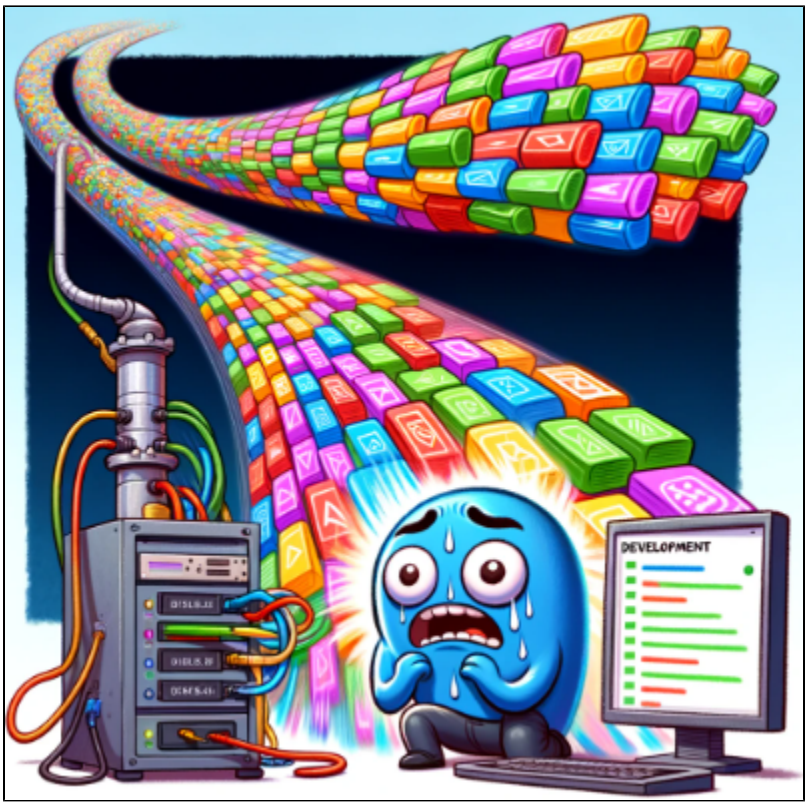
- API

, .
(API) BackPresure .

API



- : API
- : API ~

API API .

- TPS ?
- TPS ?

API . .

- TPS ?
- ?

TPS .

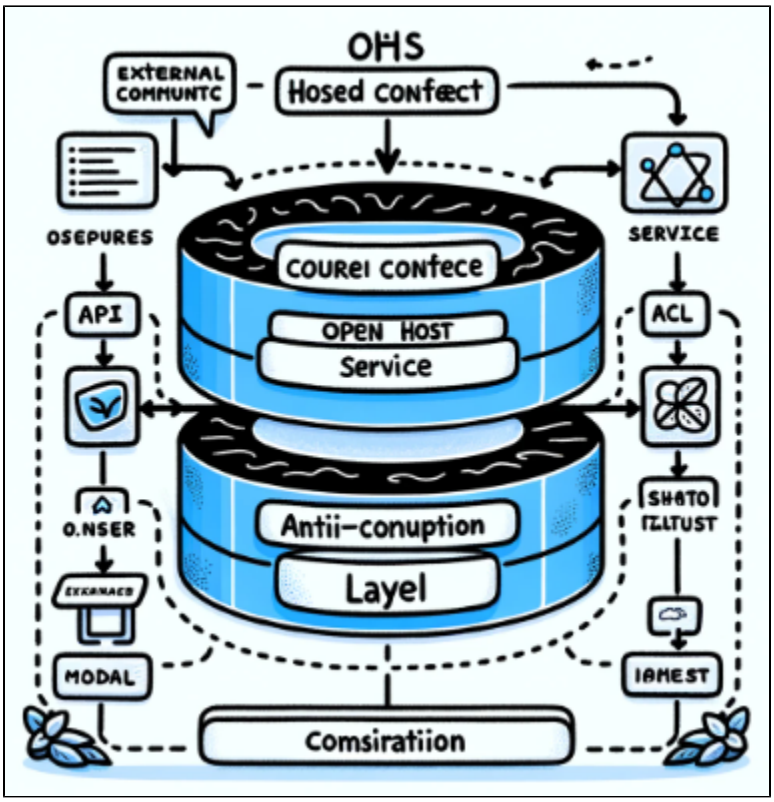
TPS .

TPS

API OHS(Open Host Service) TPS .

OHS ACL .

DDD Bounded Context



1. **Open Host Service (OHS)**:
    - : OHS Bounded Context . Bounded Context .
    - : OHS RESTful API, SOAP , RPC . Bounded Context .
    - : .
2. **Anti-Corruption Layer (ACL)**:
    - : ACL Bounded Context Context ' ' . , .
    - : , ACL . , , .
    - : , .

API Context **Open Host Service** .

, OpenHost .

