

# Laboratorium 4 - Akka

Patryk Wojtyczek, 305454

## Zadanie 1

Zadanie polegało na obserwacji jak działają różne strategie supervisorów.

Na początku warto zaznaczyć, że strategie mają znacznie tylko gdy w przypadku gdy aktor popełnia błędy - rzuca wyjątki. W przypadku tzw. happy case wszystkie strategie mają taki sam wpływ na program - czyli żaden.

Do testowania wykonuję następujące komendy:

```
// send messages
system.tell(new MathActor.MathCommandDivide( firstNumber: 15, secondNumber: 5, replyTo: null));
system.tell(new MathActor.MathCommandDivide( firstNumber: 15, secondNumber: 0, replyTo: null));
Thread.sleep( millis: 2000);

System.out.println("Z1 main: sending second package of messages");
system.tell(new MathActor.MathCommandDivide( firstNumber: 15, secondNumber: 3, replyTo: null));

System.out.println("Z1 main: messages send");
```

## Porównanie strategii

- stop - Aktor wykonuje pierwsze dzielenie, zwiększa licznik, zaczyna wykonywać drugie, zwiększa licznik, rzuca exception, które dostaje jako wiadomość supervisor i decyduje się ubić aktora strategią Stop. Przy następnej komendzie widać kolejne exception, które mówi, że wiadomość nie została wysłana bo aktor leży.

Tak więc, strategia stop w przypadku błędu zatrzymuje na stałe aktora, który rzucił wyjątek.

```
Z1 main: started
Z1 main: actor system ready
actorMath: received command: add
actorMath: add result = 8
actorMath: received command: divide
actorMath: sending to actorDivide
actorMath: received command: divide
actorMath: sending to actorDivide
ActorDivide: received command: divide
actorDivide: divide result = 3
actorDivide: sending response
actorDivide: 1 operations performed
ActorDivide: received command: divide
actorMath: received result: 3
[actorMath-akka.actor.default-dispatcher-3] ERROR akka.actor.typed.Behavior$ - Supervisor StopSupervisor saw failure: / by zero
java.lang.ArithmeticException: Create breakpoint: / by zero
    at Z1.MathActorDivide.onMathCommandDivide(MathActorDivide.java:36)
    at akka.actor.typed.javadsl.BuiltReceive.receive(ReceiveBuilder.scala:213)
    at akka.actor.typed.javadsl.BuiltReceive.receiveMessage(ReceiveBuilder.scala:284)
    at akka.actor.typed.javadsl.Receive.receive(Receive.scala:53)
    at akka.actor.typed.javadsl.AbstractBehavior.receive(AbstractBehavior.scala:64)
    at akka.actor.typed.Behavior$.interpret(Behavior.scala:224)
    at akka.actor.typed.Behavior$.interpretMessage(Behavior.scala:230)
    at akka.actor.typed.internal.InterceptorImpl$$anon$2.apply(InterceptorImpl.scala:57)
    at akka.actor.typed.internal.SimpleSupervisor.aroundReceive(Supervision.scala:124)
    at akka.actor.typed.internal.InterceptorImpl.receive(InterceptorImpl.scala:85)
    at akka.actor.typed.Behavior$.interpret(Behavior.scala:224)
    at akka.actor.typed.Behavior$.interpretMessage(Behavior.scala:230)
    at akka.actor.typed.internal.adapter.ActorAdapter.handleMessage(ActorAdapter.scala:131)
    at akka.actor.typed.internal.adapter.ActorAdapter.aroundReceive(ActorAdapter.scala:187)
    at akka.actor.ActorCell.receiveMessage(ActorCell.scala:579)
    at akka.actor.ActorCell.invoke(ActorCell.scala:547)
    at akka.dispatch.Mailbox.processMailbox(Mailbox.scala:278)
    at akka.dispatch.Mailbox.run(Mailbox.scala:231)
    at akka.dispatch.Mailbox.exec(Mailbox.scala:243) <5 internal lines>
Z1 main: sending second package of messages
Z1 main: messages send
actorMath: received command: divide
actorMath: sending to actorDivide
[actorMath-akka.actor.default-dispatcher-6] INFO akka.actor.LocalActorRef - Message [Z1.MathActor$MathCommandDivide] to Actor[akka://actorMath/user/actorDivide#1511557900] was not delivered. [1] dead letters encountered. If this is not an expected behavior then Actor[akka://actorMath/user/actorDivide#1511557900] may have terminated unexpectedly. This logging can be turned off or adjusted with configuration settings 'akka.log-dead-letters' and 'akka.log-dead-letters-during-shutdown'.
```

