partitions and consumers

- Topic 1
  - Partition 0
  - Partition 1
  - Partition 2

- Consumer group 1
  - Consumer 1
  - Consumer 2
  - Consumer 3
  - Consumer 4
kafka overview

core and streams api

kafka

core api
- KafkaProducer
- KafkaConsumer
- ProducerRecord
- ConsumerRecord

streams api
- StreamsBuilder
- StreamsConfig
- KStream
- KTable
Kafka Producers and Consumers
kafka producers and consumers
kafka producers and consumers
Properties props = new Properties();
props.put("bootstrap.servers", "localhost:9092");
props.put("key.serializer", "...kafka...StringSerializer");
props.put("value.serializer", "...kafka...StringSerializer");
KafkaProducer<String, String> producer =
    new KafkaProducer<String, String>(props);

String topic = "customer_comment_service_metrics";
String key = "duration";
String value = "320";
ProducerRecord<String, String> msg =
    new ProducerRecord<>(topic, key, value);
producer.send(msg);
producer.flush();
producer.close();

messages are sent in a batch within a separate thread
kafka producers and consumers
kafka producers and consumers
Properties props = new Properties();
props.put("bootstrap.servers", "localhost:9092");
props.put("group.id", "CG1");
props.put("key.deserializer", "...kafka...StringDeserializer");
props.put("value.deserializer", "...kafka...StringDeserializer");
KafkaConsumer<String, String> consumer =
    new KafkaConsumer<String, String>(props);

consumer.subscribe(Arrays.asList(
    "customer_comment_service_metrics"));
try {
    while (true) {
        ConsumerRecords<String, String> msgs = consumer.poll(100);
        for (ConsumerRecord<String, String> msg : msgs) {
            System.out.println("topic: "+ msg.topic());
            System.out.println("key: "+ msg.key());
            System.out.println("value: "+ msg.value());
            System.out.println("partition: "+ msg.partition());
            System.out.println("offset: "+ msg.offset());
        }
    }
} finally {
    consumer.close();
}

we are using auto commit of our offset sync point (5 sec)
try {
    while (true) {
        ConsumerRecords<String, String> msgs = consumer.poll(100);
        for (ConsumerRecord<String, String> msg : msgs) {
            System.out.println("topic: " + msg.topic());
            System.out.println("key: " + msg.key());
            System.out.println("value: " + msg.value());
            System.out.println("partition: " + msg.partition());
            System.out.println("offset: " + msg.offset());
        }
        try {
            consumer.commitSync();
        } catch (CommitFailedException e) {
            log.error("rats - I have no idea what to do now!");
        }
    }
} finally {
    consumer.close();
}
kafka producers and consumers
$ ./simpleconsumer.sh
waiting for messages...
Kafka vs. Messaging
kafka vs. standard messaging

