Reproductible benchmarks

Let’s talk tooling

Not fancy crypto, but at least it is easy to understand!

https://github.com/dgarage/LightningBenchmarks

Nicolas DORIER

Code monkey at DG Lab and Metaco

Main maintainer of NBitcoin and BTCPay Server

I was asked

- Tell you tell me about Lightning performance?

I reformulated

- How fast can Alice pay Bob?
- How fast can Alice pay Bob through Carol?
- How fast Alices can pay Bob?

I responded

c-lightning in may 2018

- 34 payments / sec
- 12 payments / sec
- 20 payments / sec (with 4 Bobs)

HE CAN PROCESS 34 PAYMENTS PER SECONDS
SEE? NOBODY CARE
imgflip.com

Does it matter?

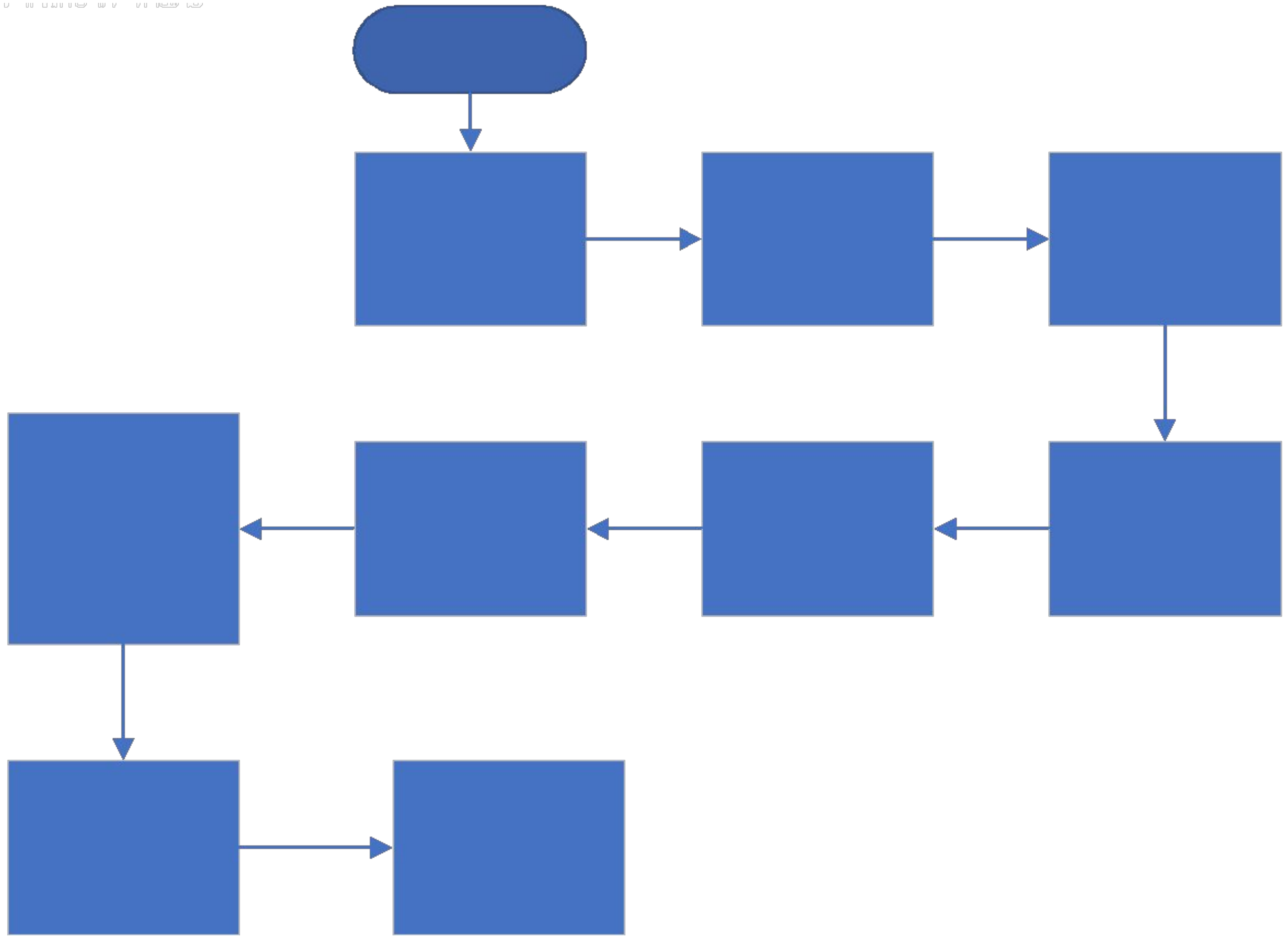
Does it matter?

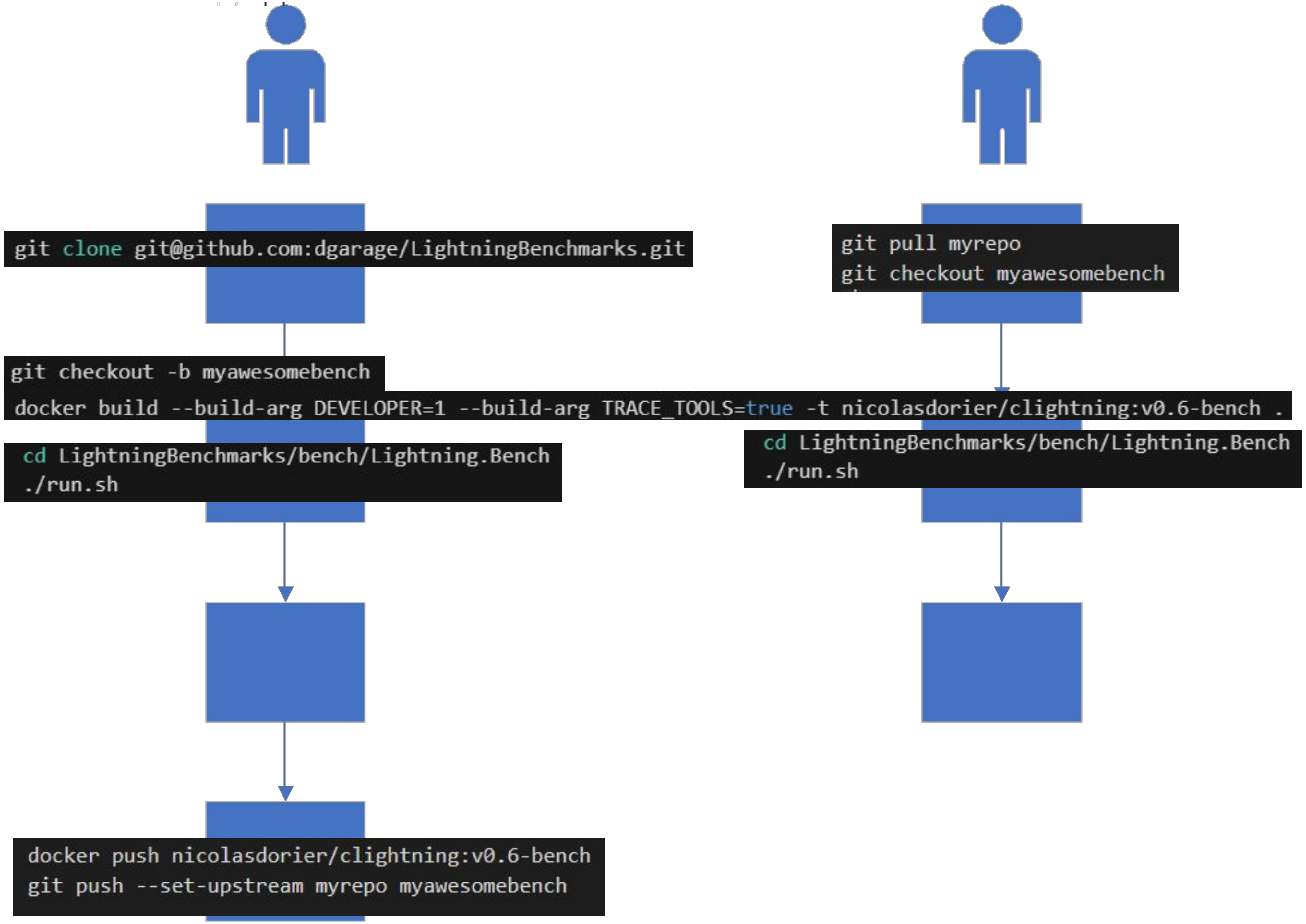
~~No~~ (interesting for developers)

- Lightning network implementers
- Services relying on micro payments

Reframing the objective

Framework Benchmarking and sharing results predictably easily





```
git clone git@github.com:dgarage/LightningBenchmarks.git
```

```
git pull myrepo
git checkout myawesomebench
```

```
git checkout -b myawesomebench
```

```
docker build --build-arg DEVELOPER=1 --build-arg TRACE_TOOLS=true -t nicolasdorier/clightning:v0.6-bench .
```

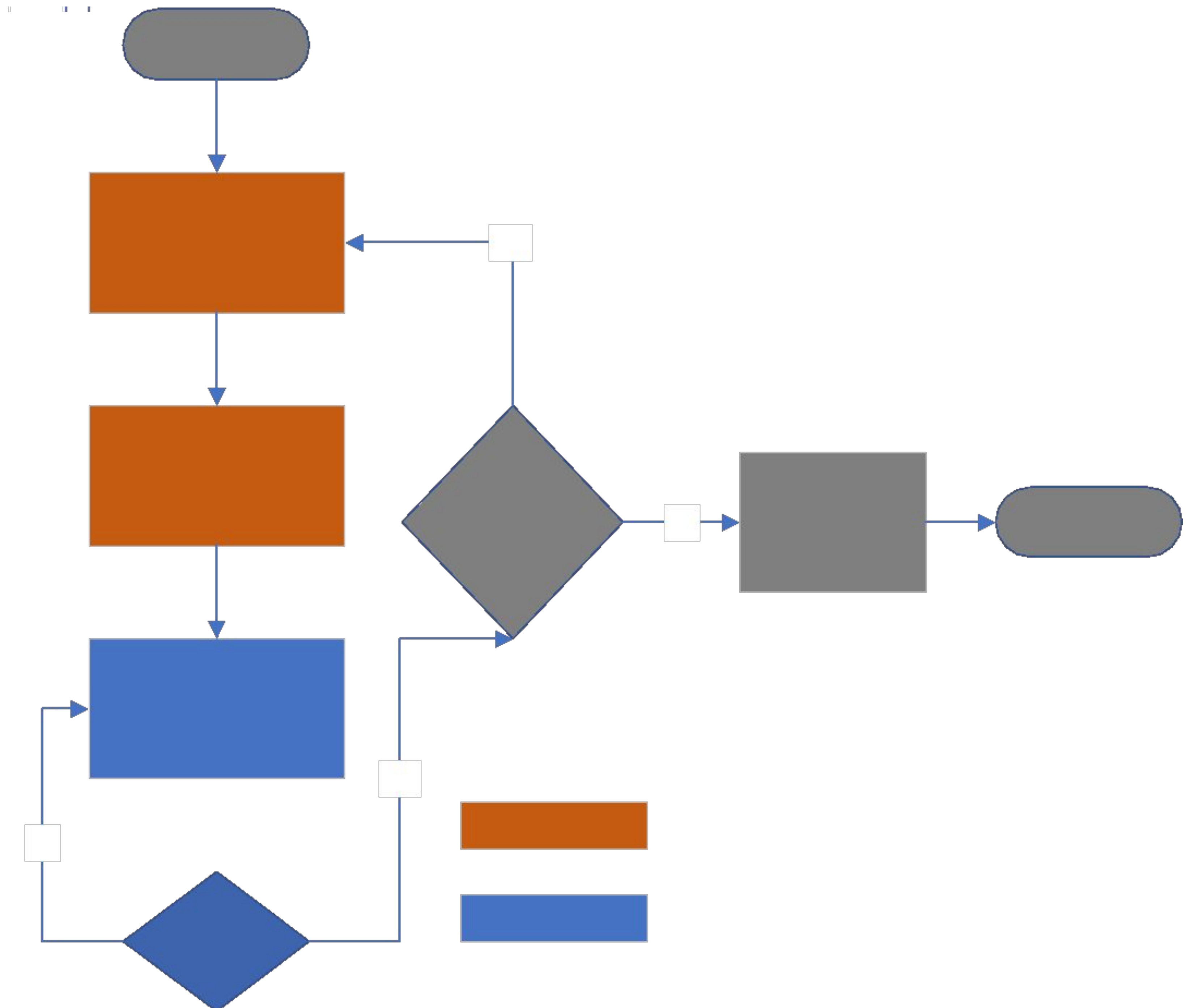
```
cd LightningBenchmarks/bench/Lightning.Bench
./run.sh
```

```
cd LightningBenchmarks/bench/Lightning.Bench
./run.sh
```