- resume - Aktor wykonuje pierwsze dzielenie, zwiększa licznik, zaczyna wykonywać drugie, zwiększa licznik, rzuca exception, które dostaje jako wiadomość supervisor i ten decyduje się wznowić aktora strategią 'Resume'. Strategia ta nie restartuje aktora ale zachowuje jego aktualny stan co widać bo licznik po wykonaniu następnego dzielnia wynosi 3 (1 poprawne, 1 przed exception i aktualne). Strategia ta nie rusza też dzieci dla których dany aktor jest supervisorom.

```

Z1 main: started
[actorMath-akka.actor.default-dispatcher-5] INFO akka.event.slf4j.Slf4jLogger - Slf4jLogger started
Z1 main: actor system ready
actorMath: received command: divide
actorMath: sending to actorDivide
actorMath: received command: divide
actorMath: sending to actorDivide
ActorDivide: received command: divide
actorDivide: divide result = 3
actorDivide: sending response
actorDivide: 1 operations performed
ActorDivide: received command: divide
actorMath: received result: 3
[actorMath-akka.actor.default-dispatcher-3] ERROR akka.actor.typed.Behavior$ - Supervisor ResumeSupervisor saw failure: / by zero
java.lang.ArithmeticException Create breakpoint : / by zero
    at Z1.MathActorDivide.onMathCommandDivide(MathActorDivide.java:36)
    at akka.actor.typed.javadsl.BuiltReceive.receive(ReceiveBuilder.scala:213)
    at akka.actor.typed.javadsl.BuiltReceive.receiveMessage(ReceiveBuilder.scala:204)
    at akka.actor.typed.javadsl.Receive.receive(Receive.scala:53)
    at akka.actor.typed.javadsl.AbstractBehavior.receive(AbstractBehavior.scala:64)
    at akka.actor.typed.Behavior$.interpret(Behavior.scala:274)
    at akka.actor.typed.Behavior$.interpretMessage(Behavior.scala:230)
    at akka.actor.typed.internal.InterceptorImpl$$anon$2.apply(InterceptorImpl.scala:57)
    at akka.actor.typed.internal.SimpleSupervisor.aroundReceive(Supervision.scala:124)
    at akka.actor.typed.internal.InterceptorImpl.receive(InterceptorImpl.scala:85)
    at akka.actor.typed.Behavior$.interpret(Behavior.scala:274)
    at akka.actor.typed.Behavior$.interpretMessage(Behavior.scala:230)
    at akka.actor.typed.internal.adapter.ActorAdapter.handleMessage(ActorAdapter.scala:131)
    at akka.actor.typed.internal.adapter.ActorAdapter.aroundReceive(ActorAdapter.scala:107)
    at akka.actor.ActorCell.receiveMessage(ActorCell.scala:579)
    at akka.actor.ActorCell.invoke(ActorCell.scala:547)
    at akka.dispatch.Mailbox.processMailbox(Mailbox.scala:270)
    at akka.dispatch.Mailbox.run(Mailbox.scala:231)
    at akka.dispatch.Mailbox.exec(Mailbox.scala:243) <5 internal lines>
Z1 main: sending second package of messages
Z1 main: messages send
actorMath: received command: divide
actorMath: sending to actorDivide
ActorDivide: received command: divide
actorDivide: divide result = 5
actorDivide: sending response
actorDivide: 3 operations performed
actorMath: received result: 5