OHS

ACL .

OHS TPS

, .

OHS ACL .

?

.

.

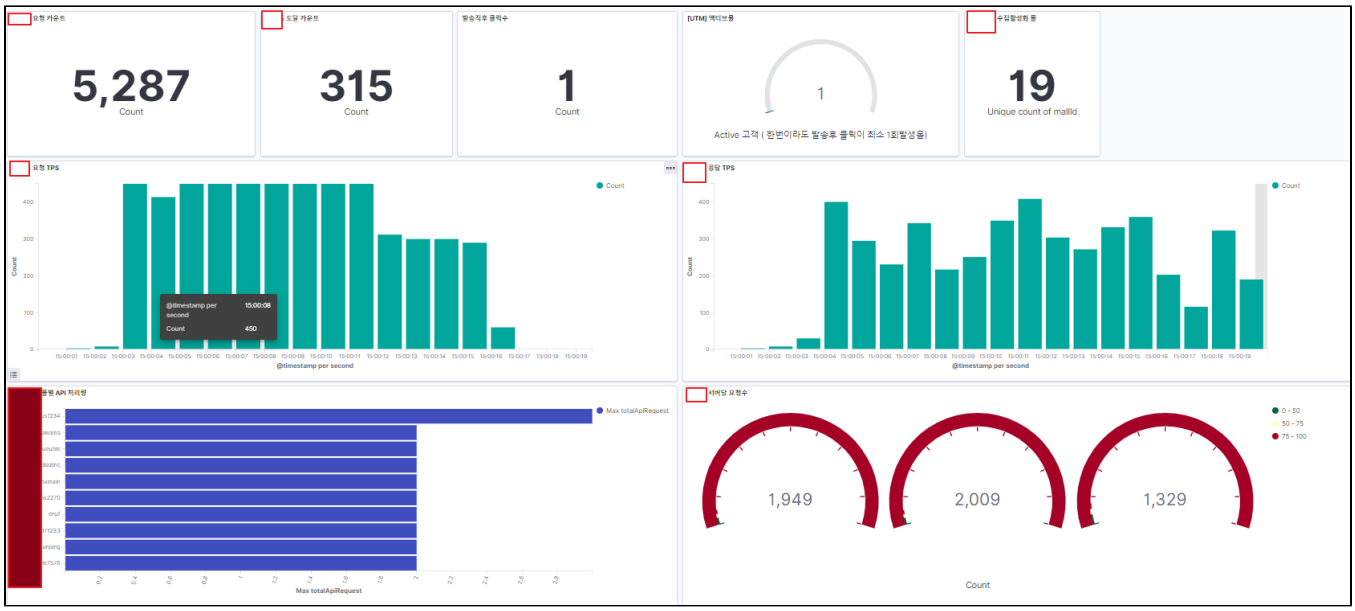
~ vs

, .

.

.

- CPU / / .
- ~ APM(Application Performance Monitor) .



API / TPS ~ .

~ APM .

TPS TPS , .

QA .

- , -
- () . -

BackPressure In reactive Streams

.

. .

.

-
-

, .

JAVA AKKA ~

Reactive Stream

AKKA .

TPS

TPS . OOP .

~ Log .

- TPS .
  - .
- TPS TPS .
- TPS 0 ~ 50% 0 .

TPS Just Tell .

```
                                for(int i=0;i<1200;i++){
                                    tpsActor.tell("some Event", getRef());
                                }
                                sleep(1500); // 1   delay 1.5

                                for(int i=0;i<600;i++){
                                    tpsActor.tell("some Event", getRef());
                                }
                                sleep(1500);

                                for(int i=0;i<300;i++){
                                    tpsActor.tell("some Event", getRef());
                                }
                                sleep(1500);

                                for(int i=0;i<500;i++){
                                    tpsActor.tell("some Event", getRef());
                                }
```

TPS , TPS .

```
[INFO ] [2023-12-10 11:47:13,575] [ClusterSystem-akka.actor.default-dispatcher-5] [TPS:1200.0]
[INFO ] [2023-12-10 11:47:14,574] [ClusterSystem-akka.actor.default-dispatcher-10] [TPS:600.0]
[INFO ] [2023-12-10 11:47:15,580] [ClusterSystem-akka.actor.default-dispatcher-10] [TPS:300.0]
[INFO ] [2023-12-10 11:47:16,562] [ClusterSystem-akka.actor.default-dispatcher-10] [TPS:150.0]
[INFO ] [2023-12-10 11:47:17,574] [ClusterSystem-akka.actor.default-dispatcher-9] [TPS:500.0]
[INFO ] [2023-12-10 11:47:18,572] [ClusterSystem-akka.actor.default-dispatcher-9] [TPS:250.0]
[INFO ] [2023-12-10 11:47:19,581] [ClusterSystem-akka.actor.default-dispatcher-6] [TPS:0.0]
[INFO ] [2023-12-10 11:47:20,559] [ClusterSystem-akka.actor.default-dispatcher-9] [TPS:0.0]
[INFO ] [2023-12-10 11:47:21,555] [ClusterSystem-akka.actor.default-dispatcher-9] [TPS:0.0]
```

~ . ? .

