
週間報告

— T18I917F 沈嘉秋 2019/11/21 —

作業内容

バグ修正 → 条件次第でデッドロックが発生していたので直した

仮説の検証、ベンチマーク

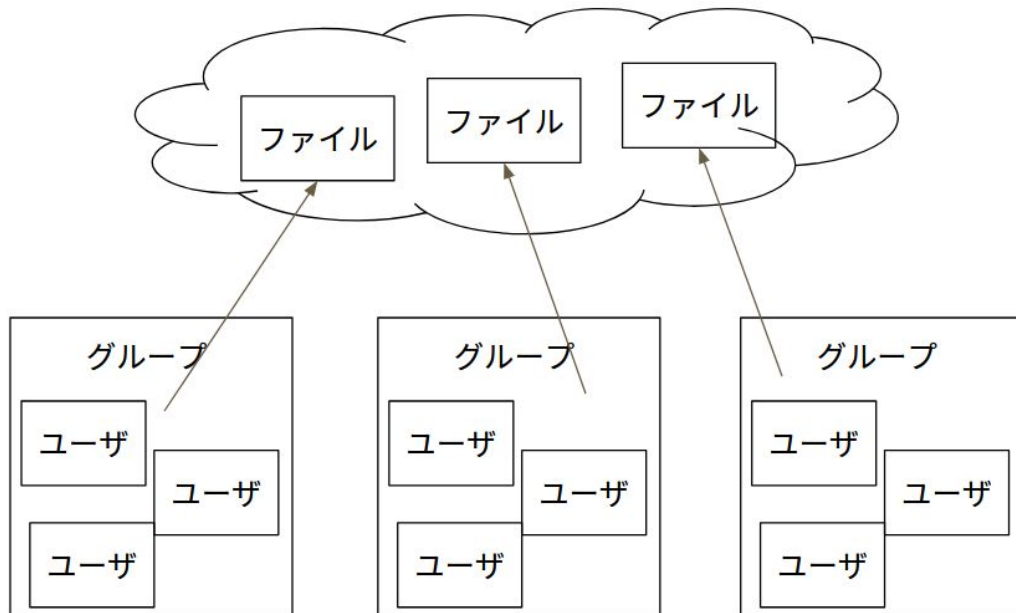
ベンチマーク

Layered KadがKademliaより優れている結果を出せるシナリオを考えた

測定項目：ネットワーク全体での通信量

シナリオ：

ネットワークに数種類（とりあえず3種）のファイルを保存し、ユーザのグループをファイルの種類分に分けて、保存されたファイルを探索する



```
const divide = 3;

const urls = await Promise.all(
  [...Array(divide)].map(async () => {
    const store = actors.shift()!;
    const res = await store.seeder
      .storeStatic(store.mainNet.kid, Buffer.from("value"))
      .catch(() => {});
    if (!res) throw new Error("fail");
    return res.url;
  })
);

const groupe = nodes.length / divide;
const values = (
  await Promise.all(
    actors.map(async (actor, i) => {
      const url = urls[Math.floor(i / groupe)];
      const res = await actor.user.findStatic(url).catch(() => {});
      if (res) return res;
    })
  )
).filter(v => !!v) as ArrayBuffer[];
```

結果

GithubActionsで計測

ノード数は40

・データの送受信回数

Layered Kad : 185回

Kademlia : 268回

```
▼ ✓ benchmark
14 [5/5] Building fresh packages...
15 Done in 7.74s.
16 yarn run v1.19.1
17 $ jest --forceExit
18   console.log layered/traffic/benchmark.ts:10
19     layered/traffic start
20
21   console.log layered/traffic/benchmark.ts:19
22     node setup done 40
23
24   console.log layered/traffic/benchmark.ts:10
25     layered/traffic findvalue 9
26
27   console.log layered/traffic/benchmark.ts:10
28     layered/traffic end bench 1081.598s 185
29
30   console.log mix/traffic.bench.ts:16
31     reset 0
32
33   console.log kad/traffic/benchmark.ts:8
34     kad/traffic start
35
36   console.log kad/traffic/benchmark.ts:8
37     kad/traffic findvalue 9
38
39   console.log kad/traffic/benchmark.ts:8
40     kad/traffic end bench 808.455s 268
41
```

Layered Kadの方がデータの送受信回数が少なく、
ネットワーク全体での通信量が少ないことがわかった。

つぎにすること

ベンチマークのパラメータを変えて変化を試みる

他のシナリオを考える

リファクタリング

ネットワークに数種類(とりあえず 3種)のファイルを保存し、ユーザのグループをファイルの種類分に分けて、保存されたファイルを探査する

