Raport: Akka

Łukasz Krówka 296602

Zad 1.

Resume vs. Restart, OUTPUT:

```
Started. Commands: 'hi', 'm [nb1] [nb2]', 'q'
d 1 3
result: 0.3333333333333333 (operation count: 1)
d 8 11
result: 0.7272727272727273 (operation count: 2)
d 1 0
result: Infinity (operation count: 3)
d 1 2
result: 0.5 (operation count: 4)
d terazBedzieRestart
[ERROR] [05/16/2020 13:04:08.305] [local_system-akka.actor.default-dispatcher-4]
[akka://local_system/user/math/divideWorker] For input string: "terazBedzieRestart"
java.lang.NumberFormatException: For input string: "terazBedzieRestart"
	at
java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
	at java.base/java.lang.Integer.parseInt(Integer.java:652)
	at java.base/java.lang.Integer.parseInt(Integer.java:770)
	at z1.Z1_DivideWorker.divide(Z1_DivideWorker.java:24)
	at z1.Z1_DivideWorker.lambda$createReceive$0(Z1_DivideWorker.java:15)
	at akka.japi.pf.UnitCaseStatement.apply(CaseStatements.scala:24)
	at akka.japi.pf.UnitCaseStatement.apply(CaseStatements.scala:20)
	at scala.PartialFunction.applyOrElse(PartialFunction.scala:187)
	at scala.PartialFunction.applyOrElse$(PartialFunction.scala:186)
	at akka.japi.pf.UnitCaseStatement.applyOrElse(CaseStatements.scala:20)
	at scala.PartialFunction$OrElse.applyOrElse(PartialFunction.scala:241)
	at akka.actor.Actor.aroundReceive(Actor.scala:535)
	at akka.actor.Actor.aroundReceive$(Actor.scala:533)
	at akka.actor.AbstractActor.aroundReceive(AbstractActor.scala:220)
	at akka.actor.ActorCell.receiveMessage(ActorCell.scala:575)
	at akka.actor.ActorCell.invoke(ActorCell.scala:545)
	at akka.dispatch.Mailbox.processMailbox(Mailbox.scala:270)
	at akka.dispatch.Mailbox.run(Mailbox.scala:231)
	at akka.dispatch.Mailbox.exec(Mailbox.scala:243)
	at java.base/java.util.concurrent.ForkJoinTask.doExec(ForkJoinTask.java:290)
	at
java.base/java.util.concurrent.ForkJoinPool$WorkQueue.topLevelExec(ForkJoinPool.java:1020)
	at java.base/java.util.concurrent.ForkJoinPool.scan(ForkJoinPool.java:1656)
	at java.base/java.util.concurrent.ForkJoinPool.runWorker(ForkJoinPool.java:1594)
	at java.base/java.util.concurrent.ForkJoinWorkerThread.run(ForkJoinWorkerThread.java:177)

d 6 9
result: 0.6666666666666666 (operation count: 1)
```

Na podstawie output'u można widać, że po wznowieniu obiekty zachowują swój stan (tu: licznik operacji), z kolei po restarcie już się tak nie dzieje.

OneForOneStrategy OUTPUT:

```
Started. Commands: 'hi', 'm [nb1] [nb2]', 'q'
m 1 2
result: 2 (operation count: 1)
d 1 2
result: 0.5 (operation count: 1)
d 1 0
result: Infinity (operation count: 2)
d 1 1
result: 1.0 (operation count: 3)
m 1 2
result: 2 (operation count: 2)
d OneForOne
[ERROR] [05/16/2020 13:14:17.195] [local_system-akka.actor.default-dispatcher-4]
[akka://local_system/user/math/divideWorker] For input string: "OneForOne"
java.lang.NumberFormatException: For input string: "OneForOne"
        at
java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
        at java.base/java.lang.Integer.parseInt(Integer.java:652)
        at java.base/java.lang.Integer.parseInt(Integer.java:770)
        at z1.Z1_DivideWorker.divide(Z1_DivideWorker.java:24)
        at z1.Z1_DivideWorker.lambda$createReceive$0(Z1_DivideWorker.java:15)
        at akka.japi.pf.UnitCaseStatement.apply(CaseStatements.scala:24)
        at akka.japi.pf.UnitCaseStatement.apply(CaseStatements.scala:20)
        at scala.PartialFunction.applyOrElse(PartialFunction.scala:187)
        at scala.PartialFunction.applyOrElse$(PartialFunction.scala:186)
        at akka.japi.pf.UnitCaseStatement.applyOrElse(CaseStatements.scala:20)
        at scala.PartialFunction$OrElse.applyOrElse(PartialFunction.scala:241)
        at akka.actor.Actor.aroundReceive(Actor.scala:535)
        at akka.actor.Actor.aroundReceive$(Actor.scala:533)
        at akka.actor.AbstractActor.aroundReceive(AbstractActor.scala:220)
        at akka.actor.ActorCell.receiveMessage(ActorCell.scala:575)
        at akka.actor.ActorCell.invoke(ActorCell.scala:545)
        at akka.dispatch.Mailbox.processMailbox(Mailbox.scala:270)
        at akka.dispatch.Mailbox.run(Mailbox.scala:231)
        at akka.dispatch.Mailbox.exec(Mailbox.scala:243)
        at java.base/java.util.concurrent.ForkJoinTask.doExec(ForkJoinTask.java:290)
        at
java.base/java.util.concurrent.ForkJoinPool$WorkQueue.topLevelExec(ForkJoinPool.java:1020)
        at java.base/java.util.concurrent.ForkJoinPool.scan(ForkJoinPool.java:1656)
        at java.base/java.util.concurrent.ForkJoinPool.runWorker(ForkJoinPool.java:1594)
        at java.base/java.util.concurrent.ForkJoinWorkerThread.run(ForkJoinWorkerThread.java:177)

d 1 2
result: 0.5 (operation count: 1)
m 1 2
result: 2 (operation count: 3)
```