```

- restart - Analogicznie do poprzedniej jednak teraz, stan (czyli licznik) nie został zachowany bo aktor został zrestartowany. Strategia ta zrestartowałaby też wszystkie dzieci dla których ten aktor byłby supervisorem.

```

Z1 main: started
[actorMath-akka.actor.default-dispatcher-5] INFO akka.event.slf4j.Slf4jLogger - Slf4jLogger started
Z1 main: actor system ready
actorMath: received command: divide
actorMath: sending to actorDivide
actorMath: received command: divide
actorMath: sending to actorDivide
ActorDivide: received command: divide
actorDivide: divide result = 3
actorDivide: sending response
actorDivide: 1 operations performed
ActorDivide: received command: divide
actorMath: received result: 3
[actorMath-akka.actor.default-dispatcher-3] ERROR akka.actor.typed.Behavior$ - Supervisor RestartSupervisor saw failure: / by zero
java.lang.ArithmeticException Create breakpoint : / by zero
    at Z1.MathActorDivide.onMathCommandDivide(MathActorDivide.java:36)
    at akka.actor.typed.javadsl.BuiltReceive.receive(ReceiveBuilder.scala:213)
    at akka.actor.typed.javadsl.BuiltReceive.receiveMessage(ReceiveBuilder.scala:204)
    at akka.actor.typed.javadsl.Receive.receive(Receive.scala:53)
    at akka.actor.typed.javadsl.AbstractBehavior.receive(AbstractBehavior.scala:64)
    at akka.actor.typed.Behavior$.interpret(Behavior.scala:274)
    at akka.actor.typed.Behavior$.interpretMessage(Behavior.scala:230)
    at akka.actor.typed.internal.InterceptorImpl$$anon$2.apply(InterceptorImpl.scala:57)
    at akka.actor.typed.internal.RestartSupervisor.aroundReceive(Supervision.scala:268)
    at akka.actor.typed.internal.InterceptorImpl.receive(InterceptorImpl.scala:85)
    at akka.actor.typed.Behavior$.interpret(Behavior.scala:274)
    at akka.actor.typed.Behavior$.interpretMessage(Behavior.scala:230)
    at akka.actor.typed.internal.adapter.ActorAdapter.handleMessage(ActorAdapter.scala:131)
    at akka.actor.typed.internal.adapter.ActorAdapter.aroundReceive(ActorAdapter.scala:107)
    at akka.actor.ActorCell.receiveMessage(ActorCell.scala:579)
    at akka.actor.ActorCell.invoke(ActorCell.scala:547)
    at akka.dispatch.Mailbox.processMailbox(Mailbox.scala:270)
    at akka.dispatch.Mailbox.run(Mailbox.scala:231)
    at akka.dispatch.Mailbox.exec(Mailbox.scala:243) <5 internal lines>
Z1 main: sending second package of messages
Z1 main: messages send
actorMath: received command: divide
actorMath: sending to actorDivide
ActorDivide: received command: divide
actorDivide: divide result = 5
actorDivide: sending response
actorDivide: 1 operations performed
actorMath: received result: 5

```

## Zadanie 2

To prosty przykład, że akka może działać na wielu jvmach jednocześnie - niekoniecznie na jednym nodzie.

- podpunkt a, output wypisany przed Z2\_Main

```

/Users/petroniuss/Library/Java/JavaVirtualMachines/adopt-openjdk-14.0.2/Contents/Home/bin/java ...
[z2main-akka.actor.default-dispatcher-5] INFO akka.event.slf4j.Slf4jLogger - Slf4jLogger started
akka://z2main/user/upper1 registered
akka://z2main/user/upper2 registered
creating receive for text service
On ListingRequest
request: hello
sending to worker: Actor[akka://z2main/user/upper1#-660076727]
sending to worker: Actor[akka://z2main/user/upper2#-559718653]
HELLO
HELLO

