**Comparação inicial Akka vs Spring Reactor:**

**Utilizei para ambos uma massa de dados Fake de 200000 registros para cada.**

Ambos usam a especificação Reactive Streams Specification (Streams reativas)

Streams reativos é uma iniciativa para fornecer um padrão para processamento de fluxo assíncrono com pressão de retorno não-bloqueante. Isso engloba esforços voltados para ambientes de tempo de execução (JVM e JavaScript), bem como protocolos de rede.

O que queremos é assíncrono e não-bloqueante. Um grupo de desenvolvedores de empresas como Netflix, Pivotal, RedHat, etc. reuniram-se e convergiram para algo chamado The Reactive Streams Specification.

****

**Akka framework**

Akka é um framework Opensource.

Akka é implementado a partir do modelo de agente (originado em 1973 por Carl Hewitt), que fornece um modelo para lidar com computação simultânea.

Akka, a unidade de execução é o Ator e seus microsserviços são construídos como atores. Um ator é leve e pode haver vários milhões de atores por GB de memória heap.

O ator é um objeto que encapsula estado e comportamento e se comunica exclusivamente trocando mensagens que são colocadas na caixa de correio do destinatário.

Sistemas Simultâneos Concorrentes e Distribuídos

Atores e Streams permitem que você construa sistemas que aumentam de tamanho, usando os recursos de um servidor com mais eficiência e usando vários servidores.

Resiliente pelo design

Baseando-se nos princípios do Manifesto Reativo, o Akka permite que você escreva sistemas que se auto-curam e permaneçam responsivos em face de falhas.

Alta performance

Até 50 milhões de msg / seg em uma única máquina. Pegada de memória pequena; ~ 2,5 milhões de atores por GB de heap.

Elástico e Descentralizado

Sistemas distribuídos sem pontos únicos de falha. Balanceamento de carga e roteamento adaptável entre nós. Fornecimento de Eventos e CQRS com Compartilhamento de Cluster. Dados Distribuídos para consistência eventual usando CRDTs.

Dados de Transmissão Reativa

Processamento assíncrono de fluxo não bloqueador com contrapressão. Servidor e cliente HTTP totalmente assíncronos e de streaming fornecem uma excelente plataforma para construir microsserviços. Integrações de streaming com Alpakka.

**Métrica de tempo usando 200000 registros para ambos:**

Akka:

-------------------------------------------------------------------------------

Test set: com.lightbend.akka.sample.AkkaQuickstartTest

-------------------------------------------------------------------------------

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.766 sec

TEMPO INICIAL: [05/10/2019 10:10:49.708]

TEMPO FINAL: [05/10/2019 10:11:03.081]

Média de 13,5 segundos

****

**Spring Reactor**

Assim como Akka o Spring também é um framework Opensource.

O Project Reactor é a implementação do Spring do The Reactive Specification e é especificamente favorecido pelo módulo Spring Webflux, embora você possa usá-lo com outros módulos como o RxJava.

A ideia é operar de forma assíncrona com Backpressure usando Editores e Assinantes.

Aqui, estamos sendo apresentados a vários novos conceitos! Vamos explicá-los um por um:

Editor - Um Publicador é um provedor de um número potencialmente ilimitado de elementos.

Assinante - Um Assinante escuta esse Publicador, solicitando novos dados. Às vezes, também é chamado de consumidor.

Backpressure - A capacidade do Assinante de permitir ao Publicador o número de solicitações que ele pode manipular no momento. Portanto, é o Assinante responsável pelo fluxo dos dados, não pelo Publicador, pois apenas fornece os dados.

O Projeto Reator oferece dois tipos de editores. Estes são considerados os principais blocos de construção do Spring Webflux:

Flux - é um editor que produz valores de 0 a N. Pode ser ilimitado. Operações que retornam vários elementos usam esse tipo.

Mono - é um editor que produz um valor de 0 a 1. Operações que retornam um único elemento usam esse tipo.

**Métrica de tempo usando 200000 registros para ambos:**

Spring Reactor:

-------------------------------------------------------------------------------

Test set: reactivewebconsumer.reactivewebconsumer.ReactiveWebConsumerApplicationTests

-------------------------------------------------------------------------------

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2.558 s - in reactivewebconsumer.reactivewebconsumer.ReactiveWebConsumerApplicationTests

TEMPO INICIAL: 2019-05-10T12:53:24.858Z

TEMPO FINAL: 2019-05-10T12:59:54.453Z

Média de 30 segundos

Conclusão:

Minha conclusão pessoal, o Akka é um framework mais robusto, porque eu escolheria o Akka:

- Melhor performance

- Maior suporte a carga de dados

- Mais rápido

O Spring Reactor, porque eu o escolheria:

- Facilidade de implementar

- Maior Produtividade ao desenvolver

- Facilidade de manutenção