Systemy rozproszone - raport z zadań z zajęć 4 (Akka)

Mateusz Pawłowicz 305391

1 Zadanie 1

Pierwsze zadanie polegało na przetestowaniu różnych strategii obsługi błędów u aktorów. Poniżej zawarłem output oraz krótki opis wyników dla trzech przypadków, które należało przetestować w ramach ćwiczenia. W ramach testów do aktorów mnożących i dzielących komunikaty były wysyłane w następującej kolejności:

1. Poprawne mnożenie
2. Poprawne dzielenie
3. Dzielenie przez 0 (wyjątek)
4. Poprawne mnożenie
5. Poprawne dzielenie

1.1 Strategia stop

```
Strategy: stop
actorDivide: opCount = 1
actorMultiply: opCount = 1
actorMultiply: opCount = 2
[actorMath-akka.actor.default-dispatcher-6] ERROR akka.actor.typed.Behavior$ - Supervisor StopSupervisor saw failure: / by zero
[actorMath-akka.actor.default-dispatcher-3] INFO akka.actor.LocalActorRef - Message [Z1.MathActor$MathCommandDivide] to Actor[akka://actorMath/user/actorDivide#377959233] was not delivered.
```

Rysunek 1: Output dla strategii stop (ostatnia linijka została osobno dołączona, aby nie wklejać tutaj całej informacji o błędzie)

W strategii stop aktor, w którym wystąpił błąd jest zatrzymywany i przestaje obsługiwać wiadomości (nie wyświetliła się kolejna informacja o ilości wykonanych operacji w aktorze *actorDivide*). Na dole widać, że kolejna wysłana wiadomość do niego nie została dostarczona po wystąpieniu wyjątku (logger wypisał informację o tym). Strategia ta wydaje się być dobra w momencie, kiedy wystąpienie błędu w aktorze sprawia, że nie może on dalej funkcjonować, bądź dalsze działanie mogłoby zepsuć pewne dane, więc najlepiej zatrzymać aktora.

1.2 Strategia resume

```
[actorMath-akka.actor.default-dispatcher-6] INFO akka.event.slf4j.Slf4jLogger - Slf4jLogger started
Strategy: resume
actorMultiply: opCount = 1
actorDivide: opCount = 1
actorMultiply: opCount = 2
actorDivide: opCount = 2
[actorMath-akka.actor.default-dispatcher-3] ERROR akka.actor.typed.Behavior$ - Supervisor ResumeSupervisor saw failure: / by zero
```

Rysunek 2: Output dla strategii resume

W tym przypadku pomimo wystąpienia błędu, aktor dalej funkcjonuje i zachowuje stan taki jakby błęda wiadomość nie została obsłużona. *actorDivide* wykonał jedną operację, która była poprawna, jedną błędną i kolejną poprawną - finalnie ilość wykonanych operacji to 2 (tyle ile poprawnych było sumarycznie). Na dole widzimy log o błędzie, który wystąpił w drugim zleceniu do *actorDivide*. Strategia ta pozwala na niezawodność i spójne działanie, o ile zlecenia do aktora są niezależne (nieobsłużenie jednego nie będzie wpływało na dalsze działanie).

1.3 Strategia restart

```
[actorMath-akka.actor.default-dispatcher-6] INFO akka.event.slf4j.Slf4jLogger - Slf4jLogger started
Strategy: restart
actorDivide: opCount = 1
actorMultiply: opCount = 1
actorMultiply: opCount = 2
actorDivide: opCount = 1
[actorMath-akka.actor.default-dispatcher-6] ERROR akka.actor.typed.Behavior$ - Supervisor RestartSupervisor saw failure: / by zero
```

Rysunek 3: Output dla strategii restart

Strategia restart w przypadku wystąpienia błędu uruchamia aktora jeszcze raz. Po ponownym uruchomieniu stan aktora jest resetowany (po obsłudze drugie poprawnego zlecenia stan to jedna wykonana operacja), ale aktor dalej działa po błędzie. Jest to dobry wybór, kiedy zależy nam na niezawodności, ale jedno zepsute zlecenie mogłoby zepsuć kolejne - dobrym pomysłem jest wtedy uruchomienie aktora od zera.

2 Zadanie 2

Drugie zadanie miało na celu zapoznanie się z mechanizmem rejestrowania i subskrybowania recepcjonisty oraz działania w klastrze.

2.1 Z2_Main

```
[z2main-akka.actor.default-dispatcher-5] INFO akka.event.slf4j.Slf4jLogger - Slf4jLogger started
akka://z2main/user/upper2 registered
akka://z2main/user/upper1 registered
creating receive for text service
request: hello
sending to worker: Actor[akka://z2main/user/upper1#247885551]
sending to worker: Actor[akka://z2main/user/upper2#361450100]
HELLO
HELLO
```

Rysunek 4: Output z Z2_Main

Powyżej widać poprawne działanie programu - po stworzeniu aktorzy są rejestrowani u recepcjonisty co jest przechwytywane w serwisie. Następnie do serwisu jest przekazywane pojedyncze zlecenie, które jest przekierowywane do wszystkich aktorów - każdy z nich wypisuje *hello* wielkim literami.