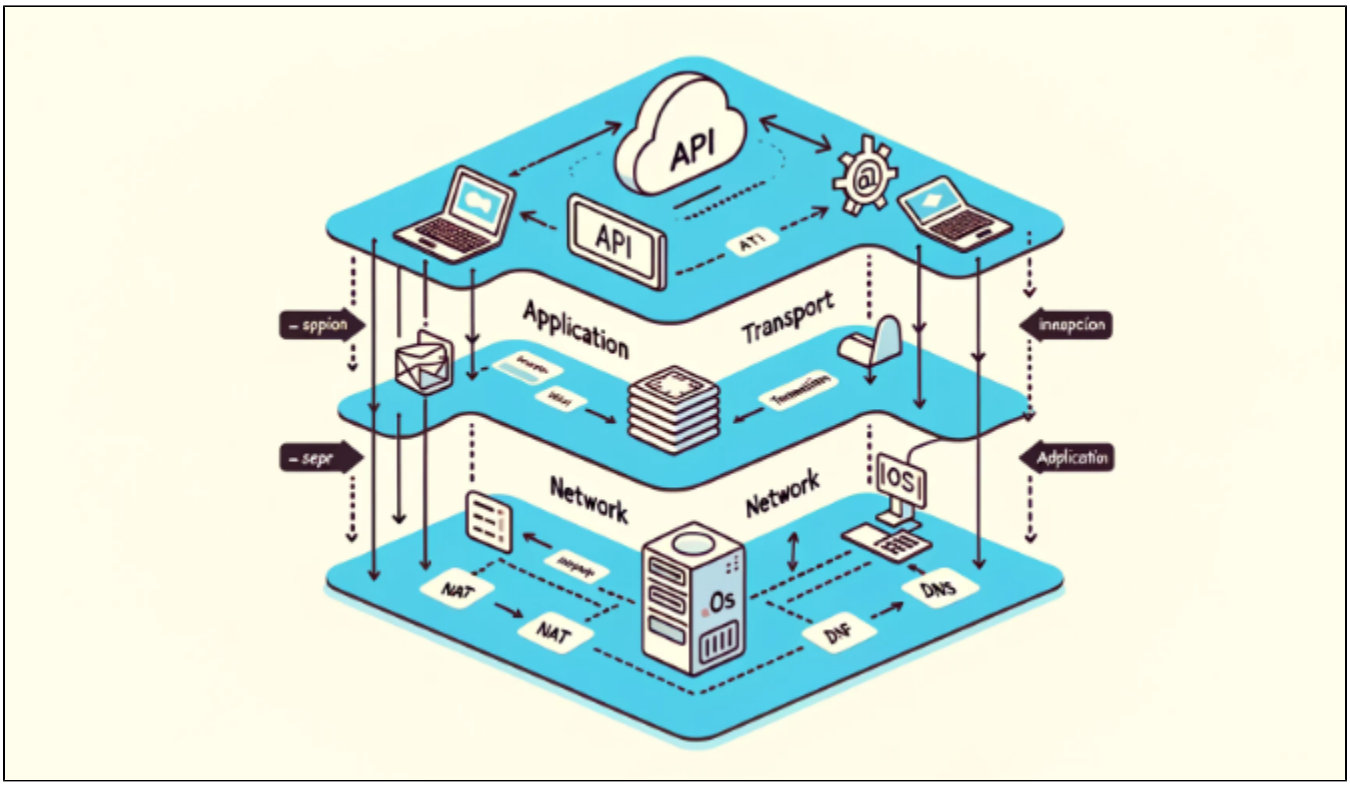
(SlowConsumerActor)

TPS , TPS 400 tps .

TPS 1 .

- TPS ~ TPS 0 ~ 0.5 50% 0. (TPS)
  - N Xms LogStash TPS .

```
                        private int tpsLimit = 400;

            if( result.tps > tpsLimit ){
                sleepValue =(long)result.tps;
                sleep(sleepValue);
                log.info("World Slow - Total:{} Sleep:{}", totalProcessCount, sleepValue);
            }
```



- API TPS .
  - .
    - 0 TPS TPS .
    - OSI ISP TPS .
      - .

```
int testCount = 50000;
int bufferSize = 100000;
int processCouuntPerSec = 400;

final ActorRef throttler =
        Source.actorRef(bufferSize, OverflowStrategy.dropNew())
                .throttle(processCouuntPerSec, FiniteDuration.create(1, TimeUnit.SECONDS),
                        processCouuntPerSec, ThrottleMode.shaping())
                .to(Sink.actorRef(slowConsumerActor, akka.NotUsed.getInstance()))
                .run(materializer);
```

TPS(400) . Block AkkaStream .