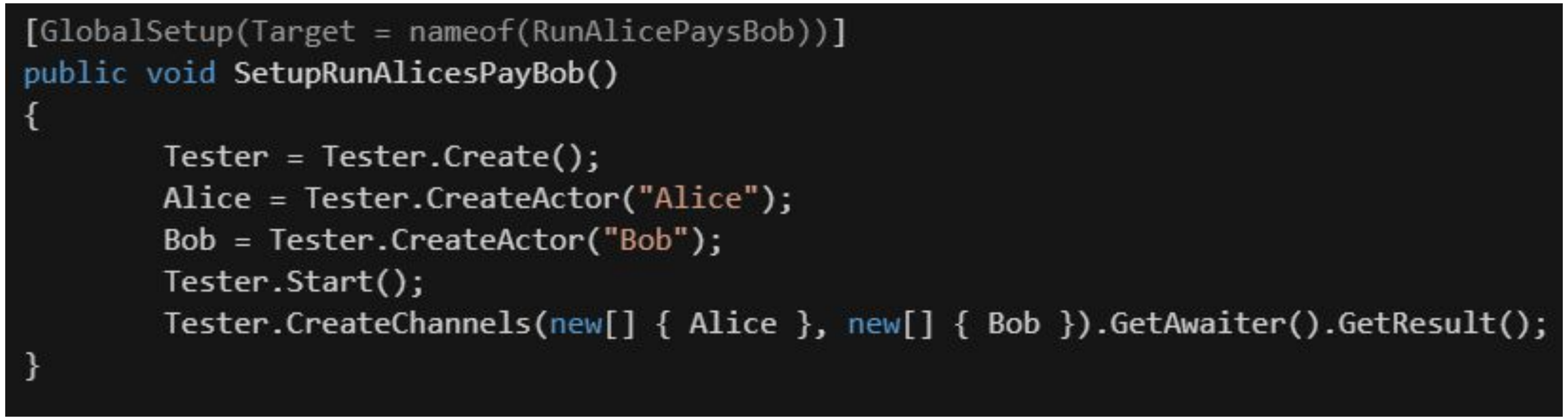
```
docker push nicolasdorier/clightning:v0.6-bench
git push --set-upstream myrepo myawesomebench
```

In the trenches

Run.sh

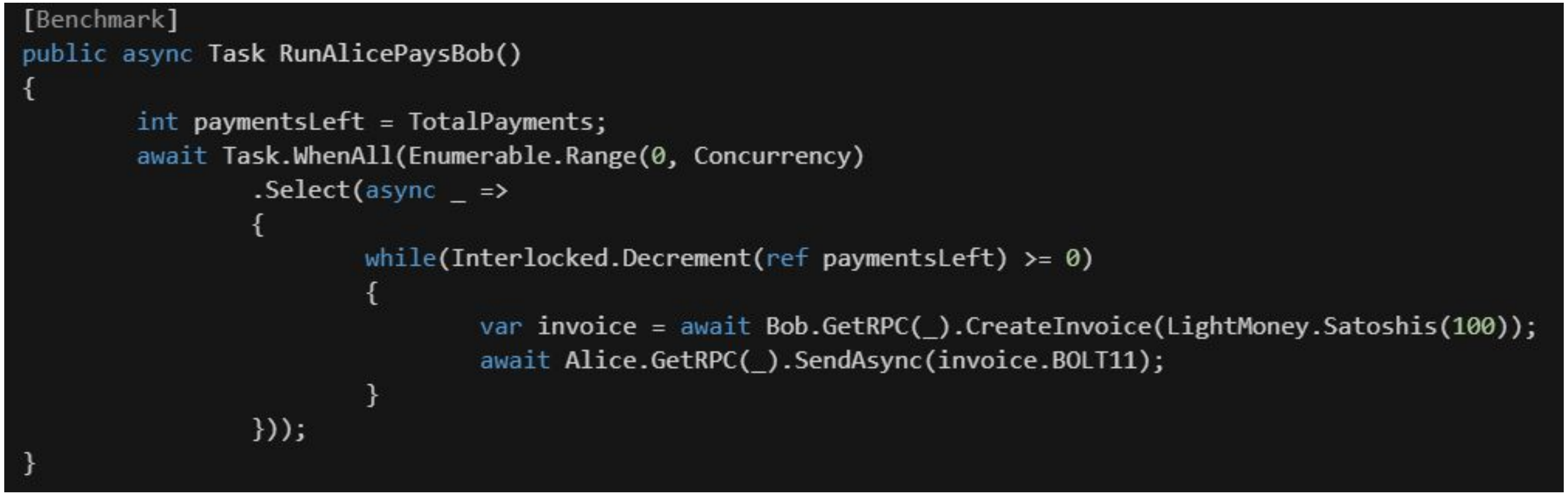


Setup: Alice pays Bob



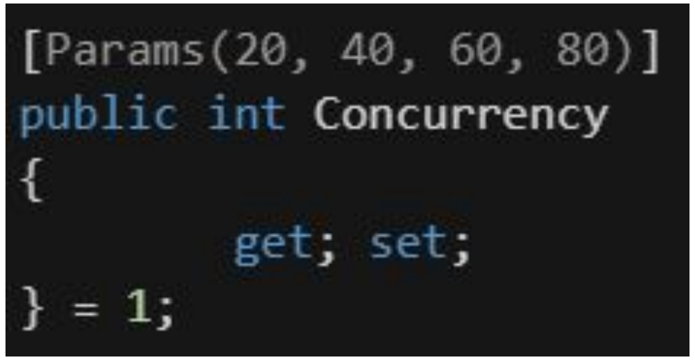
```
[GlobalSetup(Target = nameof(RunAlicePaysBob))]
public void SetupRunAlicesPayBob()
{
        Tester = Tester.Create();
        Alice = Tester.CreateActor("Alice");
        Bob = Tester.CreateActor("Bob");
        Tester.Start();
        Tester.CreateChannels(new[] { Alice }, new[] { Bob }).GetAwaiter().GetResult();
}
```

Execution: Alice pays Bob

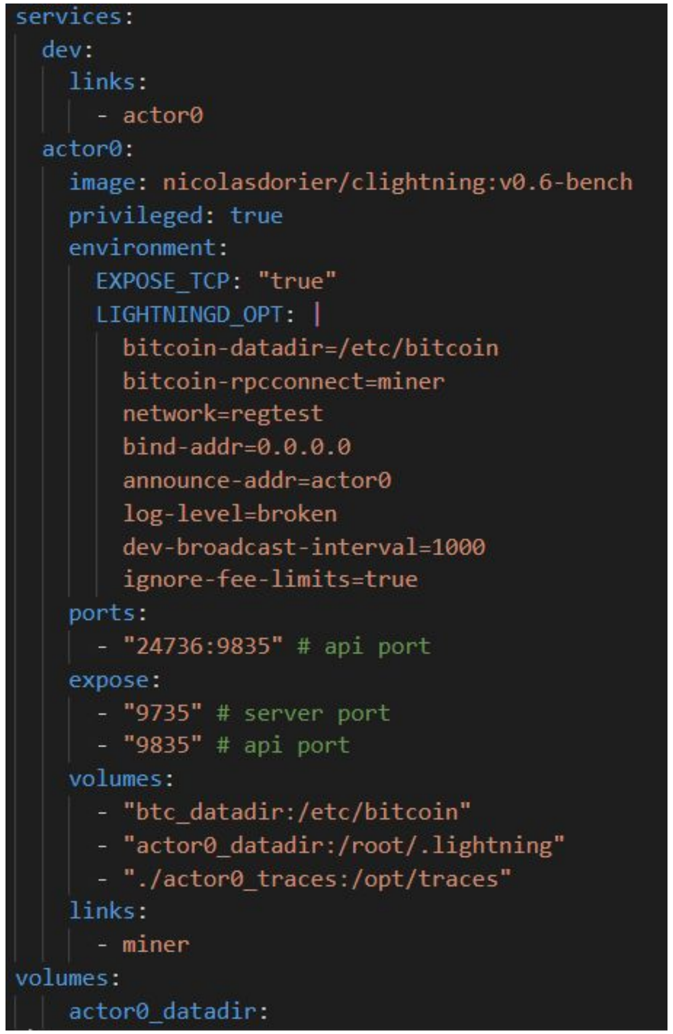


```
[Benchmark]
public async Task RunAlicePaysBob()
{
        int paymentsLeft = TotalPayments;
        await Task.WhenAll(Enumerable.Range(0, Concurrency)
                .Select(async _ =>
                {
                        while(Interlocked.Decrement(ref paymentsLeft) >= 0)
                        {
                                var invoice = await Bob.GetRPC(_).CreateInvoice(LightMoney.Satoshis(100));
                                await Alice.GetRPC(_).SendAsync(invoice.BOLT11);
                        }
                }));
}
```

Specify the concurrency values to test

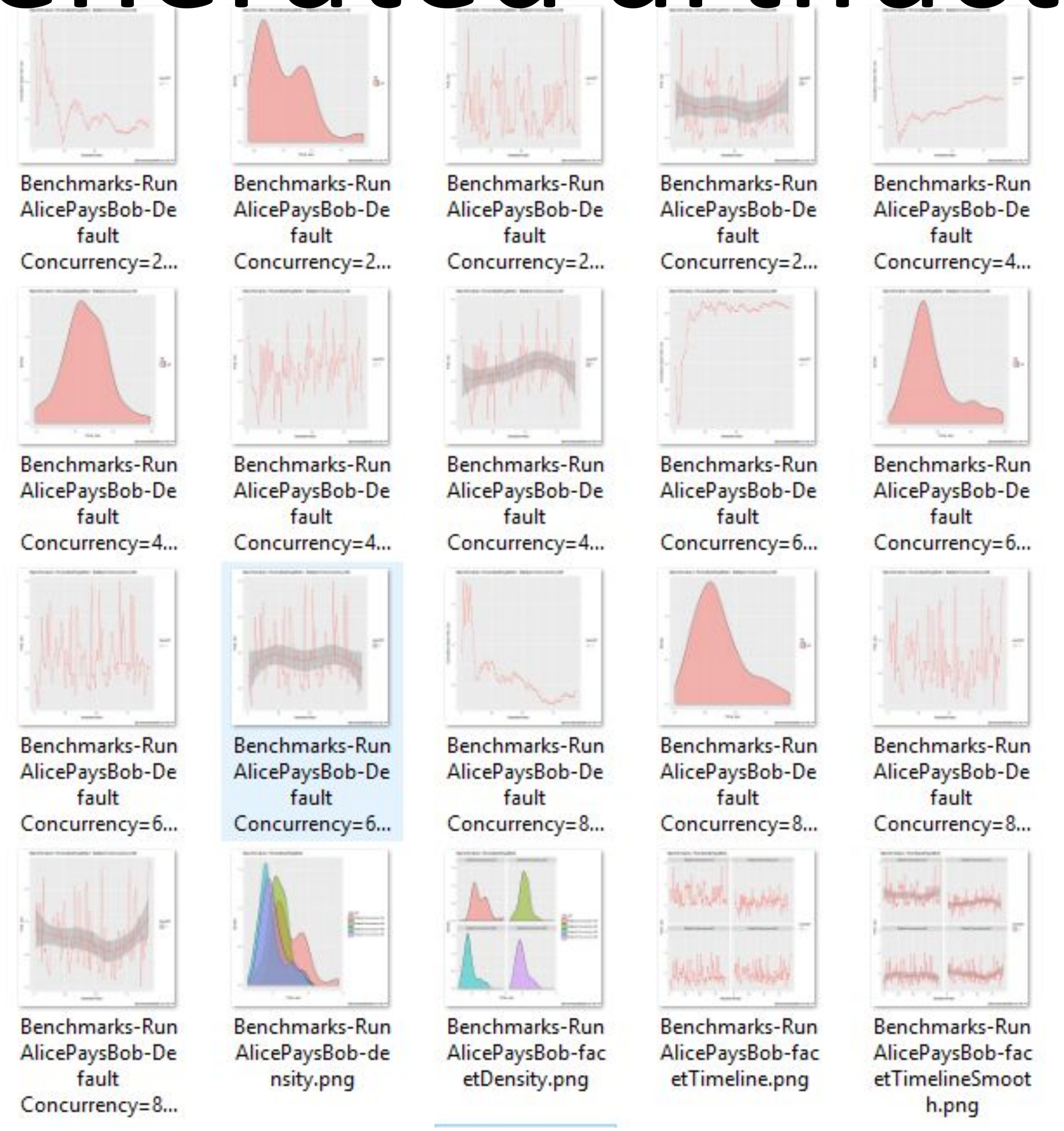

```
[Params(20, 40, 60, 80)]
public int Concurrency
{
        get; set;
} = 1;
```

Template used for actors



```yaml
services:
  dev:
    links:
      - actor0
  actor0:
    image: nicolasdorier/clightning:v0.6-bench
    privileged: true
    environment:
      EXPOSE_TCP: "true"
      LIGHTNINGD_OPT: |
        bitcoin-datadir=/etc/bitcoin
        bitcoin-rpcconnect=miner
        network=regtest
        bind-addr=0.0.0.0
        announce-addr=actor0
        log-level=broken
        dev-broadcast-interval=1000
        ignore-fee-limits=true
    ports:
      - "24736:9835" # api port
    expose:
      - "9735" # server port
      - "9835" # api port
    volumes:
      - "btc_datadir:/etc/bitcoin"
      - "actor0_datadir:/root/.lightning"
      - "./actor0_traces:/opt/traces"
    links:
      - miner
volumes:
    actor0_datadir:
```

Generated artifacts



Benchmarks-Run AlicePaysBob-Default Concurrency=2...

Benchmarks-Run AlicePaysBob-Default Concurrency=2...

Benchmarks-Run AlicePaysBob-Default Concurrency=2...

Benchmarks-Run AlicePaysBob-Default Concurrency=2...

Benchmarks-Run AlicePaysBob-Default Concurrency=4...

Benchmarks-Run AlicePaysBob-Default Concurrency=4...

Benchmarks-Run AlicePaysBob-Default Concurrency=4...

Benchmarks-Run AlicePaysBob-Default Concurrency=4...

Benchmarks-Run AlicePaysBob-Default Concurrency=6...

Benchmarks-Run AlicePaysBob-Default Concurrency=6...

Benchmarks-Run AlicePaysBob-Default Concurrency=6...

Benchmarks-Run AlicePaysBob-Default Concurrency=6...

