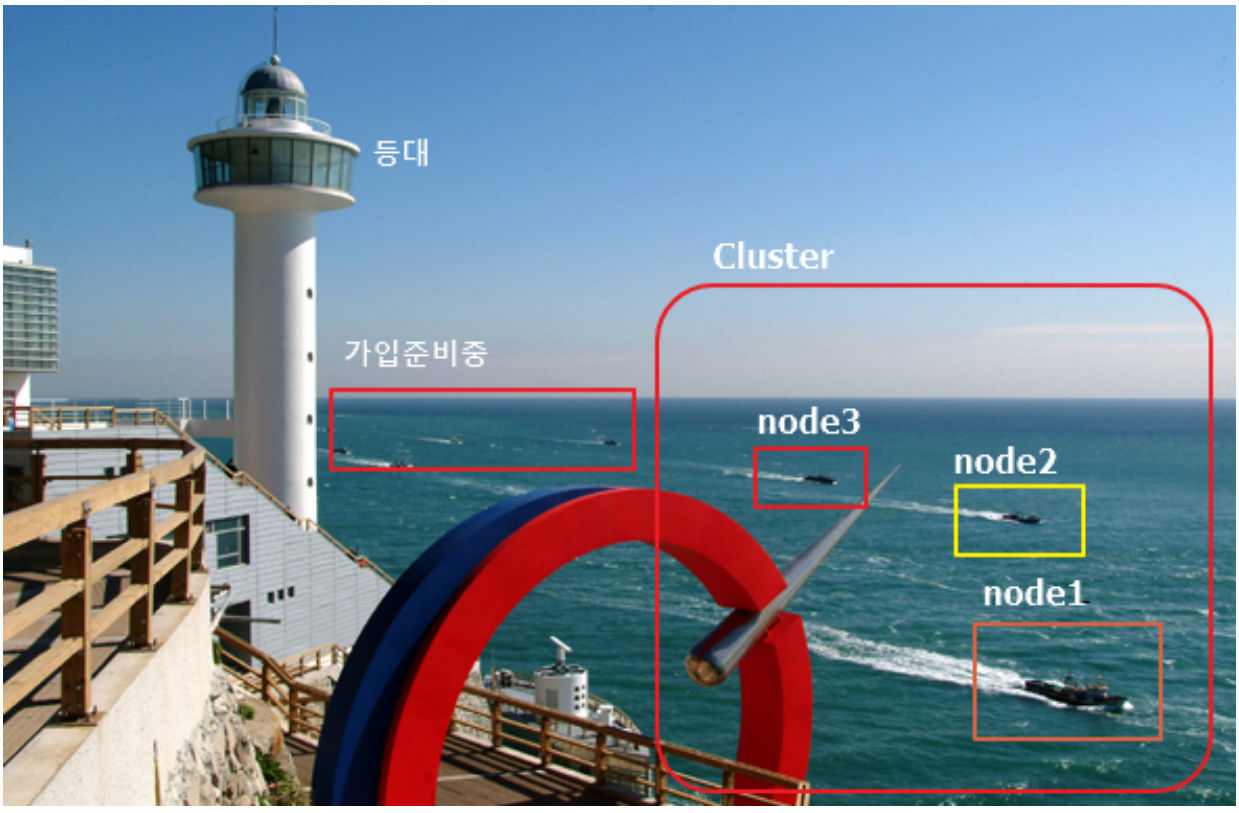
# Cluster with Actor

Akka ,

.

, .