- testCount :
- bufferSize : . .
- processCouuntPerSec : TPS .

TPS 400

```
[INFO ] [2023-12-10 14:18:58,052] [ClusterSystem-akka.actor.default-dispatcher-7] [First Tick]
[INFO ] [2023-12-10 14:18:59,091] [ClusterSystem-akka.actor.default-dispatcher-7] [TPS:928.0]
[INFO ] [2023-12-10 14:19:00,021] [ClusterSystem-akka.actor.default-dispatcher-8] [World Slow - Total:928 Sleep:
928]
[INFO ] [2023-12-10 14:19:00,077] [ClusterSystem-akka.actor.default-dispatcher-8] [TPS:2.0]
[INFO ] [2023-12-10 14:19:00,486] [ClusterSystem-akka.actor.default-dispatcher-7] [World Slow - Total:929 Sleep:
464]
[INFO ] [2023-12-10 14:19:01,073] [ClusterSystem-akka.actor.default-dispatcher-7] [TPS:800.0]
[INFO ] [2023-12-10 14:19:01,879] [ClusterSystem-akka.actor.default-dispatcher-7] [World Slow - Total:1730
Sleep:800]
[INFO ] [2023-12-10 14:19:02,081] [ClusterSystem-akka.actor.default-dispatcher-4] [TPS:403.0]
[INFO ] [2023-12-10 14:19:02,501] [ClusterSystem-akka.actor.default-dispatcher-8] [World Slow - Total:2133
Sleep:403]
[INFO ] [2023-12-10 14:19:02,904] [ClusterSystem-akka.actor.default-dispatcher-8] [World Slow - Total:2134
Sleep:403]
[INFO ] [2023-12-10 14:19:03,059] [ClusterSystem-akka.actor.default-dispatcher-9] [TPS:391.0]
[INFO ] [2023-12-10 14:19:04,052] [ClusterSystem-akka.actor.default-dispatcher-5] [TPS:397.0]
[INFO ] [2023-12-10 14:19:05,078] [ClusterSystem-akka.actor.default-dispatcher-9] [TPS:404.0]
[INFO ] [2023-12-10 14:19:05,482] [ClusterSystem-akka.actor.default-dispatcher-9] [World Slow - Total:3325
Sleep:404]
[INFO ] [2023-12-10 14:19:05,888] [ClusterSystem-akka.actor.default-dispatcher-9] [World Slow - Total:3326
Sleep:404]
[INFO ] [2023-12-10 14:19:06,089] [ClusterSystem-akka.actor.default-dispatcher-7] [TPS:404.0]
```

400TPS ~ 400TPS .

- 400 / 800 / 2
- Hello World ~ Sleep: .
  - 15 ApiTimeOut .

TPS

2 .

2 .

1. : , . , , TPS .
2. : . , , , TPS .
3. : , . , , TPS .
4. : , , TPS .