Benchmarks-Run AlicePaysBob-Default Concurrency=8...

Benchmarks-Run AlicePaysBob-Default Concurrency=8...

Benchmarks-Run AlicePaysBob-Default Concurrency=8...

Benchmarks-Run AlicePaysBob-Default Concurrency=8...

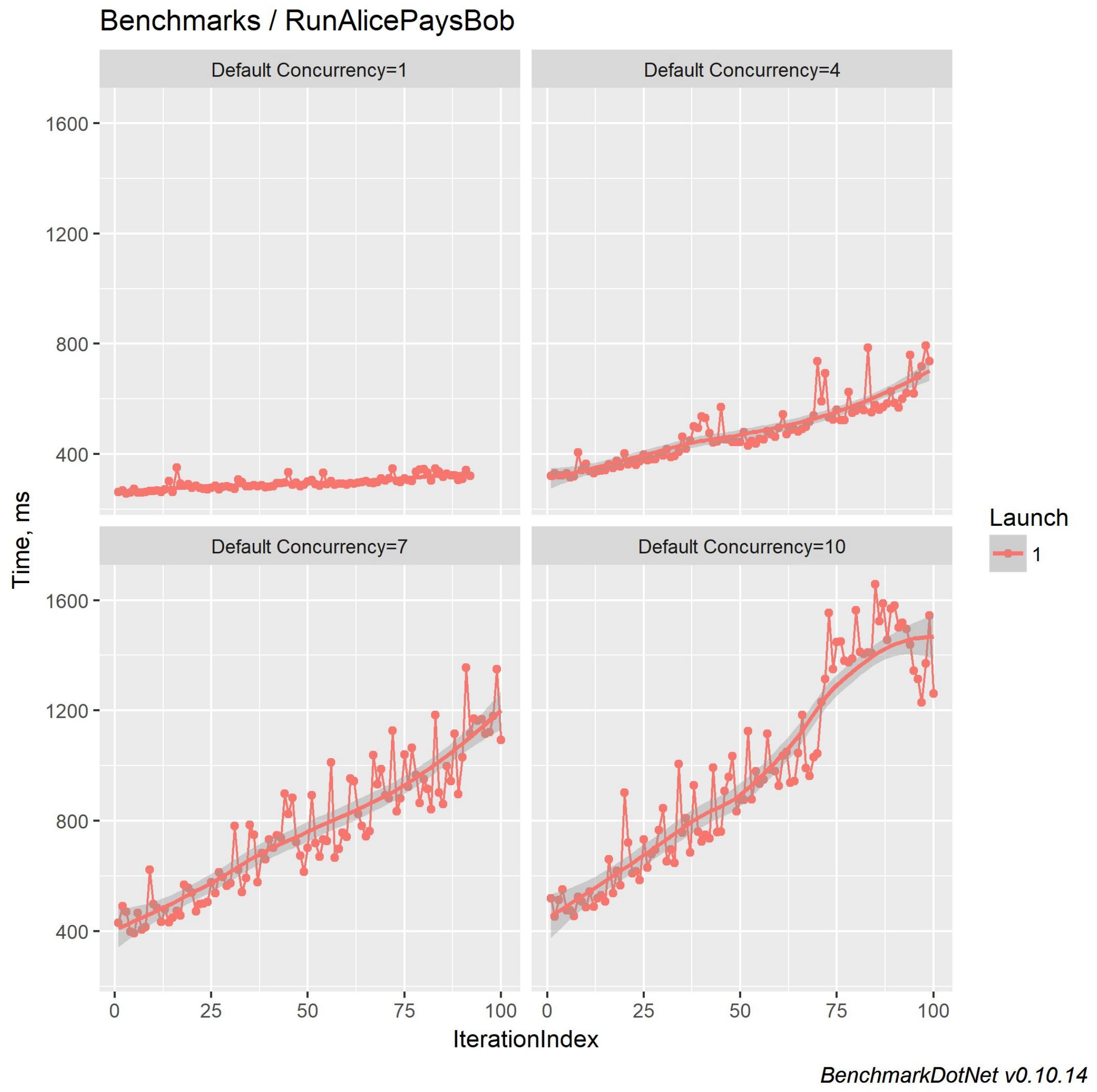
Benchmarks-Run AlicePaysBob-density.png

Benchmarks-Run AlicePaysBob-facetDensity.png

Benchmarks-Run AlicePaysBob-facetTimeline.png

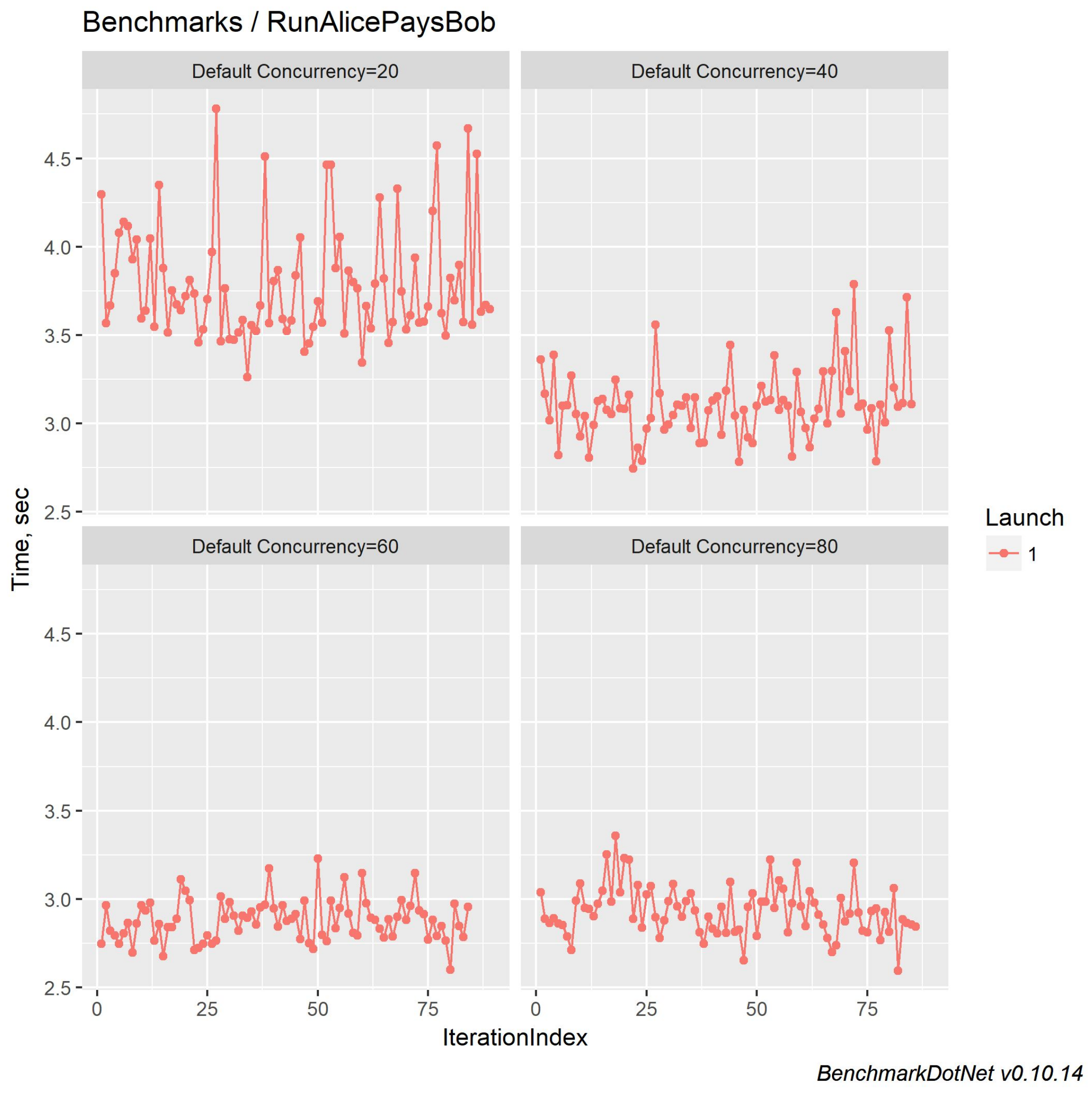
Benchmarks-Run AlicePaysBob-facetTimelineSmooth.png