- 
  - OverView
    - Gossip Protocol
- RoundRobin
- Broadcast
- ConsistentHashing
- ScatterGatherFirstCompleted
- SmallestMailbox
- TailChopping
-
-
  -
  -
  -
-
-



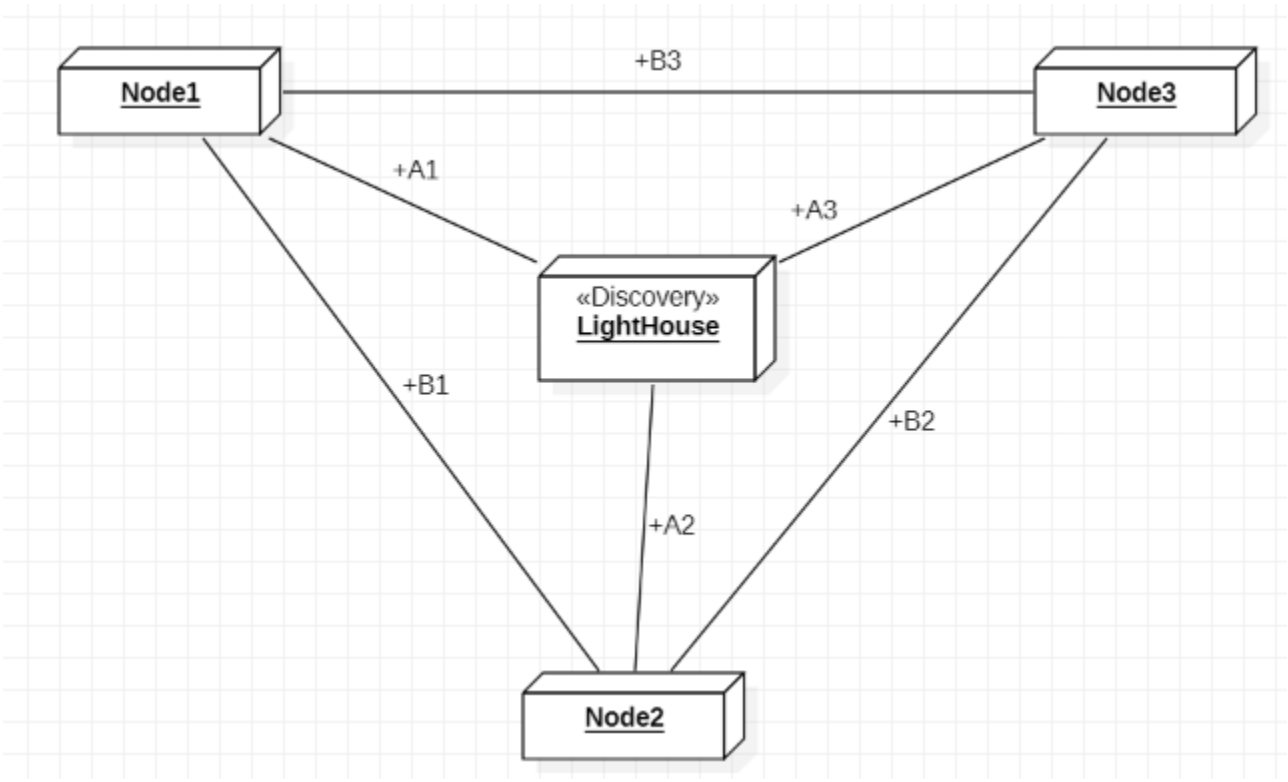

LightHouse() (Discovery) / ,

, .

> *Rancher(),()*
>
> , .
>
> *Akka.net* .

git : https://github.com/psmon/AkkaForNetCore/tree/master/LightHouse

## OverView




### Gossip Protocol

, ( )

(Gossip) .

- Node1 .
  - A1 : Node1 , .
- Node2
  - A2 : 2 , .
  - A1 : Node1 Node2 .
  - B1 : Node1 Node2 .
  - A2 : .
- Node3 .
  - : A3 (A1,A2) (B3,B2) A3
- Node1 (Down).
  - A1 : Node1 . ( Node1 )
  - A3,A2 : Node1 .
  - B3,B1 : Node1 .

,

.

- , .
- () , .
- , , .

, .

, AKKA .

.

Down .

AKKA .

