
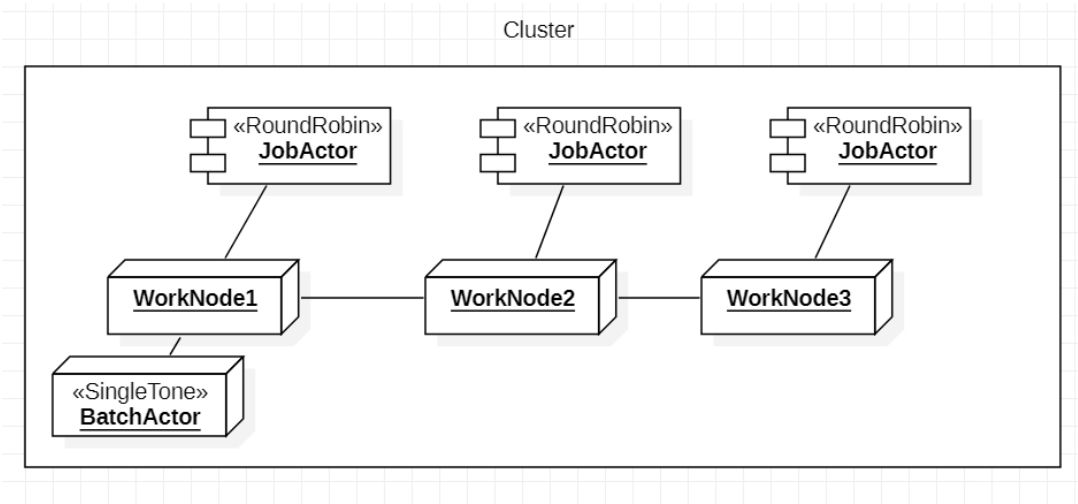


SingleTon Cluster Actor





Job ,
RDB .
TPS , Db .
N
, 3
.
• : (, ,)
• FSM Actor : . . .
• Insert : , . . .

Child .

• a,b,c 3 a .
• a , b c .
• , Role .

```

using System;
using System.Collections.Generic;
using Akka.Actor;

namespace AkkaNetCore.Actors
{
    public class SingleToneActor : ReceiveActor
    {
        private readonly ILoggingAdapter logger = Context.GetLogger();
        private readonly string id;
        private IActorRef BatchWriter_Rev;
        private IActorRef BatchManager_Rev;

        public SingleToneActor()
        {
            BatchWriter_Rev = Context.ActorOf<BatchWriterActor>();
            BatchManager_Rev = Context.ActorOf(Props.Create(() => new BatchActor(10)));
            BatchManager_Rev.Tell(new SetTarget(BatchWriter_Rev));

            id = Guid.NewGuid().ToString();
            logger.Info($"    :{id}");
            startTime = DateTime.Now;
            totalCount = 0;

            ReceiveAsync<DelayMsg>(async msg =>
            {
                BatchManager_Rev.Tell(new Queue(msg));
            });
        }
    }
}
//
var actor = AkkaBootstrap.BootstrapSingleton<SingleToneActor>(actorSystem, "SingleToneActor", "akkanet");
SingleToneActor = AkkaBootstrap.BootstrapSingletonProxy(actorSystem, "SingleToneActor",
    "akkanet", "/user/SingleToneActor", "singleToneActorProxy");

```

Child (FSM) .

FMS

BatchActor , .

, BatchActor , Actor Actor .

```

namespace AkkaNetCoreTest.Actors
{
    //
    public class TestBatchWriterActor : ReceiveActor
    {
        protected IActorRef probe;

        public TestBatchWriterActor(IActorRef _probe)
        {
            probe = _probe;
            ReceiveAsync<object>(async message =>
            {
                if (message is Batch batchMessage)
                {
                    probe.Tell(batchMessage);
                    //Insert
                    Console.WriteLine($"===== TODO :{batchMessage.Obj.Count}");
                }
            });
        }
    }
}

```

```

class BatchActorTest : TestKit
{
    protected TestProbe probe;

    [SetUp]
    public void Setup()
    {
        //
        probe = this.CreateTestProbe();
    }

    // : DB , ( )
    // 3(collectSec) ..
    [TestCase(3)]
    public void LazyBatchAreOK(int collectSec)
    {
        var batchActor = Sys.ActorOf(Props.Create(() => new BatchActor(collectSec)));

        // :
        IActorRef batchWriterActor = Sys.ActorOf(Props.Create(() => new TestBatchWriterActor(probe)));
        batchActor.Tell(new SetTarget(batchWriterActor));

        // .
        batchActor.Tell(new Queue("1"));
        batchActor.Tell(new Queue("2"));
        batchActor.Tell(new Queue("3"));

        //
        probe.ExpectNoMsg();

        // : collectSec+1
        var batchList = probe.ExpectMsg<Batch>(TimeSpan.FromSeconds(collectSec+1)).Obj;

        var firstItem = batchList[0] as string;
        Assert.AreEqual("1", firstItem);
        Assert.AreEqual(3, batchList.Count);

        // .
        batchActor.Tell(new Queue("4"));
        batchActor.Tell(new Queue("5"));
        batchActor.Tell(new Queue("6"));
        batchActor.Tell(new Queue("7"));

        //
        batchActor.Tell(new Flush());

        //
        batchList = probe.ExpectMsg<Batch>().Obj;
        firstItem = batchList[0] as string;
        Assert.AreEqual("4", firstItem);
        Assert.AreEqual(4, batchList.Count);
    }

    [TestCase(3,2)]
    public void LazyBatchAreEmpty(int collectSec,int cutoffSec)
    {
        var batchActor = Sys.ActorOf(Props.Create(() => new BatchActor(collectSec)));
        // :
        IActorRef batchWriterActor = Sys.ActorOf(Props.Create(() => new TestBatchWriterActor(probe)));
        batchActor.Tell(new SetTarget(batchWriterActor));

        // .
        batchActor.Tell(new Queue("1"));
        batchActor.Tell(new Queue("2"));
        batchActor.Tell(new Queue("3"));

        //cutoffSec .
        probe.ExpectNoMsg(TimeSpan.FromSeconds(cutoffSec));
    }
}