A bug

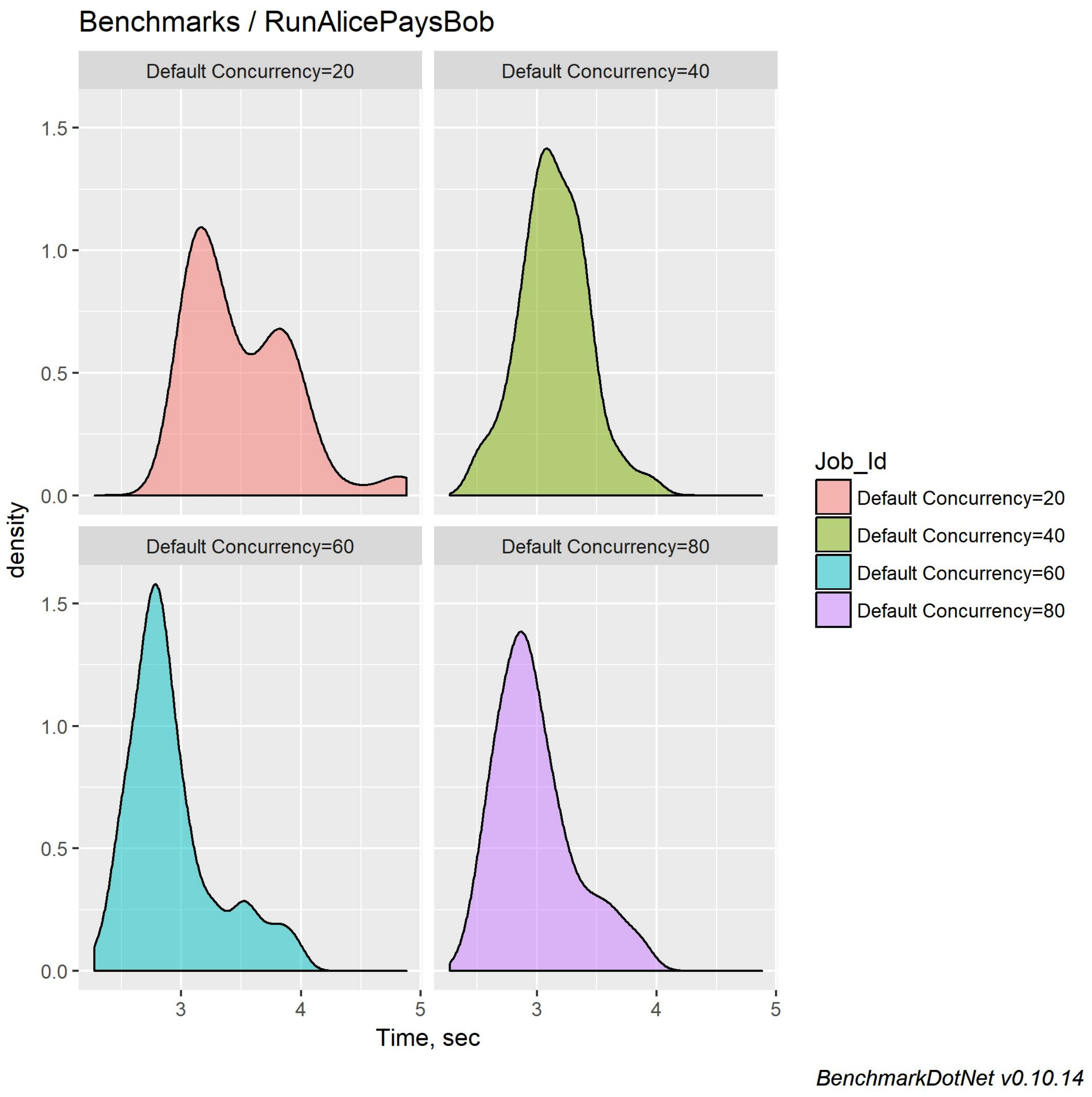


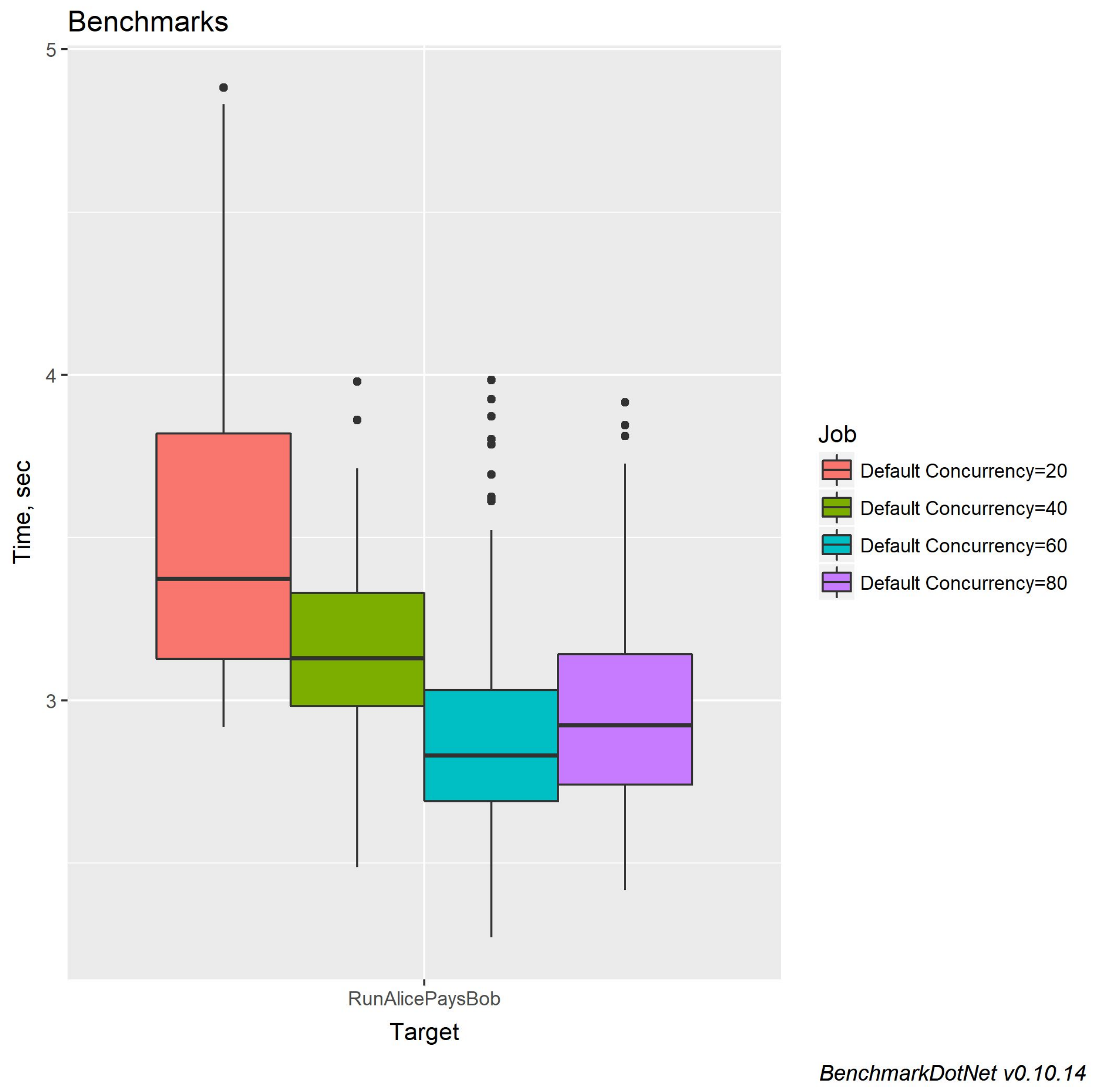
Benchmarks / RunAlicePaysBob
Default Concurrency=1
Default Concurrency=4
Default Concurrency=7
Default Concurrency=10
Time, ms
IterationIndex
Launch
1
400
800
1200
1600
0
25
50
75
100
BenchmarkDotNet v0.10.14

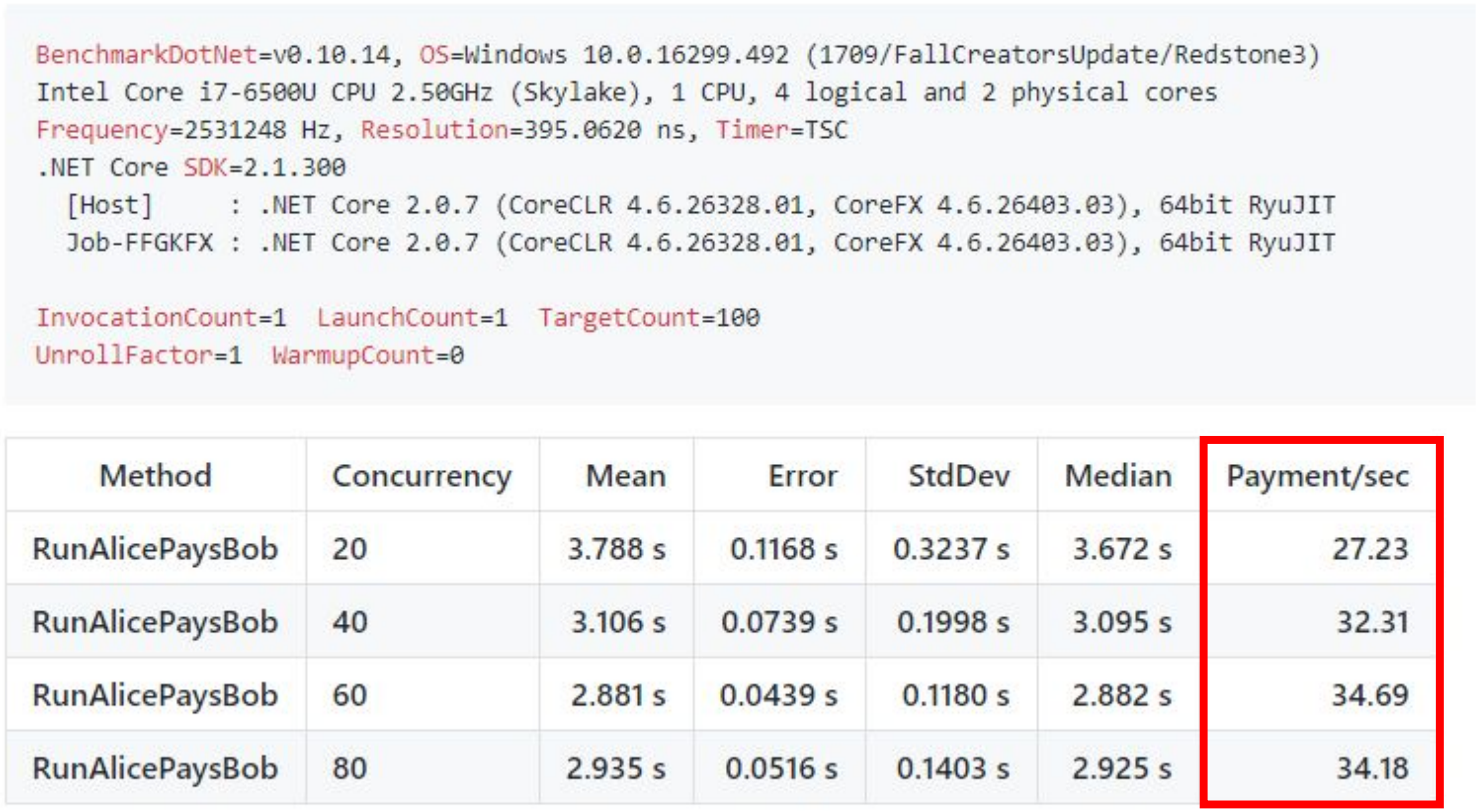
No bug



Benchmarks / RunAlicePaysBob
Default Concurrency=20
Default Concurrency=40
Default Concurrency=60
Default Concurrency=80
Time, sec
4.5
4.0
3.5
3.0
2.5
0
25
50
75
IterationIndex
Launch
1
BenchmarkDotNet v0.10.14


Benchmarks / RunAlicePaysBob
Default Concurrency=20
Default Concurrency=40
Default Concurrency=60
Default Concurrency=80
density
0.0
0.5
1.0
1.5
3
4
5
Time, sec
Job_Id
Default Concurrency=20
Default Concurrency=40
Default Concurrency=60
Default Concurrency=80
BenchmarkDotNet v0.10.14

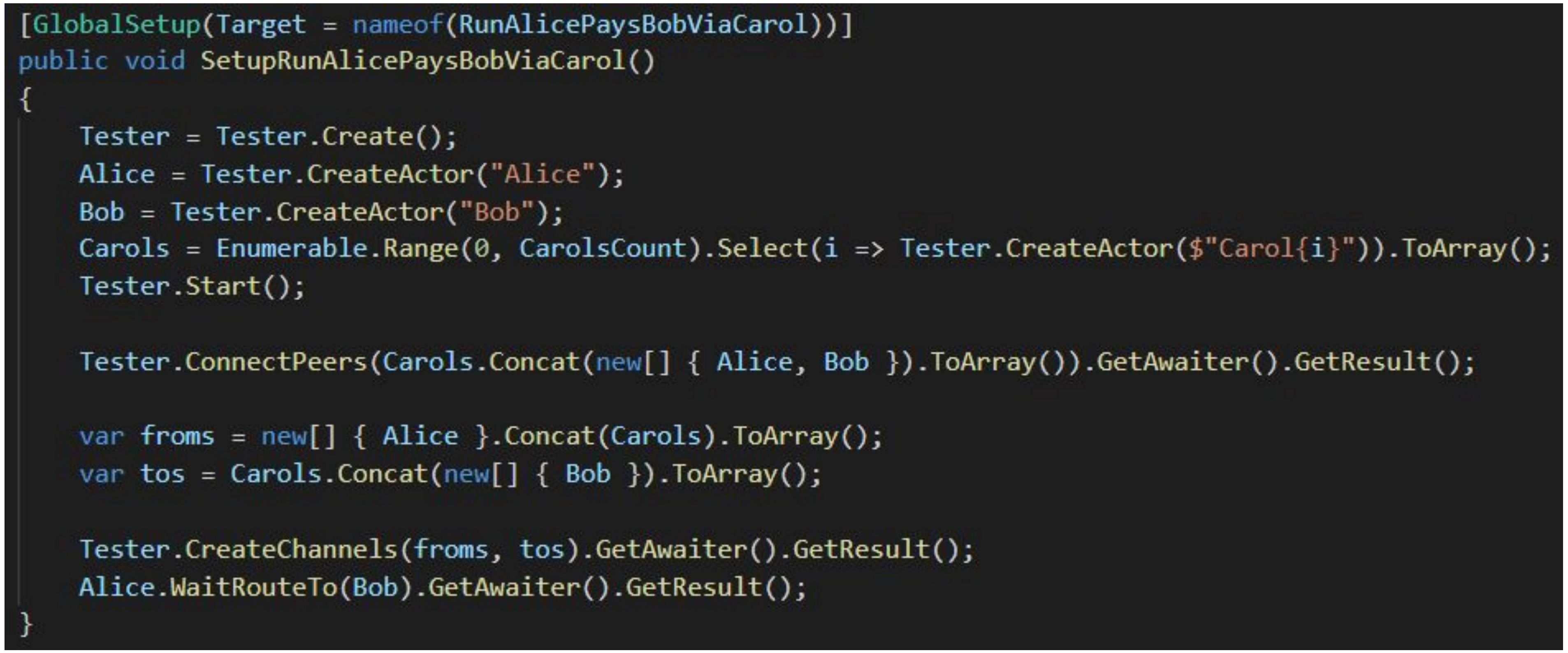

Benchmarks
Time, sec
3
4
5
RunAlicePaysBob
Target
Job
Default Concurrency=20
Default Concurrency=40
Default Concurrency=60
Default Concurrency=80
BenchmarkDotNet v0.10.14



```
BenchmarkDotNet=v0.10.14, OS=Windows 10.0.16299.492 (1709/FallCreatorsUpdate/Redstone3)
Intel Core i7-6500U CPU 2.50GHz (Skylake), 1 CPU, 4 logical and 2 physical cores
Frequency=2531248 Hz, Resolution=395.0620 ns, Timer=TSC
.NET Core SDK=2.1.300
  [Host]     : .NET Core 2.0.7 (CoreCLR 4.6.26328.01, CoreFX 4.6.26403.03), 64bit RyuJIT
  Job-FFGKFX : .NET Core 2.0.7 (CoreCLR 4.6.26328.01, CoreFX 4.6.26403.03), 64bit RyuJIT

InvocationCount=1  LaunchCount=1  TargetCount=100
UnrollFactor=1  WarmupCount=0
```

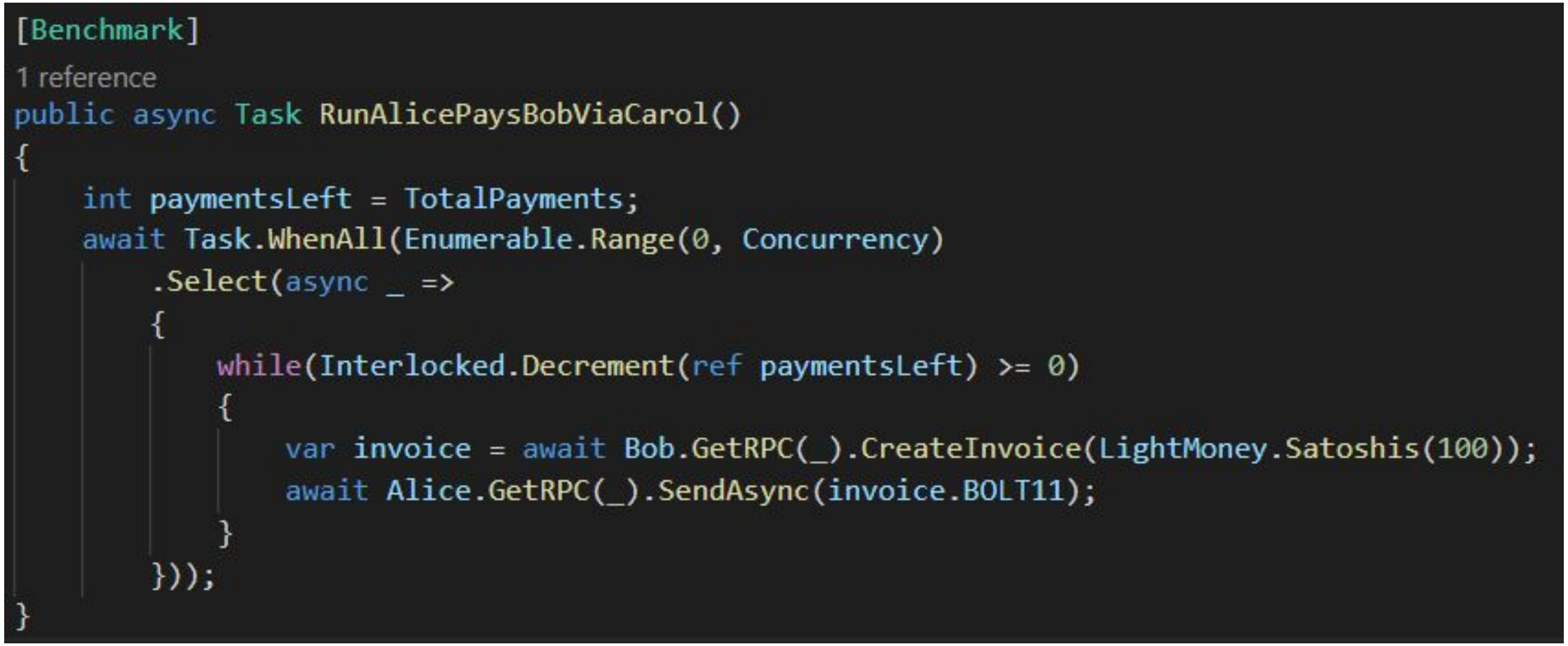
Method	Concurrency	Mean	Error	StdDev	Median	Payment/sec
RunAlicePaysBob	20	3.788 s	0.1168 s	0.3237 s	3.672 s	27.23
RunAlicePaysBob	40	3.106 s	0.0739 s	0.1998 s	3.095 s	32.31
RunAlicePaysBob	60	2.881 s	0.0439 s	0.1180 s	2.882 s	34.69
RunAlicePaysBob	80	2.935 s	0.0516 s	0.1403 s	2.925 s	34.18

