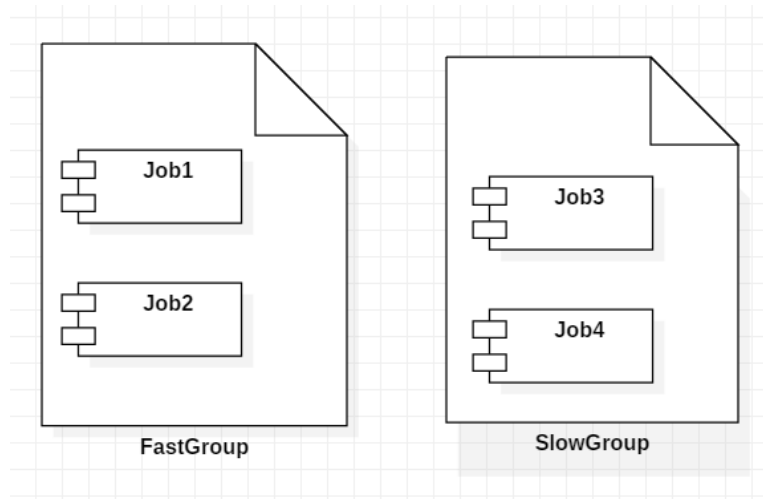


Dispatcher



Akka Dispatcher	Task
-----------------	------

. JAVA, C# .

: 100 10 .

: 1 .

100 , 100 () 101 .

```
fast-dispatcher {
type = Dispatcher
executor = "fork-join-executor"
fork-join-executor {
    parallelism-min = 5
    parallelism-factor = 2.0
    parallelism-max = 10
}
throughput = 100
}

slow-dispatcher {
type = Dispatcher
executor = "fork-join-executor"
fork-join-executor {
    parallelism-min = 1
    parallelism-factor = 2.0
    parallelism-max = 1
}
throughput = 10
}
```

;

```

public class SmartActor extends AbstractActor {
    @Override
    public Receive createReceive() {
        return receiveBuilder()
            .match(Integer.class, number -> { //
                Thread.sleep(500); // delay for test
                log.info(String.format("==== process msg:%s", name));
            })
            .matchAny(o -> log.info("received unknown message"))
            .build();
    }
}

@Test
public void testIt() throws InterruptedException {
    new TestKit(system) {{
        ActorRef probe = getRef();
        //6 ()
        ActorRef fastActor1 = system.actorOf(SmartActor.props("i'am fast").withDispatcher("fast-dispatcher"),
"fast1");
        ActorRef fastActor2 = system.actorOf(SmartActor.props("i'am fast").withDispatcher("fast-dispatcher"),
"fast2");
        ActorRef fastActor3 = system.actorOf(SmartActor.props("i'am fast").withDispatcher("fast-dispatcher"),
"fast3");

        ActorRef slowActor1 = system.actorOf(SmartActor.props("i'am slow").withDispatcher("slow-dispatcher"),
"slow1");
        ActorRef slowActor2 = system.actorOf(SmartActor.props("i'am slow").withDispatcher("slow-dispatcher"),
"slow2");
        ActorRef slowActor3 = system.actorOf(SmartActor.props("i'am slow").withDispatcher("slow-dispatcher"),
"slow3");

        for (int i = 0; i < 1000; i++) {
            fastActor1.tell(i, ActorRef.noSender());
            fastActor2.tell(i, ActorRef.noSender());
            fastActor3.tell(i, ActorRef.noSender());

            slowActor1.tell(i, ActorRef.noSender());
            slowActor2.tell(i, ActorRef.noSender());
            slowActor3.tell(i, ActorRef.noSender());
        }
    }};
}

```

:

```

[INFO] [04/02/2019 17:47:03.455] [TestSystem-fast-dispatcher-18] [akka://TestSystem/user/fast3] ===== process
msg:i'am fast
[INFO] [04/02/2019 17:47:03.455] [TestSystem-fast-dispatcher-17] [akka://TestSystem/user/fast2] ===== process
msg:i'am fast
[INFO] [04/02/2019 17:47:03.455] [TestSystem-fast-dispatcher-16] [akka://TestSystem/user/fast1] ===== process
msg:i'am fast
[INFO] [04/02/2019 17:47:03.456] [TestSystem-slow-dispatcher-21] [akka://TestSystem/user/slow3] ===== process
msg:i'am slow
[INFO] [04/02/2019 17:47:03.955] [TestSystem-fast-dispatcher-18] [akka://TestSystem/user/fast3] ===== process
msg:i'am fast
[INFO] [04/02/2019 17:47:03.955] [TestSystem-fast-dispatcher-16] [akka://TestSystem/user/fast1] ===== process
msg:i'am fast
[INFO] [04/02/2019 17:47:03.955] [TestSystem-fast-dispatcher-17] [akka://TestSystem/user/fast2] ===== process
msg:i'am fast
[INFO] [04/02/2019 17:47:03.956] [TestSystem-slow-dispatcher-21] [akka://TestSystem/user/slow1] ===== process
msg:i'am slow

10 , slow 1 1 2
fast 5 .

```

Dispatcher .

:

- : <https://github.com/psmon/springcloud/blob/master/lobbyapi/src/test/java/com/webnori/psmon/cloudspring/lobbyapi/actor/SmartActorTest.java>
- java : <https://doc.akka.io/docs/akka/current/dispatchers.html>
- .net : <https://getakka.net/articles/actors/dispatchers.html>
- (BackPressure) : <https://chariotsolutions.com/blog/post/simply-explained-akka-streams-backpressure/>