```

- podpunkt b
  - node a - tworzy workerów, a następnie obsługuje wiadomość wysłaną z nodea b.

```

/Users/petroniuss/Library/Java/JavaVirtualMachines/adopt-openjdk-14.0.2/Contents/Home/bin/java ...
Node A: config: Config(SimpleConfigObject({"akka":{"actor":{"provider":"cluster"},"cluster":{"downing-provider-class":"akka.cluster.sbr.SplitBrainResolverProvider"},"seed-nodes":["akka://ClusterSystem@127.0.0.1:2551","akka://ClusterSystem@127.0.0.1:2552"],"remote":{"artery":{"canonical":{"hostname":"127.0.0.1","port":2551}}}}}))
[ClusterSystem-akka.actor.default-dispatcher-3] INFO akka.remote.artery.tcp.ArteryTcpTransport - Remoting started with transport [Artery tcp]; listening on address [akka://ClusterSystem@127.0.0.1:2551] with UID [-2899452638012261337]
[ClusterSystem-akka.actor.default-dispatcher-3] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2551] - Starting up, Akka version [2.6.14] ...
[ClusterSystem-akka.actor.default-dispatcher-3] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2551] - Registered cluster JMX MBean [akka:type=Cluster]
[ClusterSystem-akka.actor.default-dispatcher-3] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2551] - Started up successfully
[ClusterSystem-akka.actor.default-dispatcher-3] INFO akka.cluster.sbr.SplitBrainResolver - SBR started. Config: strategy [KeepMajority], stable-after [20 seconds], down-all-when-unstable [15 seconds], selfUniqueAddress [akka://ClusterSystem@127.0.0.1:2551#-2899452638012261337], selfDc [default].
[ClusterSystem-akka.actor.default-dispatcher-3] WARN akka.stream.Materializer - [outbound connection to [akka://ClusterSystem@127.0.0.1:2552], control stream] Upstream failed, cause: StreamTcpException: Tcp command [Connect(127.0.0.1/cvnsolved>:2552,None,List(),Some(5000 milliseconds),true)] failed because of java.net.ConnectException: Connection refused
[ClusterSystem-akka.actor.default-dispatcher-3] WARN akka.stream.Materializer - [outbound connection to [akka://ClusterSystem@127.0.0.1:2552], message stream] Upstream failed, cause: StreamTcpException: Tcp command [Connect(127.0.0.1/cvnsolved>:2552,None,List(),Some(5000 milliseconds),true)] failed because of java.net.ConnectException: Connection refused
[ClusterSystem-akka.actor.default-dispatcher-26] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2551] - Received InitJoin message from [Actor[akka://ClusterSystem@127.0.0.1:2552/system/cluster/core/daemon/joinSeedNodeProcess-1#679854750]], but this node is not initialized yet
[ClusterSystem-akka.actor.default-dispatcher-26] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2551] - Received InitJoin message from [Actor[akka://ClusterSystem@127.0.0.1:2552/system/cluster/core/daemon/joinSeedNodeProcess-1#679854750]] to [akka://ClusterSystem@127.0.0.1:2551]
[ClusterSystem-akka.actor.default-dispatcher-26] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2551] - Node [akka://ClusterSystem@127.0.0.1:2551] is JOINING itself (with roles [dc-default], version [0.0.0]) and forming new cluster
[ClusterSystem-akka.actor.default-dispatcher-26] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2551] - is the new leader among reachable nodes (more leaders may exist)
[ClusterSystem-akka.actor.default-dispatcher-26] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2551] - Leader is moving node [akka://ClusterSystem@127.0.0.1:2551] to [Up]
[ClusterSystem-akka.actor.default-dispatcher-26] INFO akka.cluster.sbr.SplitBrainResolver - This node is now the leader responsible for taking SBR decisions among the reachable nodes (more leaders may exist).
[ClusterSystem-akka.actor.default-dispatcher-3] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2551] - Received InitJoin message from [Actor[akka://ClusterSystem@127.0.0.1:2552/system/cluster/core/daemon/joinSeedNodeProcess-1#679854750]] to [akka://ClusterSystem@127.0.0.1:2551]
[ClusterSystem-akka.actor.default-dispatcher-3] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2551] - Sending InitJoinAck message from node [akka://ClusterSystem@127.0.0.1:2551] to [Actor[akka://ClusterSystem@127.0.0.1:2552/system/cluster/core/daemon/joinSeedNodeProcess-1#679854750]] (version [2.6.14])
[ClusterSystem-akka.actor.default-dispatcher-3] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2551] - Node [akka://ClusterSystem@127.0.0.1:2552] is JOINING, roles [dc-default], version [0.0.0]
[ClusterSystem-akka.actor.default-dispatcher-3] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2551] - Leader is moving node [akka://ClusterSystem@127.0.0.1:2552] to [Up]
akka://ClusterSystem/user/upper1 registered
akka://ClusterSystem/user/upper2 registered
HELLO
HELLO