Setup: Alice pays Bob via Carol



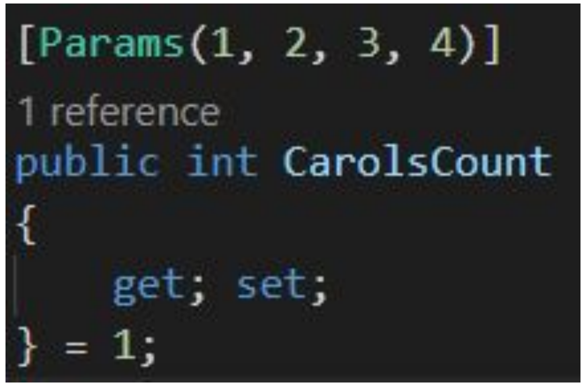
```
[GlobalSetup(Target = nameof(RunAlicePaysBobViaCarol))]
public void SetupRunAlicePaysBobViaCarol()
{
    Tester = Tester.Create();
    Alice = Tester.CreateActor("Alice");
    Bob = Tester.CreateActor("Bob");
    Carols = Enumerable.Range(0, CarolsCount).Select(i => Tester.CreateActor($"Carol{i}")).ToArray();
    Tester.Start();

    Tester.ConnectPeers(Carols.Concat(new[] { Alice, Bob }).ToArray()).GetAwaiter().GetResult();

    var froms = new[] { Alice }.Concat(Carols).ToArray();
    var tos = Carols.Concat(new[] { Bob }).ToArray();

    Tester.CreateChannels(froms, tos).GetAwaiter().GetResult();
    Alice.WaitRouteTo(Bob).GetAwaiter().GetResult();
}
```

Execution: Alice pays Bob via Carol

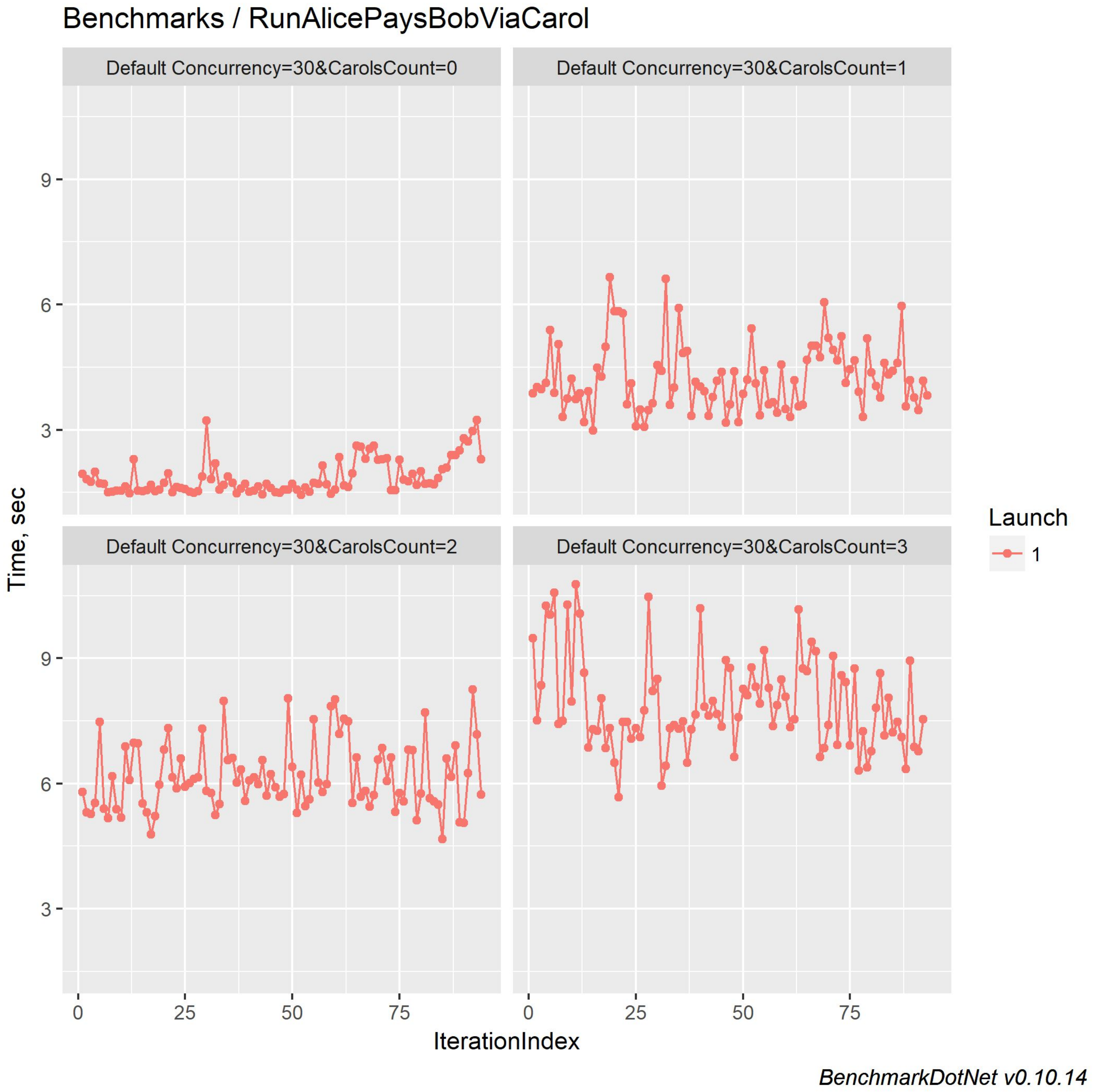


```
[Benchmark]
1 reference
public async Task RunAlicePaysBobViaCarol()
{
    int paymentsLeft = TotalPayments;
    await Task.WhenAll(Enumerable.Range(0, Concurrency)
        .Select(async _ =>
        {
            while(Interlocked.Decrement(ref paymentsLeft) >= 0)
            {
                var invoice = await Bob.GetRPC(_).CreateInvoice(LightMoney.Satoshis(100));
                await Alice.GetRPC(_).SendAsync(invoice.BOLT11);
            }
        }));
}
```

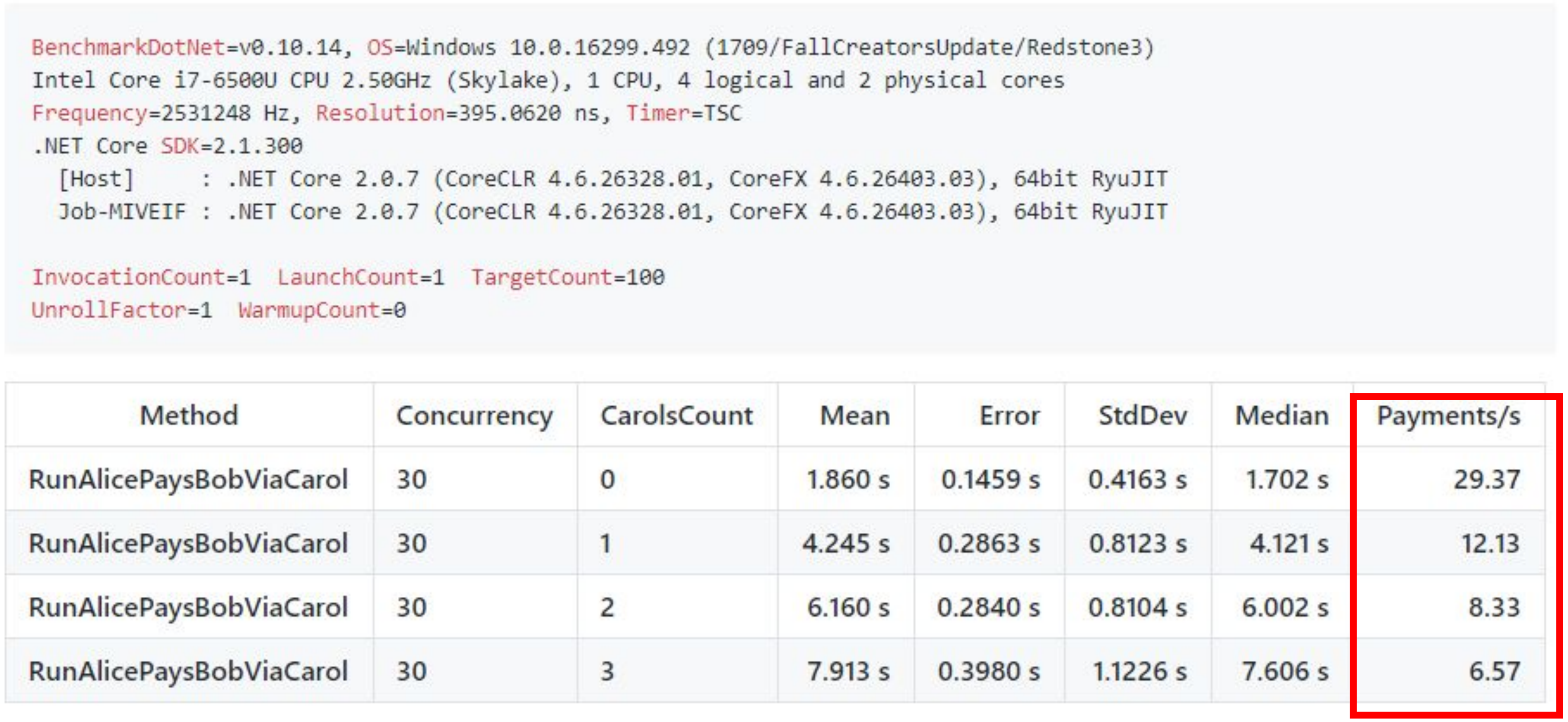
Vary the number of Carols



```
[Params(1, 2, 3, 4)]
1 reference
public int CarolsCount
{
    get; set;
} = 1;
```