W wyniku błędu zrestartowany został jedynie jeden worker - ten, który zgłosił błąd.

Strategia OneForOne podczas błędu restartuje jedynie worker’a, u którego ten błąd wystąpił.

AllForOneStrategy OUTPUT:

```
Started. Commands: 'hi', 'm [nb1] [nb2]', 'q'
m 1 2
result: 2 (operation count: 1)
d 1 2
result: 0.5 (operation count: 1)
d 1 0
result: Infinity (operation count: 2)
d 1 1
result: 1.0 (operation count: 3)
m 1 2
result: 2 (operation count: 2)
d AllForOne
[ERROR] [05/12/2020 13:10:43.836] [local_system-akka.actor.default-dispatcher-6]
[akka://local_system/user/math/divideWorker] For input string: "AllForOne"
java.lang.NumberFormatException: For input string: "AllForOne"
	at
java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
	at java.base/java.lang.Integer.parseInt(Integer.java:652)
	at java.base/java.lang.Integer.parseInt(Integer.java:770)
	at z1.Z1_DivideWorker.divide(Z1_DivideWorker.java:24)
	at z1.Z1_DivideWorker.lambda$createReceive$0(Z1_DivideWorker.java:15)
	at akka.japi.pf.UnitCaseStatement.apply(CaseStatements.scala:24)
	at akka.japi.pf.UnitCaseStatement.apply(CaseStatements.scala:20)
	at scala.PartialFunction.applyOrElse(PartialFunction.scala:187)
	at scala.PartialFunction.applyOrElse$(PartialFunction.scala:186)
	at akka.japi.pf.UnitCaseStatement.applyOrElse(CaseStatements.scala:20)
	at scala.PartialFunction$OrElse.applyOrElse(PartialFunction.scala:241)
	at akka.actor.Actor.aroundReceive(Actor.scala:535)
	at akka.actor.Actor.aroundReceive$(Actor.scala:533)
	at akka.actor.AbstractActor.aroundReceive(AbstractActor.scala:220)
	at akka.actor.ActorCell.receiveMessage(ActorCell.scala:575)
	at akka.actor.ActorCell.invoke(ActorCell.scala:545)
	at akka.dispatch.Mailbox.processMailbox(Mailbox.scala:270)
	at akka.dispatch.Mailbox.run(Mailbox.scala:231)
	at akka.dispatch.Mailbox.exec(Mailbox.scala:243)
	at java.base/java.util.concurrent.ForkJoinTask.doExec(ForkJoinTask.java:290)
	at
java.base/java.util.concurrent.ForkJoinPool$WorkQueue.topLevelExec(ForkJoinPool.java:1020)
	at java.base/java.util.concurrent.ForkJoinPool.scan(ForkJoinPool.java:1656)
	at java.base/java.util.concurrent.ForkJoinPool.runWorker(ForkJoinPool.java:1594)
	at java.base/java.util.concurrent.ForkJoinWorkerThread.run(ForkJoinWorkerThread.java:177)

d 1 2
result: 0.5 (operation count: 1)
m 1 2
result: 2 (operation count: 1)
```

W wyniku błędu zrestartowane zostały obydwa worker'y, pomimo tego, że błąd zgłosił tylko jeden z nich. Strategia AllForOne podczas błędu restartuje wszystkich podwładnych.

Zad 2.