*AKKA .*

*Down*

*. AKKA ,*

*. ( .)*

- *AKKA IT .*

## RoundRobin

,

ex> RestAPI ( Pool Node Crash )

## Broadcast

,

## ConsistentHashing

ex>

- ( )
- ( , )
- ( )
- SSL ( )
- X ( ) API n TPS . .
  - Redis : Redis Key/Value RDB ... . / . .

## ScatterGatherFirstCompleted

,

ex> 1 ,

## SmallestMailbox

## TailChopping

, (5s)

~
> GC GC CPU . .

https://github.com/reactive-streams

- https://github.com/akka/reactive-kafka
- https://blog.knoldus.com/2017/06/18/when-akka-stream-meets-rabbitmq/

( Java/C#/SCALA/JS )

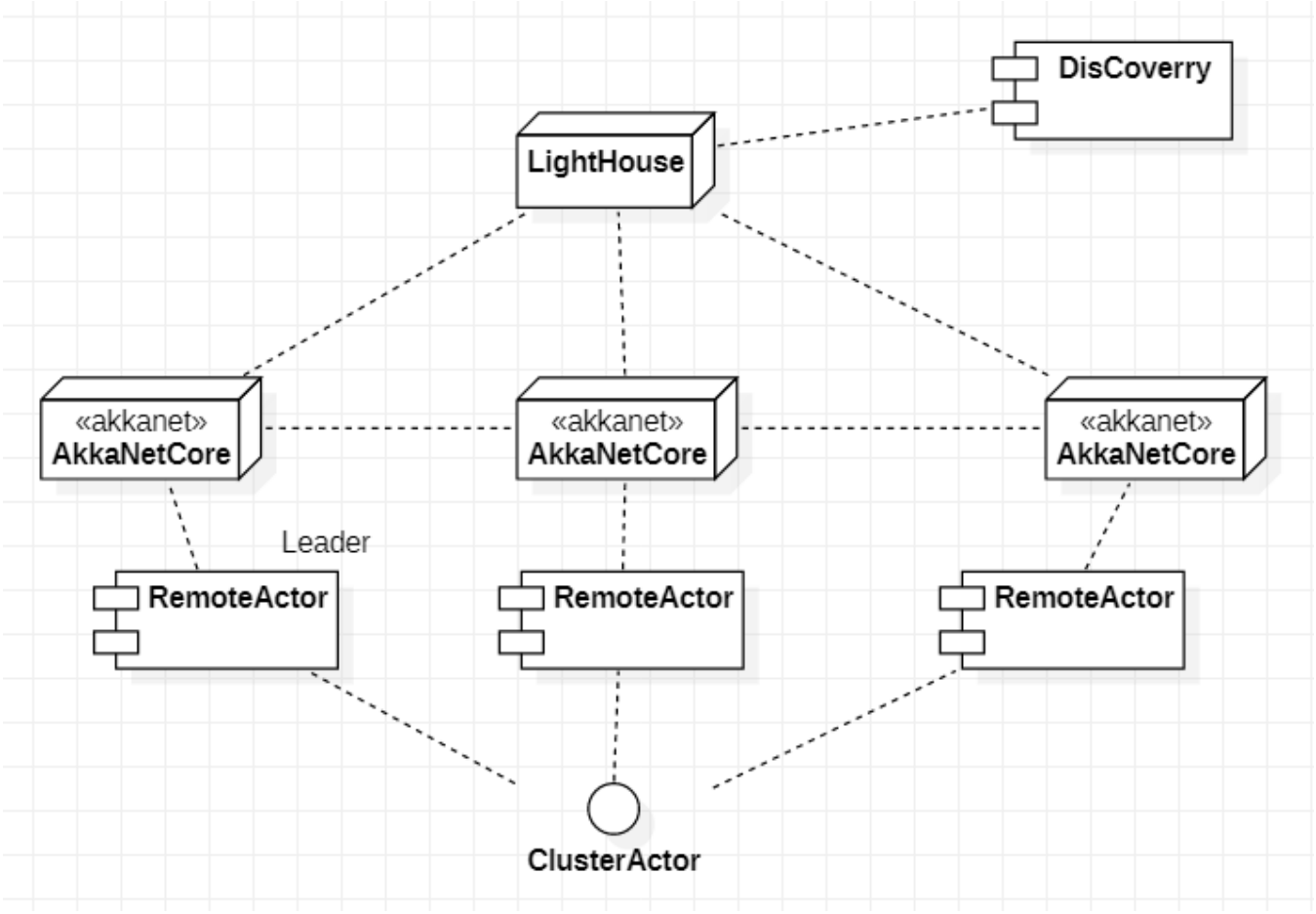
, reactive-streams .