TPS 2

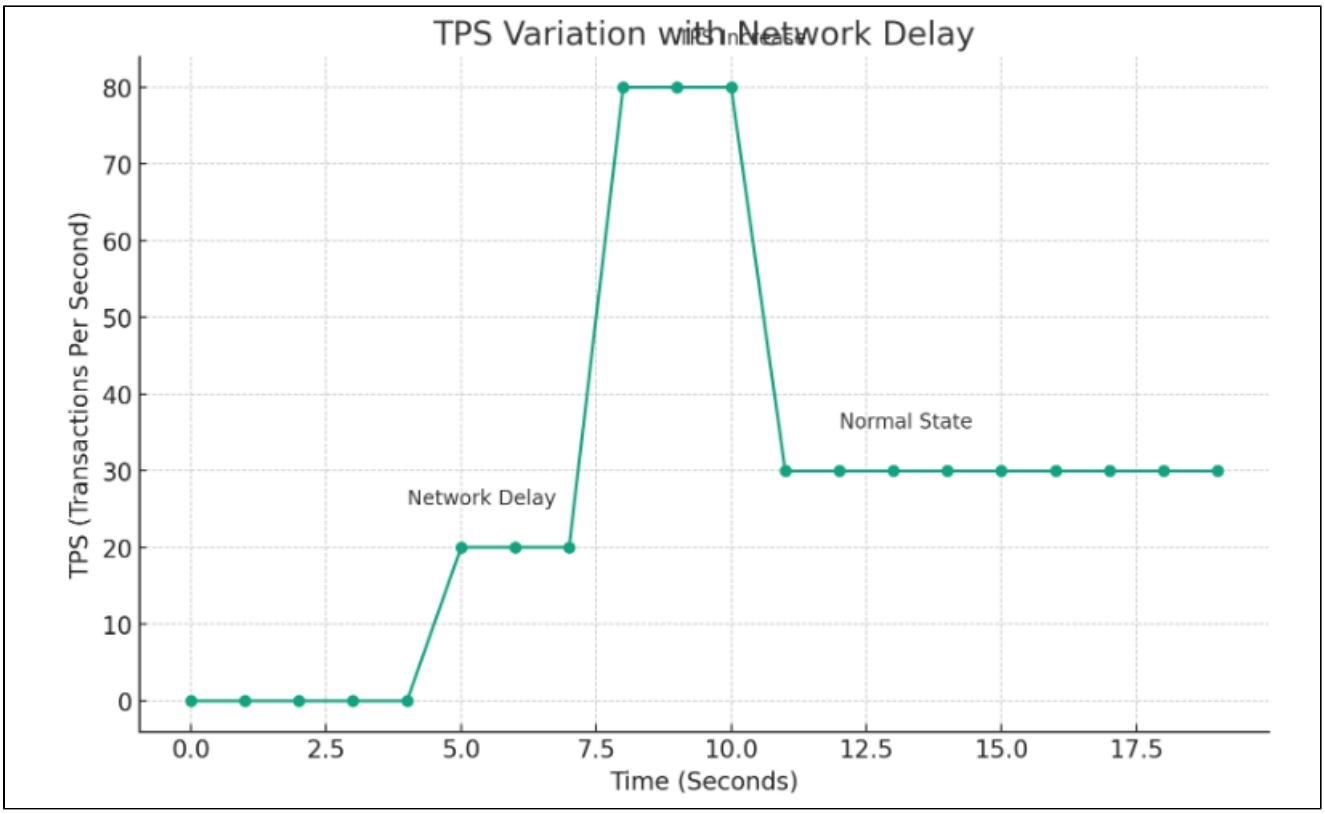
TPS . .

~ .

80 , TPS 80

.

~ TimeOut .



TPS . TPS :

- **(5-8)**: TPS .
- **TPS (8-11)**: TPS .
- **(11)**: TPS .

. .

GPT TPS ? .

GPT .

TPS .

TPS 200

```
[INFO ] [2023-12-10 14:38:55,256] [ClusterSystem-akka.actor.default-dispatcher-9] [TPS:198.0]
[INFO ] [2023-12-10 14:38:56,282] [ClusterSystem-akka.actor.default-dispatcher-6] [TPS:209.0]
[INFO ] [2023-12-10 14:38:57,261] [ClusterSystem-akka.actor.default-dispatcher-6] [TPS:195.0]
[INFO ] [2023-12-10 14:38:58,258] [ClusterSystem-akka.actor.default-dispatcher-8] [TPS:200.0]
[INFO ] [2023-12-10 14:38:59,284] [ClusterSystem-akka.actor.default-dispatcher-9] [TPS:202.0]
[INFO ] [2023-12-10 14:39:00,281] [ClusterSystem-akka.actor.default-dispatcher-6] [TPS:202.0]
[INFO ] [2023-12-10 14:39:01,260] [ClusterSystem-akka.actor.default-dispatcher-9] [TPS:193.0]
[INFO ] [2023-12-10 14:39:02,264] [ClusterSystem-akka.actor.default-dispatcher-8] [TPS:206.0]
[INFO ] [2023-12-10 14:39:03,274] [ClusterSystem-akka.actor.default-dispatcher-4] [TPS:202.0]
[INFO ] [2023-12-10 14:39:04,278] [ClusterSystem-akka.actor.default-dispatcher-6] [TPS:196.0]
[INFO ] [2023-12-10 14:39:05,275] [ClusterSystem-akka.actor.default-dispatcher-7] [TPS:199.0]
[INFO ] [2023-12-10 14:39:06,274] [ClusterSystem-akka.actor.default-dispatcher-7] [TPS:200.0]
[INFO ] [2023-12-10 14:39:07,281] [ClusterSystem-akka.actor.default-dispatcher-7] [TPS:204.0]
```

400TPS 200 .

. .

TPS

/ TPS .

1 ~ TPS

200~400 .

```
                int testCount = 50000;
                int bufferSize = 100000;
                int processCouuntPerSec = 400; //  TSP

                final ActorRef throttler =
                        Source.actorRef(bufferSize, OverflowStrategy.dropNew())
                                .throttle(processCouuntPerSec, FiniteDuration.create(1, TimeUnit.SECONDS),
                                        processCouuntPerSec, ThrottleMode.shaping())
                                .to(Sink.actorRef(slowConsumerActor, akka.NotUsed.getInstance()))
                                .run(materializer);
```

(, Kafka) TPS TPS

.

TPS . TPS ~ API / .