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "properties": {
    "acct": {"type": "number"},
    "cusip": {"type": "string"},
    "shares": {"type": "number", "minimum": 100}
  },
  "required": ["acct", "cusip", "shares"]
}
```
kafka vs. standard messaging

unbounded continuous flow of data

distinct bounded messages
kafka vs. standard messaging

Key vs. Value

Throughput up to 1 million messages/sec

Throughput up to 4K/10K messages/sec
kafka vs. standard messaging

- **kafka**: good for operational data
- **ActiveMQ, RabbitMQ**: good for transactional data
kafka vs. standard messaging

- **kafka**
  - producer → topic
  - pub/sub

- **ActiveMQ, RabbitMQ**
  - producer → queue
  - point-to-point

- **Exchange**
  - producer → exchange
  - exchange
  - pub/sub
  - queue → consumer
Real-World Examples of Streaming Data
microservices metrics analytics
microservices metrics analytics

placing trade: BUY AAPL 3907 SHARES
placing trade: BUY GOOG 1790 SHARES
trade error: BUY AAPL 3177 SHARES
placing trade: BUY ATT 371 SHARES
placing trade: BUY ATT 2487 SHARES
placing trade: BUY AAPL 781 SHARES
trade error: BUY AAPL 506 SHARES
placing trade: BUY AAPL 3109 SHARES
placing trade: BUY AAPL 3935 SHARES

validating trade: BUY AAPL 3841 SHARES
validating trade: BUY AAPL 2834 SHARES
validating trade: BUY AAPL 3617 SHARES
validating trade: BUY AAPL 2348 SHARES
validating trade: BUY IBM 3039 SHARES
validating trade: BUY ATT 673 SHARES
validating trade: BUY AAPL 2258 SHARES
validating trade: BUY AAPL 2535 SHARES
validating trade: BUY IBM 447 SHARES
microservices metrics analytics

placing trade: BUY AAPL 3907 SHARES
placing trade: BUY GOOG 1790 SHARES
trade error: BUY AAPL 3177 SHARES
placing trade: BUY ATT 371 SHARES
placing trade: BUY ATT 2487 SHARES
placing trade: BUY AAPL 781 SHARES
trade error: BUY AAPL 506 SHARES
placing trade: BUY AAPL 3109 SHARES
placing trade: BUY AAPL 3935 SHARES

trade_gen_service_metrics

validating trade: BUY AAPL 3841 SHARES
validating trade: BUY AAPL 2834 SHARES
validating trade: BUY AAPL 3617 SHARES
validating trade: BUY AAPL 2348 SHARES
validating trade: BUY IBM 3039 SHARES
validating trade: BUY ATT 673 SHARES
validating trade: BUY AAPL 2258 SHARES
validating trade: BUY AAPL 2535 SHARES
validating trade: BUY IBM 447 SHARES

trade_validation_service_metrics

_metrics(5797): duration.req = 28
_metrics(5798): duration.min = 711
_metrics(5799): duration.max = 117
_metrics(5800): variance.req = 3864
_metrics(5801): variance.min = 295
_metrics(5802): variance.max = 583
_metrics(5803): stddev.req = 59
_metrics(5804): stddev.min = 5
_metrics(5805): stddev.max = 82
_metrics(5806): duration.90th = 57
_metrics(5807): duration.95th = 494
_metrics(5808): duration.99th = 808
microservices metrics analytics

_metrics(5797): duration.req = 28
_metrics(5798): duration.min = 711
_metrics(5799): duration.max = 117
_metrics(5800): variance.req = 3864
_metrics(5801): variance.min = 295
_metrics(5802): variance.max = 503
_metrics(5803): stddev.req = 59
_metrics(5804): stddev.min = 5
_metrics(5805): stddev.max = 82
_metrics(5806): duration.90th = 57
_metrics(5807): duration.95th = 494
_metrics(5808): duration.99th = 808
microservices metrics analytics

placing trade: BUY AAPL 3907 SHARES
placing trade: BUY GOOG 1790 SHARES
trade error: BUY AAPL 3177 SHARES
placing trade: BUY ATT 371 SHARES