akka stream /,

.

, .

.


DisCoverry
LightHouse
«akkanet»
AkkaNetCore
«akkanet»
AkkaNetCore
«akkanet»
AkkaNetCore
Leader
RemoteActor
RemoteActor
RemoteActor
ClusterActor

```
akka {
        remote {
                log-remote-lifecycle-events = debug
        dot-netty.tcp {
            port = 7000
                        hostname = 127.0.0.1
                }
    }

        actor {
                provider = cluster
                deployment {
                        /cluster-roundrobin {
                                router = round-robin-pool # routing strategy
                                #routees.paths = ["/user/clustermsg"]
                                nr-of-instances = 500 # max number of total routees
                                cluster {
                                        enabled = on
                                        allow-local-routees = on
                                        use-role = akkanet
                                        max-nr-of-instances-per-node = 20
                                }
                        }
                }
        }

        cluster {
        seed-nodes = ["akka.tcp://actor-cluster@127.0.0.1:7100"] # address of seed node
        roles = ["akkanet"] # roles this member is in
        auto-down-unreachable-after = 300s
                debug {
                        verbose-heartbeat-logging = off
                        verbose-receive-gossip-logging = off
                }
    }
}
```

- actor.privider = cluster :  .
- cluster-roundrobin.cluster.user-role =  : ,
- cluster.seed-nodes :
- cluster.roles :  ()

```
protected Cluster Cluster = Akka.Cluster.Cluster.Get(Context.System);
private bool ClusterMode = true;

protected override void PreStart()
{
        // subscribe to IMemberEvent and UnreachableMember events
        if (ClusterMode)
        {
                Cluster.Subscribe(Self, ClusterEvent.InitialStateAsEvents,
                new[] { typeof(ClusterEvent.IMemberEvent), typeof(ClusterEvent.UnreachableMember) });
        }
}

protected override void PostStop()
{
        if (ClusterMode) Cluster.Unsubscribe(Self);
}
// src : https://github.com/psmon/AkkaForNetCore/blob/master/AkkaNetCore/Actors/ClusterMsgActor.cs
```

, (Gossip)  .

Received .