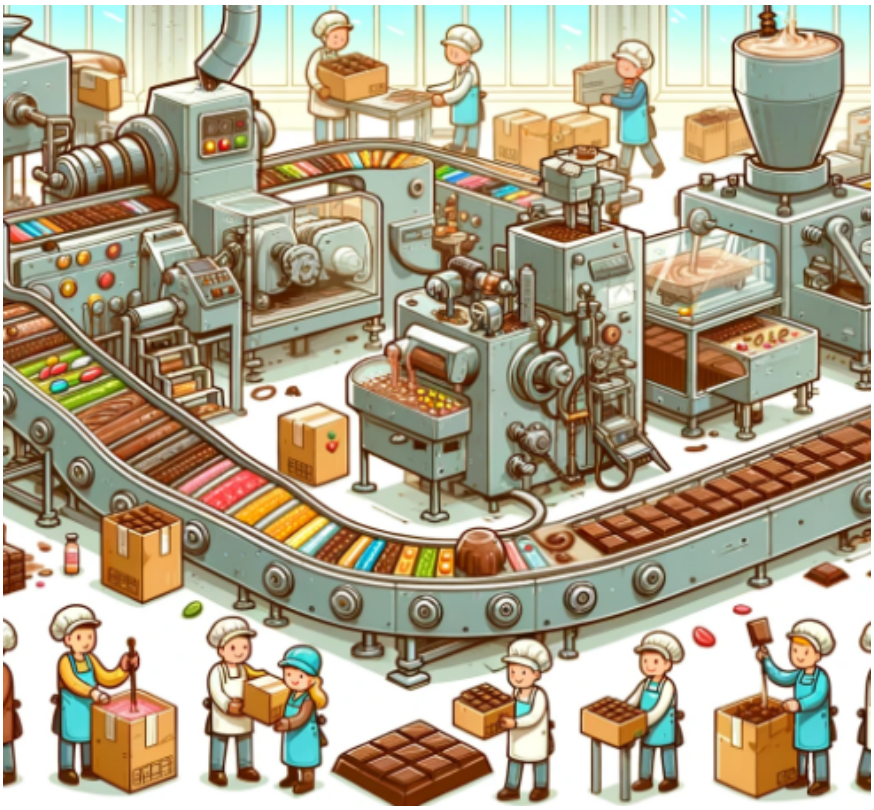
~ **BackPressure** .

.

~ ApiTimeOut .

.

BackPressure



- .
  -

BackPresure ~ () .

- API

```
    private static String callApi(Integer param) {
        // API
        try {
            Thread.sleep(100); // API
            tpsActor.tell("CompletedEvent", ActorRef.noSender());
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        return "Response for " + param;
    }
```

API , 100(0.1) .

- .
- TPS .

TPS 10. .

AkkaStream BackPressure

```
    @Test
    @DisplayName("BackPressureTest")
    public void BackPressureTest() {
        new TestKit(actorSystem) {
            {
                final Materializer materializer = ActorMaterializer.create(actorSystem);
                final TestKit probe = new TestKit(actorSystem);

                // Source
                Source<Integer, NotUsed> source = Source.range(1, 1000);

                // Flow  (API   )
                Flow<Integer, String, NotUsed> flow = Flow.fromFunction(BackPressureTest::callApi);

                // Buffer   OverflowStrategy.backpressure
                int bufferSize = 100;
                Flow<Integer, Integer, NotUsed> backpressureFlow = Flow.<Integer>create()
                        .buffer(bufferSize, OverflowStrategy.backpressure());

                // Sink
                Sink<String, CompletionStage<Done>> sink = Sink.foreach(System.out::println);

                // RunnableGraph
                source.via(backpressureFlow).via(flow).to(sink).run(materializer);

                within(
                        Duration.ofSeconds(15),
                        () -> {
                            // Will wait for the rest of the 10 seconds
                            expectNoMessage(Duration.ofSeconds(10));
                            return null;
                        });
            }
        };
    }
```

BackPressure TPS . .

.