Z2_AppLocal OUTPUT:

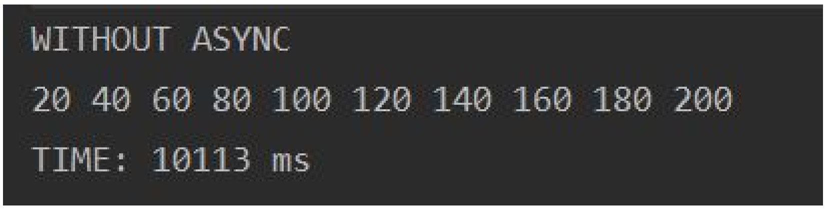
```
[WARN] [05/12/2020 14:22:06.288] [main] [akka.remote.RemoteActorRefProvider] Using the 'remote'
ActorRefProvider directly, which is a low-level layer. For most use cases, the 'cluster' abstraction on
top of remoting is more suitable instead.
[WARN] [05/12/2020 14:22:06.288] [main] [akka.remote.RemoteActorRefProvider] Akka Cluster not in
use - Using Akka Cluster is recommended if you need remote watch and deploy.
[INFO] [05/12/2020 14:22:07.151] [main] [ArteryTcpTransport(akka://local_system)] Remoting started
with transport [Artery tcp]; listening on address [akka://local_system@127.0.0.1:2551] with UID
[5583959716408565384]
nie pytają cię o imię walczą z ostrym cieniem mgły
Local actor got message: nie pytają cię o imię walczą z ostrym cieniem mgły
Local actor got response: NIE PYTAJĄ CIĘ O IMIĘ WALCZĄ Z OSTRYM CIENIEM MGŁY
```

Z2_AppRemote OUTPUT:

```
[WARN] [05/12/2020 14:21:59.255] [main] [akka.remote.RemoteActorRefProvider] Using the 'remote'
ActorRefProvider directly, which is a low-level layer. For most use cases, the 'cluster' abstraction on
top of remoting is more suitable instead.
[WARN] [05/12/2020 14:21:59.255] [main] [akka.remote.RemoteActorRefProvider] Akka Cluster not in
use - Using Akka Cluster is recommended if you need remote watch and deploy.
[INFO] [05/12/2020 14:22:00.027] [main] [ArteryTcpTransport(akka://remote_system)] Remoting
started with transport [Artery tcp]; listening on address [akka://remote_system@127.0.0.1:2552] with
UID [-3935256013183212123]
Remote actor got message: nie pytają cię o imię walczą z ostrym cieniem mgły
```

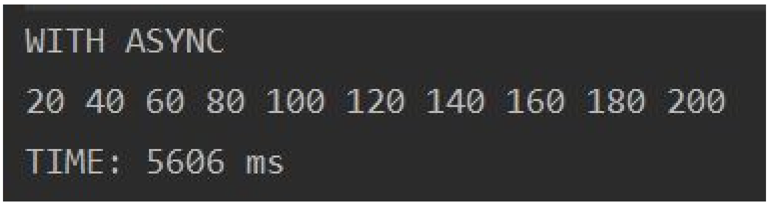
Zad 3.

a) bez async:



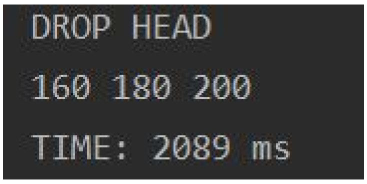
```
WITHOUT ASYNC
20 40 60 80 100 120 140 160 180 200
TIME: 10113 ms
```

b) z async:



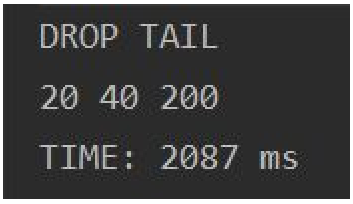
```
WITH ASYNC
20 40 60 80 100 120 140 160 180 200
TIME: 5606 ms
```

c) dropHead:



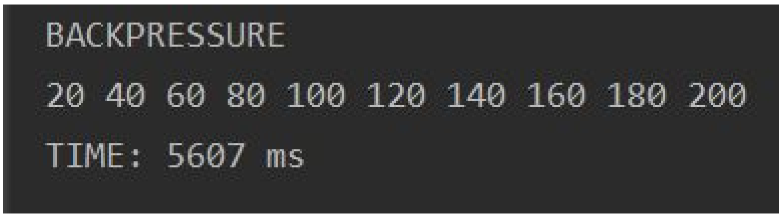
```
DROP HEAD
160 180 200
TIME: 2089 ms
```

d) dropTail:



```
DROP TAIL
20 40 200
TIME: 2087 ms
```

e) backpressure:



```
BACKPRESSURE
20 40 60 80 100 120 140 160 180 200
TIME: 5607 ms
```