```

- node b - tworzy serwis, który wykorzystując mechanizm recepcjonisty dostaje referencje do workerów, które są na nodzie a i po odpowiednio długiej przerwie wysyła do nich requesta: `hello`.

```

Z2_NodeB x Z2_NodeA x
/Users/petroniuss/Library/Java/JavaVirtualMachines/adopt-openjdk-14.0.2/Contents/Home/bin/java ...
Node B: config: Config(SimpleConfigObject({"akka":{"actor":{"provider":"cluster"},"cluster":{"downing-provider-class":"akka.cluster.sbr.SplitBrainResolverProvider"},"seed-nodes":["akka://ClusterSystem@127.0.0.1:2551","akka://ClusterSystem@127.0.0.1:2552"],"remote":{"artery":{"canonical":{"hostname":"127.0.0.1","port":2552}}}}}))
[ClusterSystem-akka.actor.default-dispatcher-3] INFO akka.remote.artery.tcp.ArteryTcpTransport - Remoting started with transport [Artery tcp]; listening on address [akka://ClusterSystem@127.0.0.1:2552] with UID [7915062842362609888]
[ClusterSystem-akka.actor.default-dispatcher-3] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2552] - Starting up, Akka version [2.6.14] ...
[ClusterSystem-akka.actor.default-dispatcher-3] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2552] - Registered cluster JMX MBean [akka:type=Cluster]
[ClusterSystem-akka.actor.default-dispatcher-3] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2552] - Started up successfully
[ClusterSystem-akka.actor.default-dispatcher-3] INFO akka.cluster.sbr.SplitBrainResolver - SBR started. Config: strategy [KeepMajority], stable-after [20 seconds], down-all-when-unstable [15 seconds], selfUniqueAddress [akka://ClusterSystem@127.0.0.1:2552#7915062842362609888], selfDc [default].
creating receive for text service
On ListingRequest
[ClusterSystem-akka.actor.default-dispatcher-5] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2552] - Received InitJoin message from [Actor[akka://ClusterSystem@127.0.0.1:2551/system/cluster/core/daemon/firstSeedNodeProcess-1#-557233915]], but this node is not initialized yet
[ClusterSystem-akka.actor.default-dispatcher-5] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2552] - Received InitJoin message from [Actor[akka://ClusterSystem@127.0.0.1:2551/system/cluster/core/daemon/firstSeedNodeProcess-1#-557233915]], but this node is not initialized yet
[ClusterSystem-akka.actor.default-dispatcher-5] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2552] - Received InitJoin message from [Actor[akka://ClusterSystem@127.0.0.1:2551/system/cluster/core/daemon/firstSeedNodeProcess-1#-557233915]], but this node is not initialized yet
[ClusterSystem-akka.actor.default-dispatcher-5] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2552] - Received InitJoin message from [Actor[akka://ClusterSystem@127.0.0.1:2551/system/cluster/core/daemon/firstSeedNodeProcess-1#-557233915]], but this node is not initialized yet
[ClusterSystem-akka.actor.default-dispatcher-5] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2552] - Received InitJoin message from [Actor[akka://ClusterSystem@127.0.0.1:2551/system/cluster/core/daemon/firstSeedNodeProcess-1#-557233915]], but this node is not initialized yet
[ClusterSystem-akka.actor.default-dispatcher-5] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2552] - Received InitJoinAck message from [Actor[akka://ClusterSystem@127.0.0.1:2551/system/cluster/core/daemon#-1482431624]] to [akka://ClusterSystem@127.0.0.1:2552]
[ClusterSystem-akka.actor.default-dispatcher-5] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2552] - Welcome from [akka://ClusterSystem@127.0.0.1:2551]
On ListingRequest
request: hello
sending to worker: Actor[akka://ClusterSystem@127.0.0.1:2551/user/upper1#77224913]
sending to worker: Actor[akka://ClusterSystem@127.0.0.1:2551/user/upper2#2077895553]

```