placing trade: BUY ATT 2487 SHARES
placing trade: BUY AAPL 781 SHARES
trade error: BUY AAPL 506 SHARES
placing trade: BUY AAPL 3109 SHARES
placing trade: BUY AAPL 3935 SHARES

validating trade: BUY AAPL 3841 SHARES
validating trade: BUY AAPL 2834 SHARES
validating trade: BUY AAPL 3617 SHARES
validating trade: BUY AAPL 2348 SHARES
validating trade: BUY IBM 3039 SHARES
validating trade: BUY ATT 673 SHARES
validating trade: BUY AAPL 2258 SHARES
validating trade: BUY AAPL 2535 SHARES
validating trade: BUY IBM 447 SHARES

trade_gen_service_metrics

<table>
<thead>
<tr>
<th>metric</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>duration.req</td>
<td>28</td>
</tr>
<tr>
<td>duration.min</td>
<td>711</td>
</tr>
<tr>
<td>duration.max</td>
<td>117</td>
</tr>
<tr>
<td>variance.req</td>
<td>3864</td>
</tr>
<tr>
<td>variance.min</td>
<td>295</td>
</tr>
<tr>
<td>variance.max</td>
<td>583</td>
</tr>
<tr>
<td>stddev.req</td>
<td>50</td>
</tr>
<tr>
<td>stddev.min</td>
<td>5</td>
</tr>
<tr>
<td>stddev.max</td>
<td>82</td>
</tr>
<tr>
<td>duration.90th</td>
<td>57</td>
</tr>
<tr>
<td>duration.95th</td>
<td>494</td>
</tr>
<tr>
<td>duration.99th</td>
<td>808</td>
</tr>
</tbody>
</table>

trade_validation_service_metrics

<table>
<thead>
<tr>
<th>metric</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>response exceeded</td>
<td>90ms: 99</td>
</tr>
<tr>
<td>response exceeded</td>
<td>90ms: 92</td>
</tr>
<tr>
<td>response exceeded</td>
<td>90ms: 96</td>
</tr>
<tr>
<td>response exceeded</td>
<td>90ms: 91</td>
</tr>
<tr>
<td>response exceeded</td>
<td>90ms: 98</td>
</tr>
<tr>
<td>response exceeded</td>
<td>90ms: 98</td>
</tr>
<tr>
<td>response exceeded</td>
<td>90ms: 99</td>
</tr>
<tr>
<td>response exceeded</td>
<td>90ms: 92</td>
</tr>
<tr>
<td>response exceeded</td>
<td>90ms: 96</td>
</tr>
<tr>
<td>response exceeded</td>
<td>90ms: 97</td>
</tr>
<tr>
<td>response exceeded</td>
<td>90ms: 99</td>
</tr>
<tr>
<td>response exceeded</td>
<td>90ms: 92</td>
</tr>
<tr>
<td>metrics</td>
<td>response exceeded</td>
</tr>
<tr>
<td>------------</td>
<td>------------------</td>
</tr>
<tr>
<td>530</td>
<td></td>
</tr>
<tr>
<td>531</td>
<td></td>
</tr>
<tr>
<td>532</td>
<td></td>
</tr>
<tr>
<td>533</td>
<td></td>
</tr>
<tr>
<td>534</td>
<td></td>
</tr>
<tr>
<td>535</td>
<td></td>
</tr>
<tr>
<td>536</td>
<td></td>
</tr>
<tr>
<td>537</td>
<td></td>
</tr>
<tr>
<td>538</td>
<td></td>
</tr>
<tr>
<td>539</td>
<td></td>
</tr>
<tr>
<td>540</td>
<td></td>
</tr>
<tr>
<td>541</td>
<td></td>
</tr>
<tr>
<td>542</td>
<td></td>
</tr>
</tbody>
</table>
microservices metrics analytics
| _errors(401) | BUY AAPL 1686 SHARES |
| _errors(402) | BUY AAPL 1069 SHARES |
| _errors(403) | BUY IBM 634 SHARES |
| _errors(404) | BUY AAPL 2404 SHARES |
| _errors(405) | BUY ATT 3501 SHARES |
| _errors(406) | BUY GOOG 1823 SHARES |
| _errors(407) | BUY GOOG 1163 SHARES |
| _errors(408) | BUY ATT 238 SHARES |
| _errors(409) | BUY AAPL 3736 SHARES |
| _errors(410) | BUY AAPL 1298 SHARES |
| _errors(411) | BUY AAPL 2429 SHARES |
microservices metrics analytics

placing trade: BUY AAPL 3907 SHARES
placing trade: BUY GOOG 1790 SHARES
trade error: BUY AAPL 3177 SHARES
placing trade: BUY ATT 371 SHARES
placing trade: BUY AAPL 781 SHARES
trade error: BUY AAPL 506 SHARES
placing trade: BUY AAPL 3189 SHARES
placing trade: BUY AAPL 3935 SHARES

trade_gen_service_symbol

trade_gen_service_metrics