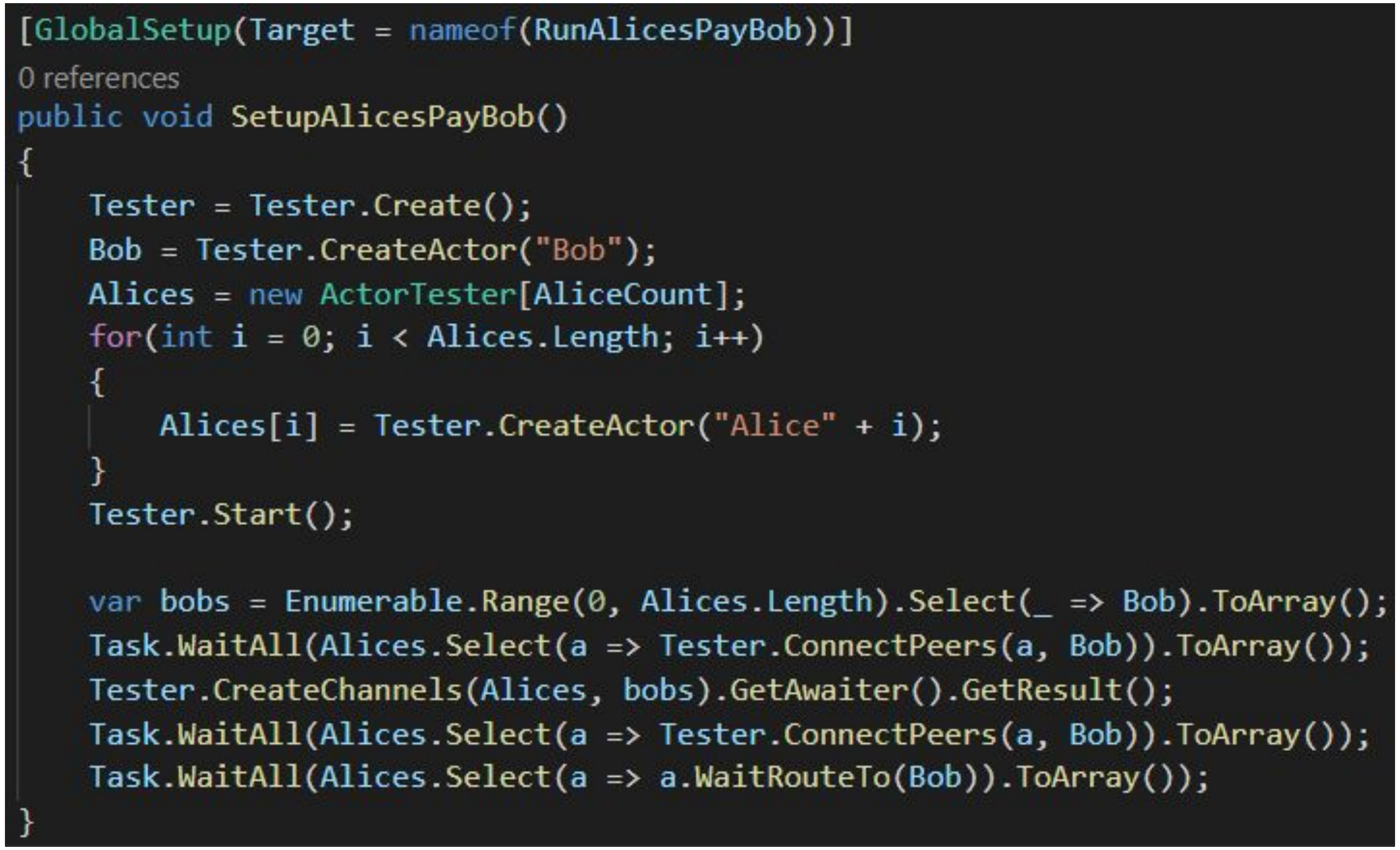
Benchmarks / RunAlicePaysBobViaCarol
Default Concurrency=30&CarolsCount=0
Default Concurrency=30&CarolsCount=1
Default Concurrency=30&CarolsCount=2
Default Concurrency=30&CarolsCount=3
Time, sec
3
6
9
0
25
50
75
IterationIndex
Launch
1
BenchmarkDotNet v0.10.14



```
BenchmarkDotNet=v0.10.14, OS=Windows 10.0.16299.492 (1709/FallCreatorsUpdate/Redstone3)
Intel Core i7-6500U CPU 2.50GHz (Skylake), 1 CPU, 4 logical and 2 physical cores
Frequency=2531248 Hz, Resolution=395.0620 ns, Timer=TSC
.NET Core SDK=2.1.300
  [Host]     : .NET Core 2.0.7 (CoreCLR 4.6.26328.01, CoreFX 4.6.26403.03), 64bit RyuJIT
  Job-MIVEIF : .NET Core 2.0.7 (CoreCLR 4.6.26328.01, CoreFX 4.6.26403.03), 64bit RyuJIT

InvocationCount=1  LaunchCount=1  TargetCount=100
UnrollFactor=1  WarmupCount=0
```

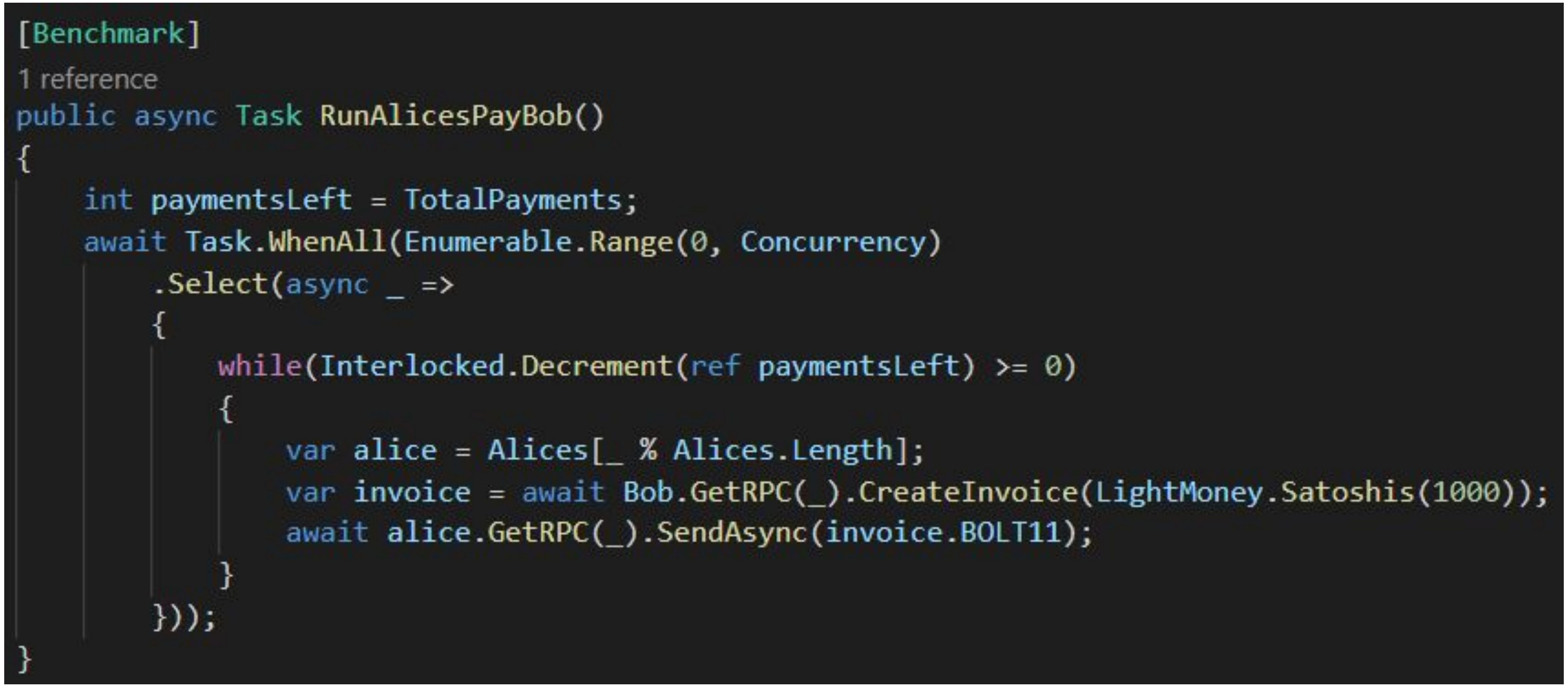
Method	Concurrency	CarolsCount	Mean	Error	StdDev	Median	Payments/s
RunAlicePaysBobViaCarol	30	0	1.860 s	0.1459 s	0.4163 s	1.702 s	29.37
RunAlicePaysBobViaCarol	30	1	4.245 s	0.2863 s	0.8123 s	4.121 s	12.13
RunAlicePaysBobViaCarol	30	2	6.160 s	0.2840 s	0.8104 s	6.002 s	8.33
RunAlicePaysBobViaCarol	30	3	7.913 s	0.3980 s	1.1226 s	7.606 s	6.57

Setup: Alices pay Bob



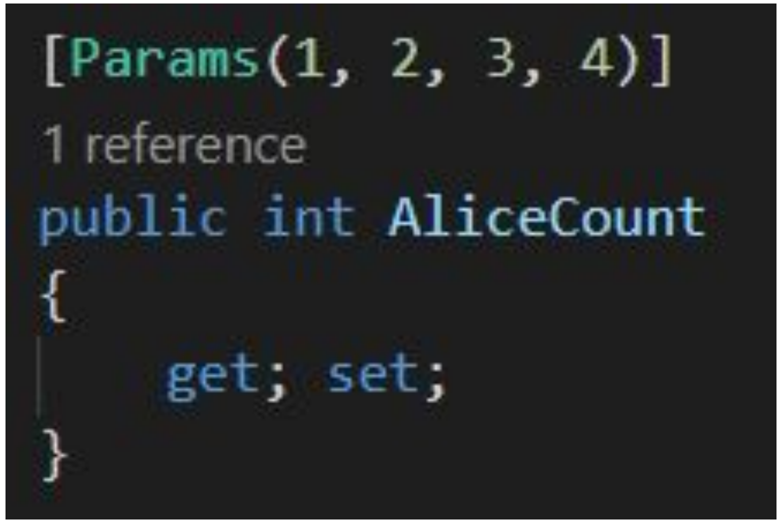
```
[GlobalSetup(Target = nameof(RunAlicesPayBob))]
0 references
public void SetupAlicesPayBob()
{
    Tester = Tester.Create();
    Bob = Tester.CreateActor("Bob");
    Alices = new ActorTester[AliceCount];
    for(int i = 0; i < Alices.Length; i++)
    {
        Alices[i] = Tester.CreateActor("Alice" + i);
    }
    Tester.Start();

    var bobs = Enumerable.Range(0, Alices.Length).Select(_ => Bob).ToArray();
    Task.WaitAll(Alices.Select(a => Tester.ConnectPeers(a, Bob)).ToArray());
    Tester.CreateChannels(Alices, bobs).GetAwaiter().GetResult();
    Task.WaitAll(Alices.Select(a => Tester.ConnectPeers(a, Bob)).ToArray());
    Task.WaitAll(Alices.Select(a => a.WaitRouteTo(Bob)).ToArray());
}
```

Execution: Alices pay Bob

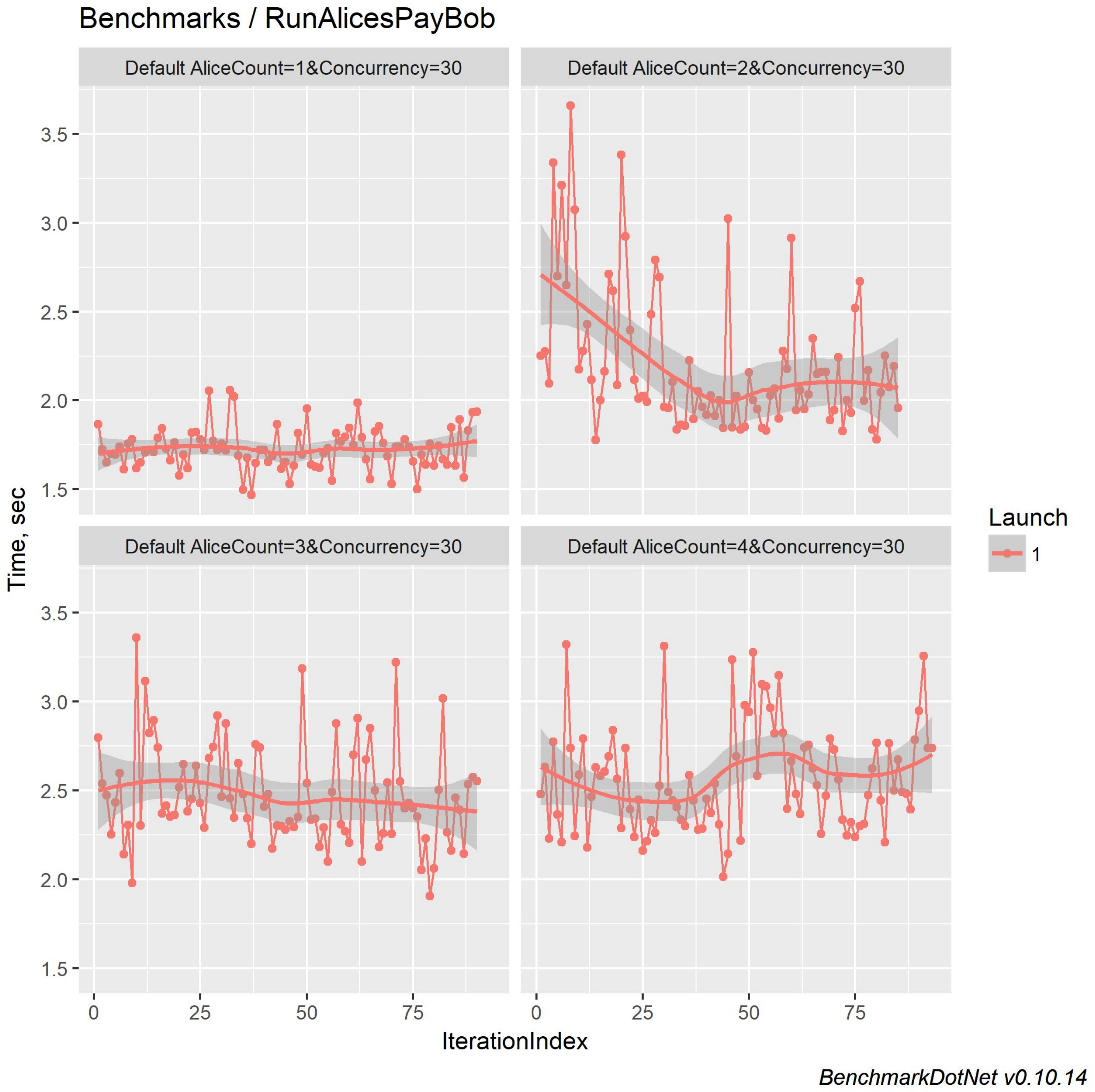


```
[Benchmark]
1 reference
public async Task RunAlicesPayBob()
{
    int paymentsLeft = TotalPayments;
    await Task.WhenAll(Enumerable.Range(0, Concurrency)
        .Select(async _ =>
        {
            while(Interlocked.Decrement(ref paymentsLeft) >= 0)
            {
                var alice = Alices[_ % Alices.Length];
                var invoice = await Bob.GetRPC(_).CreateInvoice(LightMoney.Satoshis(1000));
                await alice.GetRPC(_).SendAsync(invoice.BOLT11);
            }
        }));
}
```