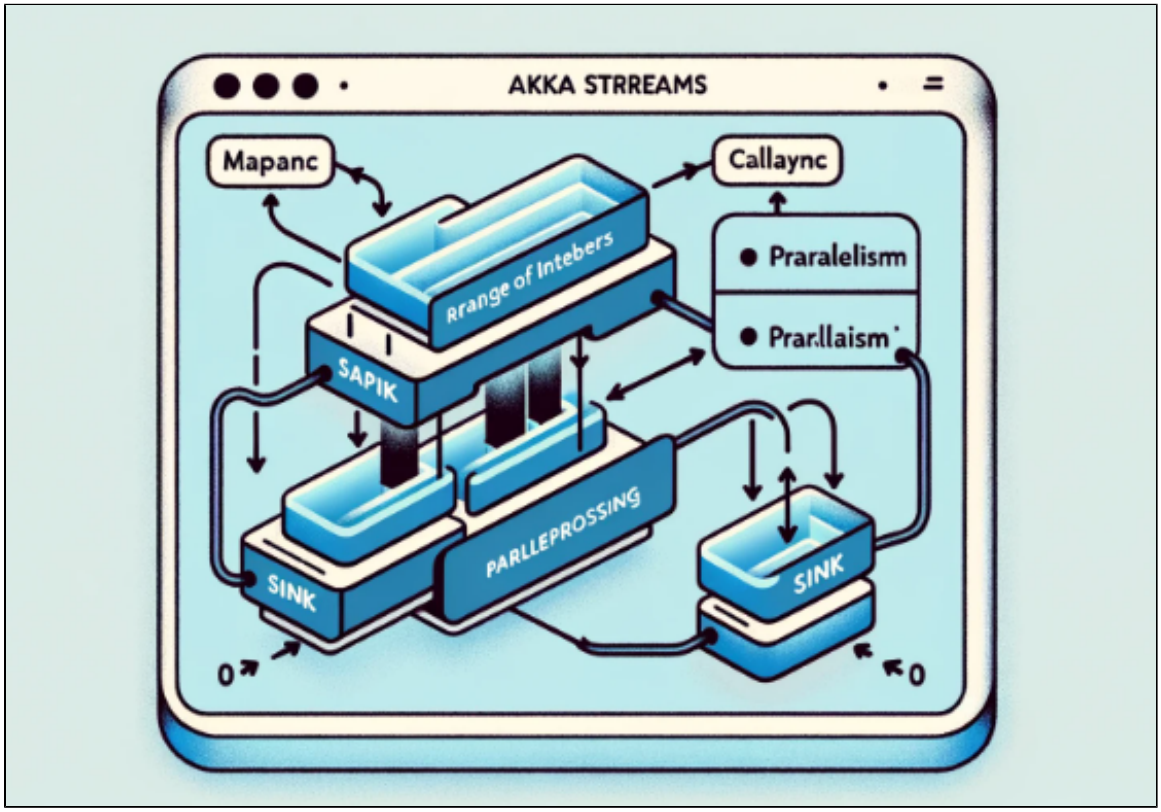
```
Response for 13
Response for 14
Response for 15
Response for 16
Response for 17
Response for 18
Response for 19
Response for 20
Response for 21
[INFO ] [2023-12-10 15:02:58,844] [ClusterSystem-akka.actor.default-dispatcher-6] [TPS:9.0]
Response for 22
Response for 23
Response for 24
Response for 25
Response for 26
Response for 27
Response for 28
Response for 29
Response for 30
Response for 31
[INFO ] [2023-12-10 15:02:59,845] [ClusterSystem-akka.actor.default-dispatcher-4] [TPS:10.0]
```

BackPresure

Flow API 1

RestClient .

Akka Stream Java 9 StreamAPI Linq(+TPL) .



- Source Flow Sink
  - , Flow Sink .
  - BackPresure Sink .

Flow Flow .