贸易生成服务错误

trade_gen_service_error

贸易验证服务指标

trade_validation_service_metrics

 validating trade: BUY AAPL 3841 SHARES
validating trade: BUY AAPL 2834 SHARES
validating trade: BUY AAPL 3617 SHARES
validating trade: BUY AAPL 2348 SHARES
validating trade: BUY IBM 3039 SHARES
validating trade: BUY ATT 673 SHARES
validating trade: BUY AAPL 2258 SHARES
validating trade: BUY AAPL 2535 SHARES
validating trade: BUY IBM 447 SHARES

_errors(401): BUY AAPL 1686 SHARES
_errors(402): BUY AAPL 1069 SHARES
_errors(403): BUY IBM 634 SHARES
_errors(404): BUY AAPL 2404 SHARES
_errors(405): BUY ATT 3581 SHARES
_errors(406): BUY GOOG 1823 SHARES
_errors(407): BUY GOOG 1163 SHARES
_errors(408): BUY ATT 236 SHARES
_errors(409): BUY AAPL 3736 SHARES
_errors(410): BUY AAPL 1298 SHARES
_errors(411): BUY AAPL 2429 SHARES

AAPL=668, GOOG=221, IBM=253, ATT=464, TOTAL=1817
AAPL=668, GOOG=221, IBM=253, ATT=465, TOTAL=1819
AAPL=669, GOOG=221, IBM=254, ATT=465, TOTAL=1820
AAPL=670, GOOG=221, IBM=254, ATT=465, TOTAL=1821
AAPL=671, GOOG=221, IBM=254, ATT=465, TOTAL=1822
AAPL=671, GOOG=221, IBM=254, ATT=466, TOTAL=1824
AAPL=672, GOOG=221, IBM=254, ATT=466, TOTAL=1825
AAPL=672, GOOG=221, IBM=254, ATT=467, TOTAL=1827
AAPL=672, GOOG=221, IBM=254, ATT=467, TOTAL=1828
AAPL=672, GOOG=221, IBM=255, ATT=467, TOTAL=1828
AAPL=672, GOOG=221, IBM=255, ATT=467, TOTAL=1829
microservices metrics analytics

AAPL=668, G00G=221, IBM=253, ATT=464, TOTAL=1817
AAPL=668, G00G=221, IBM=253, ATT=465, TOTAL=1819
AAPL=669, G00G=221, IBM=253, ATT=465, TOTAL=1820
AAPL=669, G00G=221, IBM=254, ATT=465, TOTAL=1820
AAPL=670, G00G=221, IBM=254, ATT=465, TOTAL=1821
AAPL=671, G00G=221, IBM=254, ATT=465, TOTAL=1822
AAPL=671, G00G=221, IBM=254, ATT=466, TOTAL=1824
AAPL=672, G00G=221, IBM=254, ATT=466, TOTAL=1825
AAPL=672, G00G=221, IBM=254, ATT=467, TOTAL=1827
AAPL=672, G00G=222, IBM=254, ATT=467, TOTAL=1828
AAPL=672, G00G=222, IBM=255, ATT=467, TOTAL=1828
AAPL=673, G00G=222, IBM=255, ATT=467, TOTAL=1829
microservices metrics analytics

placing trade: BUY AAPL 3907 SHARES
placing trade: BUY GOOG 1790 SHARES
trade error: BUY ATT 371 SHARES
placing trade: BUY AAPL 781 SHARES
trade error: BUY AAPL 506 SHARES
placing trade: BUY AAPL 3109 SHARES
placing trade: BUY AAPL 3935 SHARES

trade_gen_service_error

trade_gen_service_symbol

trade_gen_service_metrics

 validations trade: BUY AAPL 3841 SHARES
validating trade: BUY AAPL 2834 SHARES
validating trade: BUY AAPL 3617 SHARES
validating trade: BUY AAPL 2348 SHARES
validating trade: BUY IBM 3039 SHARES
validating trade: BUY ATT 673 SHARES
validating trade: BUY AAPL 2258 SHARES
validating trade: BUY AAPL 2535 SHARES
validating trade: BUY IBM 447 SHARES

trade_validation_service_metrics
Summary
Streaming Architecture using Kafka

Mark Richards
Independent Consultant
Hands-on Software Architect
Published Author / Conference Speaker
www.wmrichards.com

Author of *Software Architecture Fundamentals Video Series* (O'Reilly)
Author of Microservices Pitfalls and AntiPatterns (O’Reilly)
Author of Microservices vs. Service-Oriented Architecture (O’Reilly)
Author of *Enterprise Messaging Video Series* (O'Reilly)
Author of *Java Message Service 2nd Edition* (O’Reilly)
GREAT INDIAN DEVELOPER SUMMIT 2019
Conference: April 23-26, Bangalore

Register early and get the best discounts!

www.developersummit.com @greatindiandev bit.ly/gidslinkedin