Vary the number of Alices

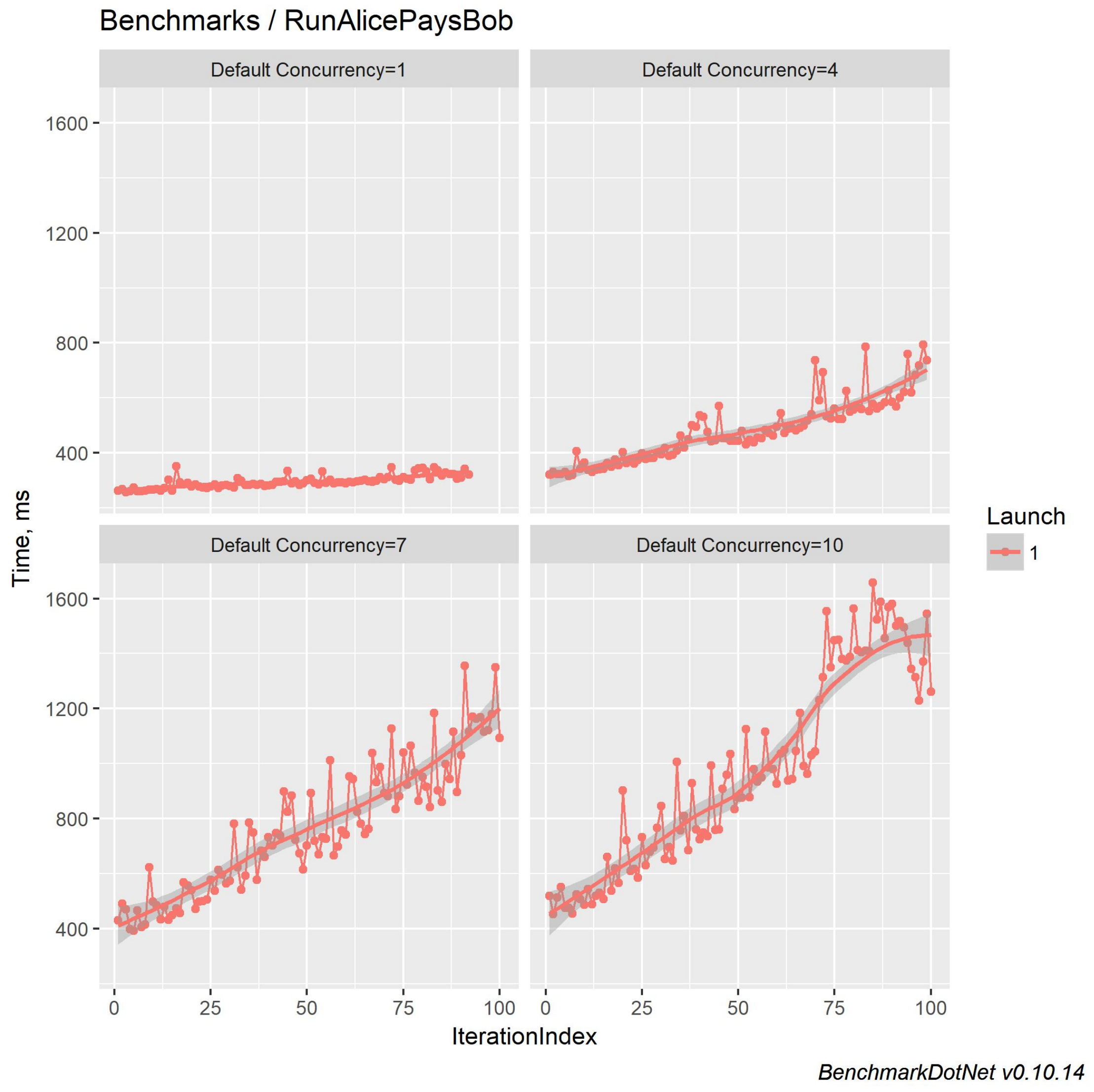


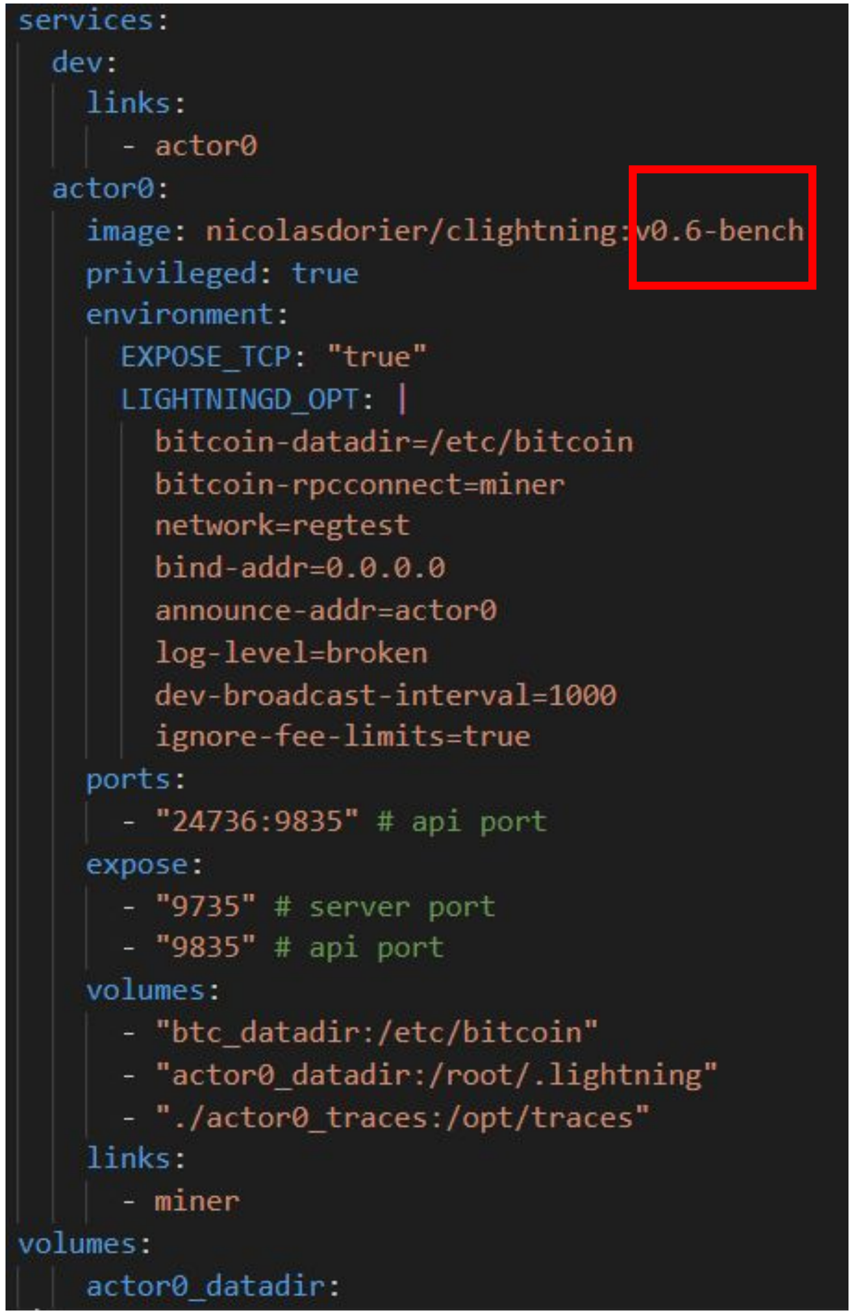
```
[Params(1, 2, 3, 4)]
1 reference
public int AliceCount
{
    get; set;
}
```


Benchmarks / RunAlicesPayBob
Default AliceCount=1&Concurrency=30
Default AliceCount=2&Concurrency=30
Default AliceCount=3&Concurrency=30
Default AliceCount=4&Concurrency=30
Time, sec
3.5
3.0
2.5
2.0
1.5
0
25
50
75
IterationIndex
Launch
1
BenchmarkDotNet v0.10.14