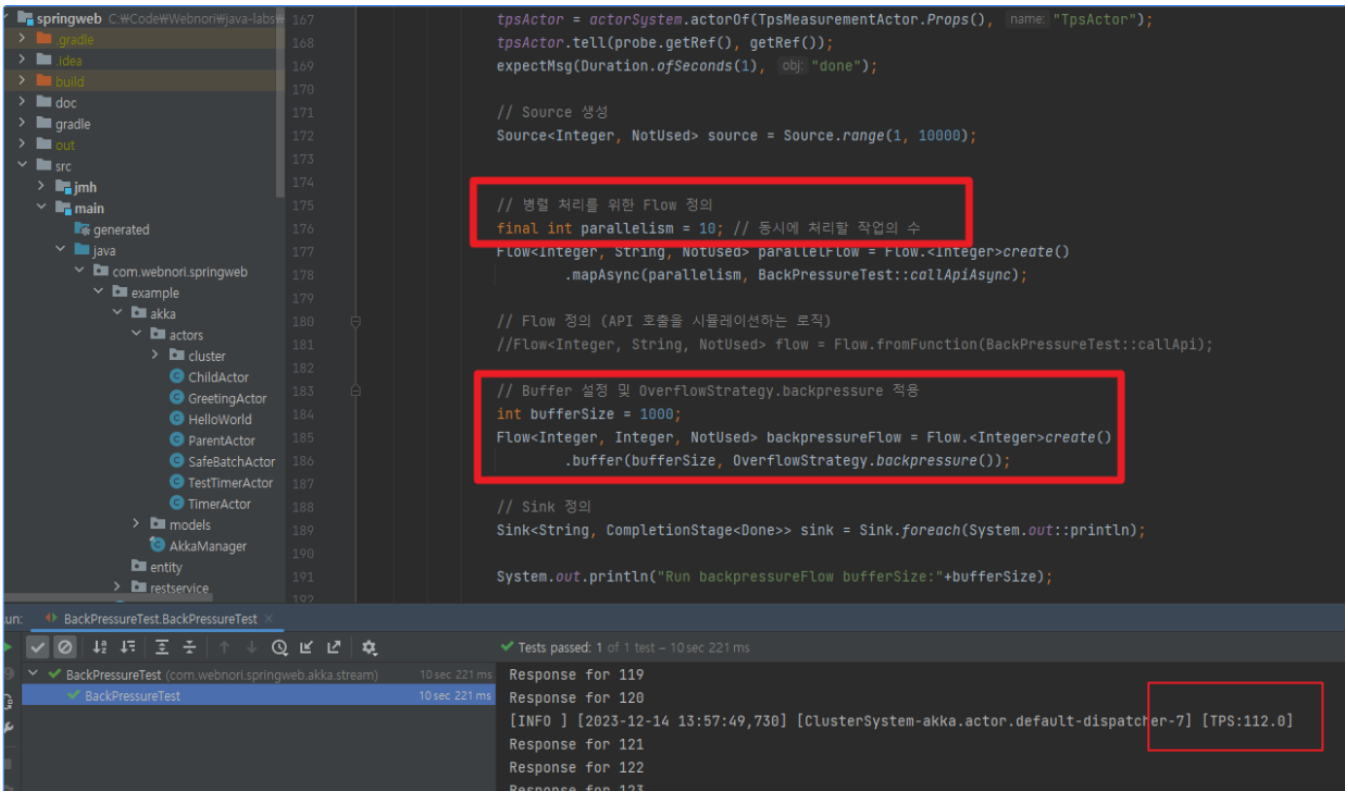
```
// Source
Source<Integer, NotUsed> source = Source.range(1, 10000);


//    Flow
final int parallelism = 10; //
Flow<Integer, String, NotUsed> parallelFlow = Flow.<Integer>create()
        .mapAsync(parallelism, BackPressureTest::callApiAsync);

// Flow  (API   )
//Flow<Integer, String, NotUsed> flow = Flow.fromFunction(BackPressureTest::callApi);

// Buffer   OverflowStrategy.backpressure
int bufferSize = 1000;
Flow<Integer, Integer, NotUsed> backpressureFlow = Flow.<Integer>create()
        .buffer(bufferSize, OverflowStrategy.backpressure());

// Sink
Sink<String, CompletionStage<Done>> sink = Sink.foreach(System.out::println);

System.out.println("Run backpressureFlow bufferSize:"+bufferSize);

// RunnableGraph
source.via(backpressureFlow).via(parallelFlow).to(sink).run(materializer);


private static CompletionStage<String> callApiAsync(Integer param) {
// CompletableFuture
return CompletableFuture.supplyAsync(() -> {
        try {
                Thread.sleep(100); // API
                tpsActor.tell("CompletedEvent", ActorRef.noSender());
        } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
        }
        return "Response for " + param;
});
```

Buffer (parallelism) ~ BackPresure .

TPS . .

BackPresure ~ .

Backpresure



- Backpresure .
- Backpresure TPS ~ () ~ .
  - ~ .

.

GitHub .

- https://github.com/psmon/java-labs/tree/master/springweb/src/test/java/com/webnori/springweb/akka/stream

GPT .

Java Akka . :

1. : `BackPressureTest` . Akka `TestKit` . `TestKit` Akka .
2. :
   - `Materializer` `TestKit` . .
   - `TpsMeasurementActor` , . .
3. **Akka Streams** :
   - : `Source.range(1, 10000)` 1 10,000 .
   - **Flow** : API `mapAsync` `buffer` `Flow` . `mapAsync` , `buffer` (Back Pressure) .
   - **(Sink)** : `Sink` .
4. : `bufferSize` , `OverflowStrategy.backpressure()` . .
5. **RunnableGraph** : , , .
6. **API** : `callApiAsync` `CompletableFuture` API . 100ms , `tpsActor` .
7. : `within 15` , .

Akka Streams .

TPS ~ Backpresure .

BackPresure , .

AKKA .

JAVA7 , AKKA AKKA . Backpresure .

1. **Java (Reactive Streams)**
    - Java Reactive Streams API BackPressure . , `Flow` `Publisher`, `Subscriber` .
2. **JavaScript (RxJS)**
    - JavaScript RxJS BackPressure . RxJS `Observable` `BackPressure` .
3. **Python (RxPy)**
    - Python RxPy(ReactiveX for Python) BackPressure . .
4. **C# (System.Reactive)**
    - C# System.Reactive , BackPressure . `IObservable<T>` `IObserver<T>` .
5. **Scala (Akka Streams)**
    - Scala Akka Streams BackPressure . Akka Streams Akka .

BackPressure . API .

DB/API TPS AKKA

- TPS
- [Labs] AKKA