```

```
}
```

Flush VS Clear

Clear , Flush .
Flush , .

BulkInsert

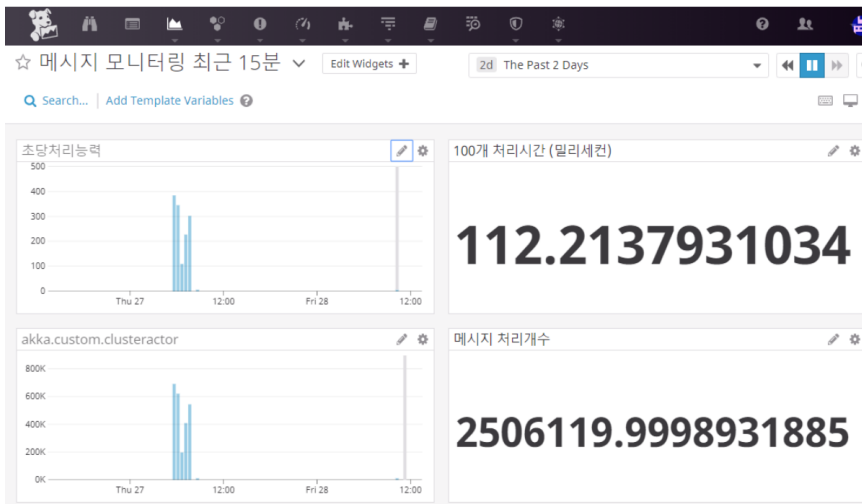
Performance Comparisons

Operations	1,000 Entities	2,000 Entities	5,000 Entities
SaveChanges	1,000 ms	2,000 ms	5,000 ms
BulkInsert	6 ms	10 ms	15 ms

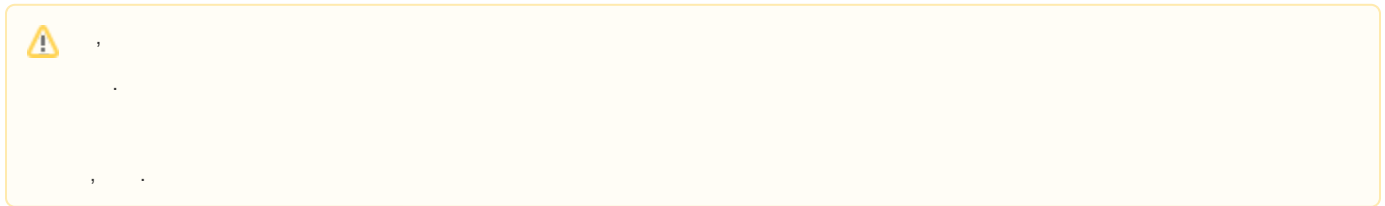
1 Insert , ORM (->DB) .
BulkInsert , ORM? SQL? .
ORM . (Message Object Entity Object)

AKKA .., 1

```
- .  
  
if (bulkItems_completed.Count > 0 && IsWriteDB)  
{  
    BatchType = "completed";  
    EntityFrameworkManager.ContextFactory = context => new BatchRepository(Startup.AppSettings);  
    using (var context = new BatchRepository(Startup.AppSettings))  
    {  
        await context.BulkInsertAsync(bulkItems_completed, options => {  
            options.BatchSize = BatchSize;  
        });  
        Context.IncrementCounter("akka.custom.received1", bulkItems_completed.Count);  
    }  
}
```



: Real time performance counters



:

- : <https://github.com/psmon/AkkaForNetCore/blob/master/AkkaNetCore/Actors/Study/SingleToneActor.cs>
- FSM : <https://github.com/psmon/AkkaForNetCore/blob/master/AkkaNetCore/Actors/Utils/BatchActor.cs>
- : <https://github.com/psmon/AkkaForNetCore/blob/025e04a74dba7389cc7675ebaccd403d959a129b/AkkaNetCore/Startup.cs#L184>
- : <https://github.com/psmon/AkkaForNetCore/blob/master/AkkaNetCore/akka.Development.conf>
- Insert : <https://entityframeworkcore.com/saving-data-bulk-insert>