2.2 Z2_NodeA i Z2_NodeB (klaster)

```
Node A: config: Config(SimpleConfigObject({"akka":{"actor":{"provider":"cluster"},"cluster":{"downing-provider-class":"akka.cluster.sbr.SplitBrainResolverProvider",
[ClusterSystem-akka.actor.default-dispatcher-5] INFO akka.event.slf4j.Slf4jLogger - Slf4jLogger started
[ClusterSystem-akka.actor.default-dispatcher-3] INFO akka.remote.artery.tcp.ArteryTcpTransport - Remoting started with transport [Artery tcp]; listening on address
[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:
[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-14] INFO akka.cluster.sbr.SplitBrainResolver - SBR started. Config: strategy [KeepMajority], stable-after [20 seconds],
akka://ClusterSystem/user/upper1 registered
akka://ClusterSystem/user/upper2 registered
[ClusterSystem-akka.actor.default-dispatcher-5] WARN akka.stream.Materializer - [outbound connection to [akka://ClusterSystem@127.0.0.1:2552], message stream] Upstr
[ClusterSystem-akka.actor.default-dispatcher-5] WARN akka.stream.Materializer - [outbound connection to [akka://ClusterSystem@127.0.0.1:2552], control stream] Upstr
[ClusterSystem-akka.actor.default-dispatcher-5] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2551] - Node [akka://ClusterSystem@127.0.0.
[ClusterSystem-akka.actor.default-dispatcher-5] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2551] - is the new leader among reachable n
[ClusterSystem-akka.actor.default-dispatcher-5] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2551] - Leader is moving node [akka://Clust
[ClusterSystem-akka.actor.default-dispatcher-5] INFO akka.cluster.sbr.SplitBrainResolver - This node is now the leader responsible for taking SBR decisions among th
[ClusterSystem-akka.actor.default-dispatcher-14] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2551] - Received InitJoin message from [Ac
[ClusterSystem-akka.actor.default-dispatcher-14] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2551] - Sending InitJoinAck message from n
[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.
[ClusterSystem-akka.actor.default-dispatcher-5] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2551] - Leader is moving node [akka://Clust
HELLO
HELLO
```

```
Node B: config: Config(SimpleConfigObject({"akka":{"actor":{"provider":"cluster"},"cluster":{"downing-provider-class":"akka.cluster.sbr.SplitBrainResolverProv
[ClusterSystem-akka.actor.default-dispatcher-3] INFO akka.event.slf4j.Slf4jLogger - Slf4jLogger started
SLF4J: A number (1) of logging calls during the initialization phase have been intercepted and are
SLF4J: now being replayed. These are subject to the filtering rules of the underlying logging system.
SLF4J: See also http://www.slf4j.org/codes.html#replay
[ClusterSystem-akka.actor.default-dispatcher-12] INFO akka.remote.artery.tcp.ArteryTcpTransport - Remoting started with transport [Artery tcp]; listening on a
[ClusterSystem-akka.actor.default-dispatcher-3] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2552] - Starting up, Akka version [2.
[ClusterSystem-akka.actor.default-dispatcher-3] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2552] - Registered cluster JMX MBean
[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-5] INFO akka.cluster.sbr.SplitBrainResolver - SBR started. Config: strategy [KeepMajority], stable-after [20 sec
creating receive for text service
[ClusterSystem-akka.actor.default-dispatcher-5] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2552] - Received InitJoin message fr
[ClusterSystem-akka.actor.default-dispatcher-5] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2552] - Received InitJoin message fr
[ClusterSystem-akka.actor.default-dispatcher-5] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2552] - Received InitJoin message fr
[ClusterSystem-akka.actor.default-dispatcher-5] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2552] - Received InitJoin message fr
[ClusterSystem-akka.actor.default-dispatcher-5] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2552] - Received InitJoin message fr
[ClusterSystem-akka.actor.default-dispatcher-5] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2552] - Received InitJoinAck message
[ClusterSystem-akka.actor.default-dispatcher-12] INFO akka.cluster.Cluster - Cluster Node [akka://ClusterSystem@127.0.0.1:2552] - Welcome from [akka://Cluster
request: hello
sending to worker: Actor[akka://ClusterSystem@127.0.0.1:2551/user/upper1#-1329302215]
sending to worker: Actor[akka://ClusterSystem@127.0.0.1:2551/user/upper2#-1641592413]
```

Rysunek 5: Output z Z2_NodeA i Z2_NodeB

W przypadku klastra również udało się uzyskać poprawne działanie systemu - dodani aktorzy są przechwytywani do listy w serwisie i serwis może do nich rozsyłać zlecenia (ponownie wysyłane jest *hello* i widać poprawny wynik działania aktorów - ten sam napis wielkimi literami). W tym przypadku udało się uzyskać poprawne działanie w przypadku, gdy działanie systemu było rozdzielone pomiędzy dwa procesy (węzły) - w systemie rozproszonym.