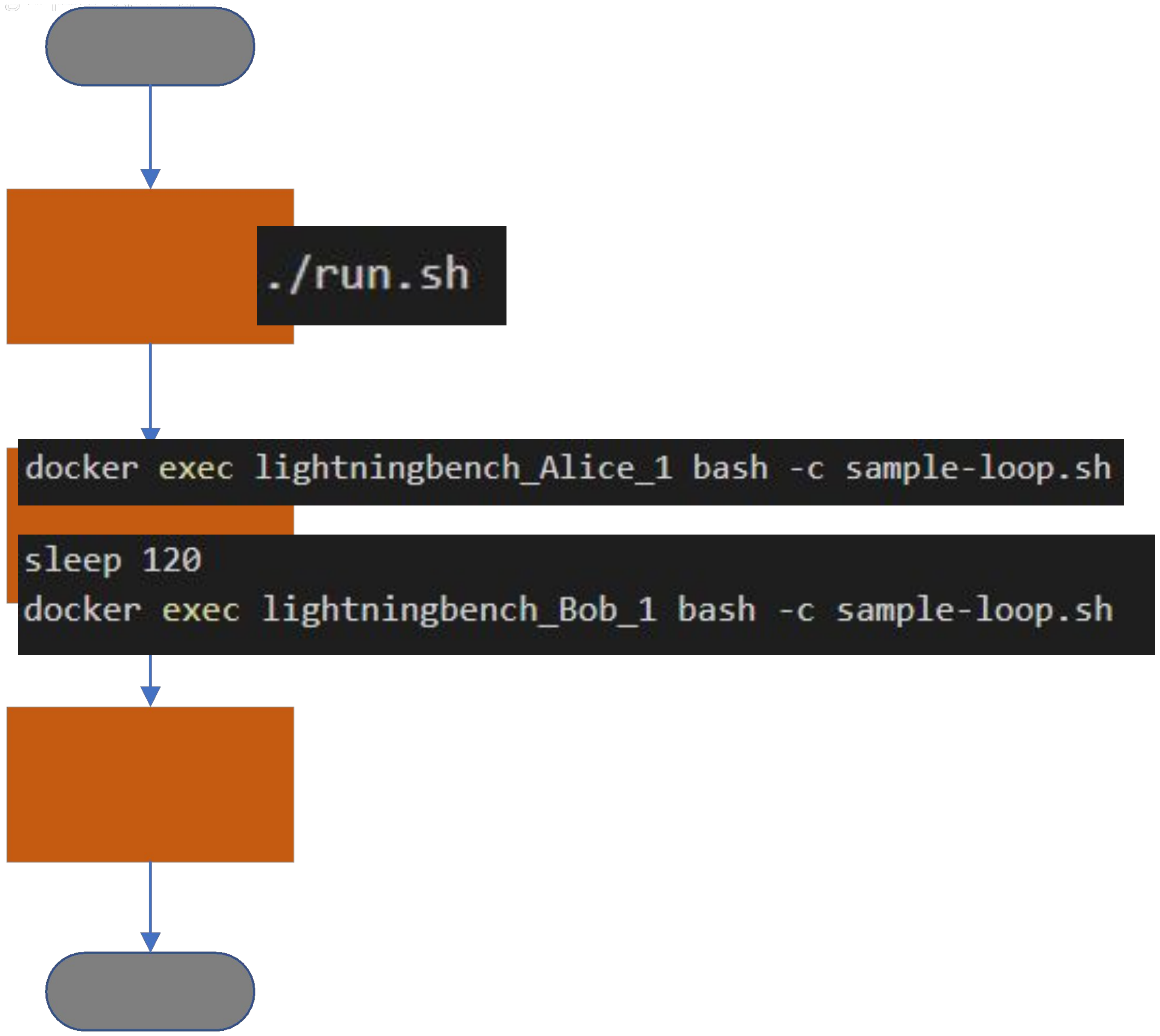
Instrumentation

Revisiting the bug





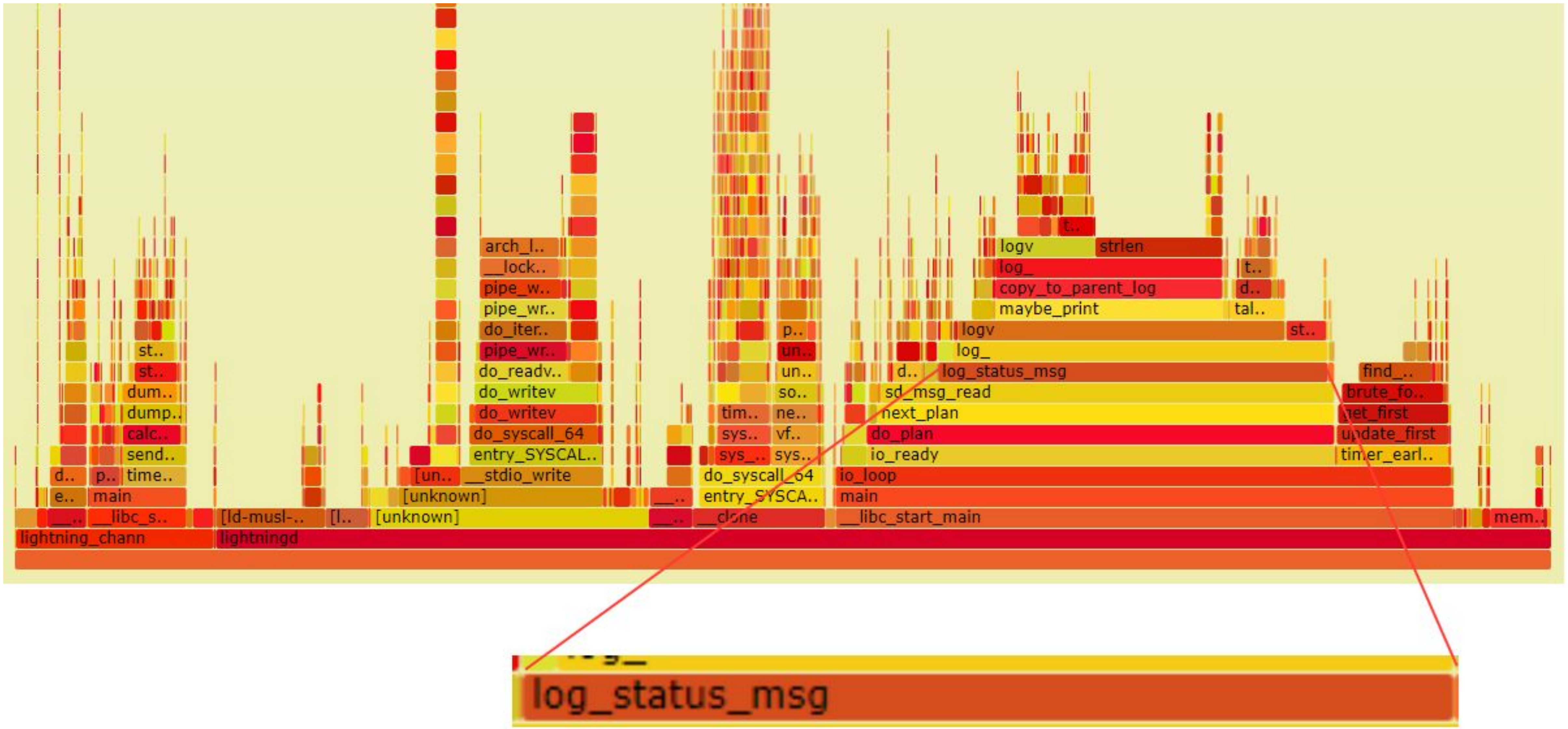
```
services:
  dev:
    links:
      - actor0
  actor0:
    image: nicolasdorier/clightning:v0.6-bench
    privileged: true
    environment:
      EXPOSE_TCP: "true"
      LIGHTNINGD_OPT: |
        bitcoin-datadir=/etc/bitcoin
        bitcoin-rpcconnect=miner
        network=regtest
        bind-addr=0.0.0.0
        announce-addr=actor0
        log-level=broken
        dev-broadcast-interval=1000
        ignore-fee-limits=true
    ports:
      - "24736:9835" # api port
    expose:
      - "9735" # server port
      - "9835" # api port
    volumes:
      - "btc_datadir:/etc/bitcoin"
      - "actor0_datadir:/root/.lightning"
      - "./actor0_traces:/opt/traces"
    links:
      - miner
volumes:
  actor0_datadir:
```

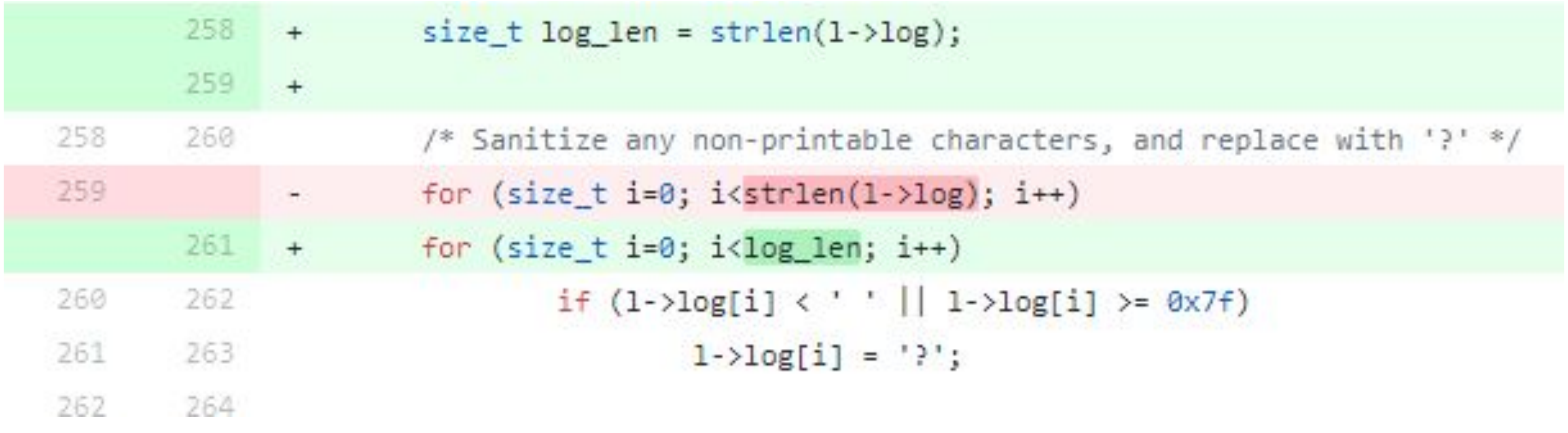


```
./run.sh
```

```
docker exec lightningbench_Alice_1 bash -c sample-loop.sh
```

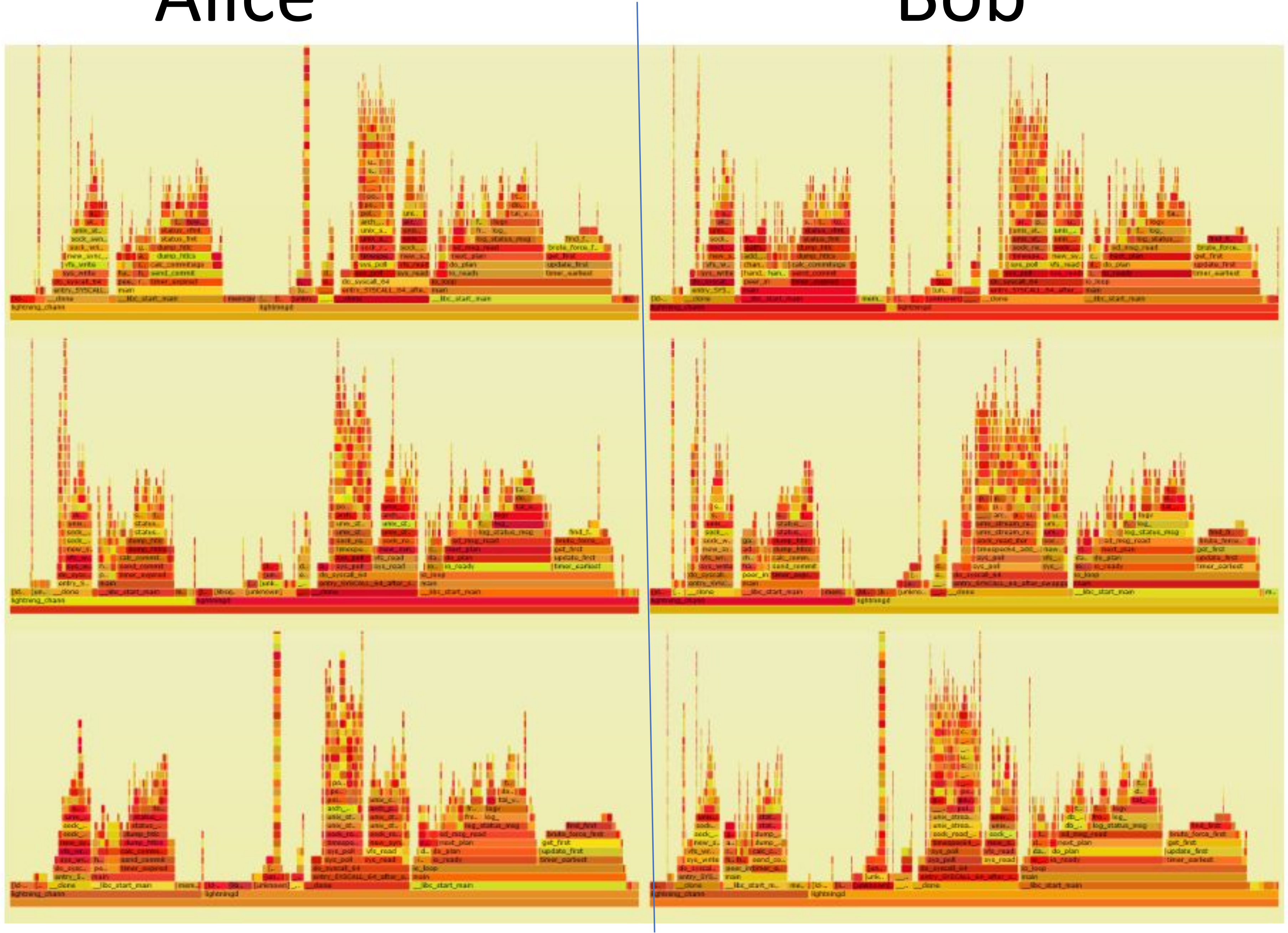
```
sleep 120
docker exec lightningbench_Bob_1 bash -c sample-loop.sh
```

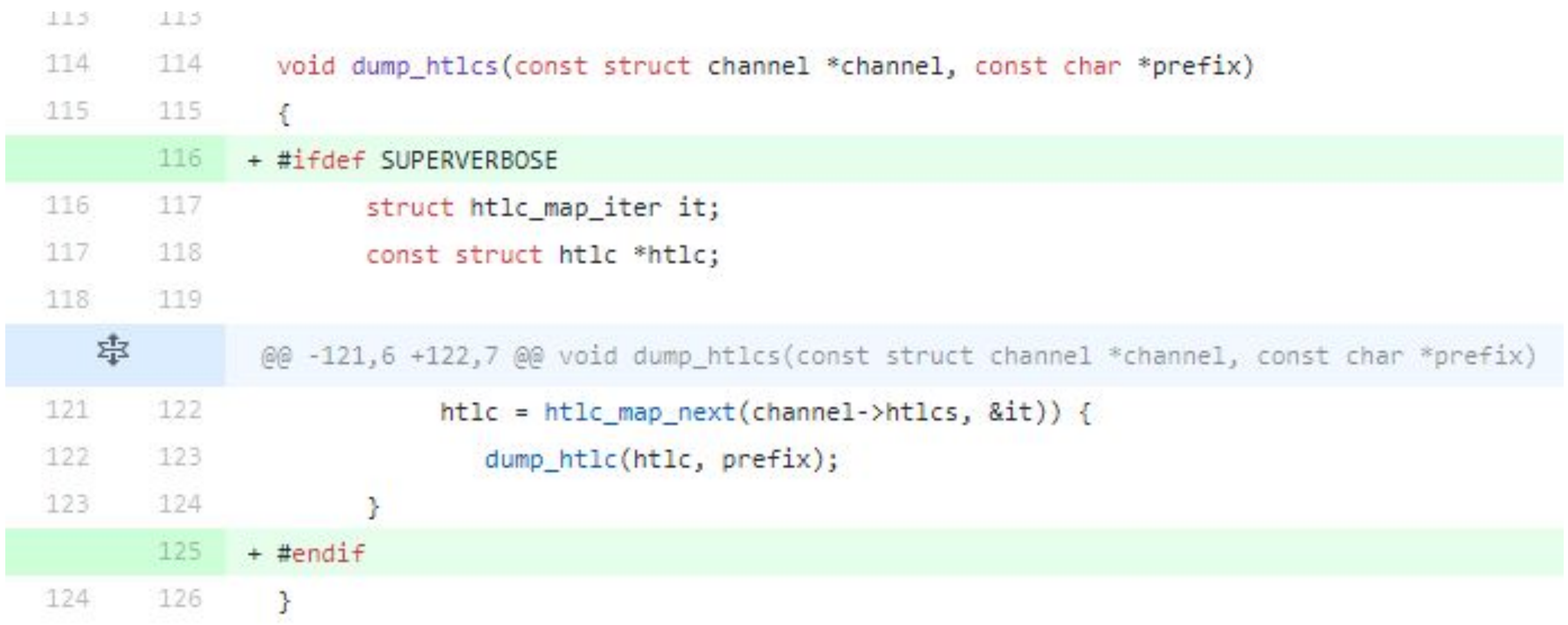

arch_l..
__lock..
pipe_w..
pipe_wr..
do_iter..
pipe_wr..
do_readv..
do_writev
do_writev
do_syscall_64
entry_SYSCAL..
[un..
__stdio_write
[unknown]
[unknown]
[ld-musl-..
[l..
lightningd
lightning_chann
st..
st..
dum..
dump..
calc..
send..
d..
p..
time..
e..
main
__..
__libc_s..
tim..
sys..
sys_..
do_syscall_64
entry_SYSCA..
__..
__..
__clone
p..
un..
un..
so..
ne..
vf..
sys..
logv
strlen
log_
copy_to_parent_log
maybe_print
logv
log_
log_status_msg
sd_msg_read
next_plan
do_plan
io_ready
io_loop
main
__libc_start_main
t..
t..
d..
tal..
st..
d..
find_..
brute_fo..
get_first
update_first
timer_earl..
mem..
log_status_msg



```diff
+       size_t log_len = strlen(l->log);
+
        /* Sanitize any non-printable characters, and replace with '?' */
-       for (size_t i=0; i<strlen(l->log); i++)
+       for (size_t i=0; i<log_len; i++)
                if (l->log[i] < ' ' || l->log[i] >= 0x7f)
                        l->log[i] = '?';

```

Alice Bob





```
 void dump_htlcs(const struct channel *channel, const char *prefix)
 {
+#ifdef SUPERVERBOSE
        struct htlc_map_iter it;
        const struct htlc *htlc;

@@ -121,6 +122,7 @@ void dump_htlcs(const struct channel *channel, const char *prefix)
             htlc = htlc_map_next(channel->htlcs, &it)) {
                dump_htlc(htlc, prefix);
        }
+#endif
 }
```

Careful!

- Containers are running in **priviledged mode**

TODO

- Depending on a common abstraction to clightning, LND and charge.
- Making Lightning implementations parametrizable

Thank you

https://github.com/dgarage/LightningBenchmarks