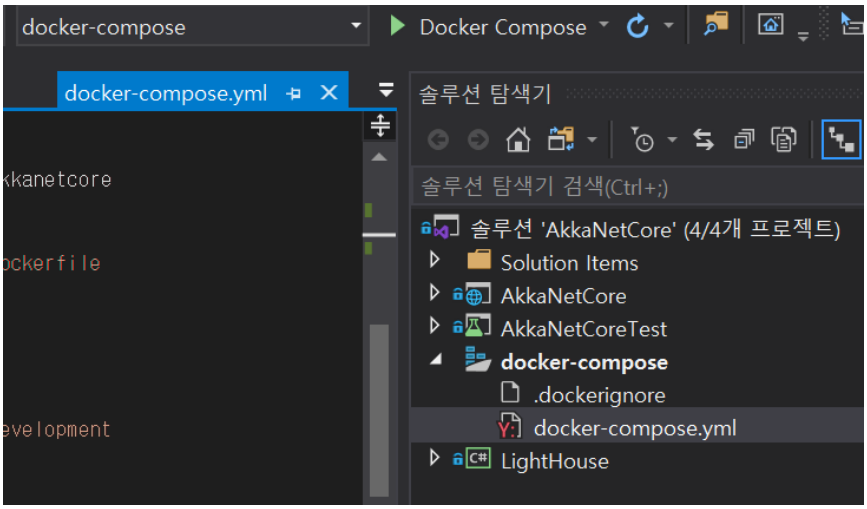
```
//    : https://github.com/psmon/AkkaForNetCore/blob/master/AkkaNetCore/Startup.cs

AkkaLoad.RegisterActor(
        "clusterRoundRobin",
        actorSystem.ActorOf(Props.Create<ClusterMsgActor>()
                        .WithDispatcher("fast-dispatcher")
                        .WithRouter(FromConfig.Instance),
                        "cluster-roundrobin"
));


//   DI      : https://github.com/psmon/AkkaForNetCore/blob/master/AkkaNetCore/Controllers/ActorTestController.
cs

private readonly IActorRef clusterRoundbin1;
public ActorTestController()
{
        clusterRoundbin1 = AkkaLoad.ActorSelect("clusterRoundRobin");
}

[HttpPost("/cluster/msg/tell")]
public void ClusterMsg(string value, int count)
{
        for (int i = 0; i < count; i++)
                clusterRoundbin1.Tell(value);
}
```




, N ,

.

IDE Docker(+Compose,+)

.

DataDog(https://www.datadoghq.com/) . ( )

**docker-compose.yml**

```
version: '3.4'

services:
  datadog:
    image: datadog/agent:7
    ports:
      - 8125/udp
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock:ro
      - /proc/:/host/proc/:ro
      - /sys/fs/cgroup/:/host/sys/fs/cgroup:ro
    environment:
      DD_API_KEY: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
      DD_DOGSTATSD_NON_LOCAL_TRAFFIC: "true"

  lighthouse:
    image: ${DOCKER_REGISTRY-}lighthouse
    build:
      context: .
      dockerfile: LightHouse/Dockerfile
    environment:
      CLUSTER_IP: lighthouse
      CLUSTER_PORT: 4053
      CLUSTER_SEEDS: akka.tcp://actor-cluster@lighthouse:4053

  akkanetcore:
    image: ${DOCKER_REGISTRY-}akkanetcore
    build:
      context: .
      dockerfile: AkkaNetCore/Dockerfile
    depends_on:
     - lighthouse
    ports:
     - 8080:5000
    environment:
      ASPNETCORE_ENVIRONMENT: Development
      MonitorTool: datadog
      MonitorToolCon: datadog
      port: 5000
      akkaport: 7100
      akkaip: akkanetcore
      roles: "akkanet"
      akkaseed: "akka.tcp://actor-cluster@lighthouse:4053"

  akkanetcore2:
    image: ${DOCKER_REGISTRY-}akkanetcore
    build:
      context: .
      dockerfile: AkkaNetCore/Dockerfile
    depends_on:
     - lighthouse
    environment:
      ASPNETCORE_ENVIRONMENT: Development
      MonitorTool: datadog
      MonitorToolCon: datadog
      port: 5000
      akkaport: 7000
      akkaip: akkanetcore2
      roles: "akkanet"
      akkaseed: "akka.tcp://actor-cluster@lighthouse:4053"

  akkanetcore3:
    image: ${DOCKER_REGISTRY-}akkanetcore
    build:
      context: .
      dockerfile: AkkaNetCore/Dockerfile
    depends_on:
     - lighthouse
    environment:
      ASPNETCORE_ENVIRONMENT: Development
      MonitorTool: datadog
```

```
      MonitorToolCon: datadog
      port: 5000
      akkaport: 7000
      akkaip: akkanetcore3
      roles: "akkanet"
      akkaseed: "akka.tcp://actor-cluster@lighthouse:4053"
```

, .

()

.

.

Links :

- https://getakka.net/articles/clustering/cluster-overview.html
- https://getakka.net/articles/clustering/cluster-sharding.html
- https://getakka.net/articles/clustering/distributed-data.html