

## ローカルの UDP

Java プログラム同士で UDP 通信を使って  
双方向でメッセージをピンポンで送信するときの時間を測定してみました。  
指定したバイト数のメッセージをクライアントからサーバに送信,  
サーバは同じサイズのメッセージを送り返す, というもの。  
100000 回繰り返して測定。

経過時間

スループット

トータルのメッセージ (メッセージ長 \* 2 \* 送受信回数) を  
経過時間で割った値を可視化した結果。

実行環境

Mavericks な Mac mini で実行。  
system\_profiler SPHardwareDataType は な感じ。

```
Model Name: Mac mini
Model Identifier: Macmini6,2
Processor Name: Intel Core i7
Processor Speed: 2.6 GHz
Number of Processors: 1
Total Number of Cores: 4
L2 Cache (per Core): 256 KB
L3 Cache: 6 MB
Memory: 16 GB
```

使ったプログラム

よくあるサンプルと同様な感じ。

```
// クライアント側
import java.net.*;
import java.util.*;

public class UDPClient{

    private DatagramSocket socket;
    private InetAddress addr;

    public UDPClient() throws Exception {
        socket = new DatagramSocket();
        socket.setSendBufferSize(16*1024*1024);
        addr = InetAddress.getByName("127.0.0.1");
    }

    private byte[] recv = new byte[65536];
    public boolean send(byte[] buf) throws Exception {
        DatagramPacket packet = new DatagramPacket(buf, buf.length, addr, 16384);
        DatagramPacket reply = new DatagramPacket(recv, recv.length);
        socket.send(packet);
        socket.receive(reply);
        return true;
    }

    public static void main(String [] args) throws Exception{
```

```

        UDPCient test = new UDPCient();
        byte[] buf = new byte[Integer.parseInt(args[0])];
        int times = 100000;
        long t0, t1;
        t0 = Calendar.getInstance().getTimeInMillis();
        for(int i = 0; i < times; i++){
            test.send(buf);
        }
        t1 = Calendar.getInstance().getTimeInMillis();
        test.socket.close();
        System.out.println(buf.length + " " + (t1-t0));
    }
}

```

// サーバ側

```

import java.io.*;
import java.net.*;

public class UDPServer{

    private DatagramSocket socket;
    private byte[] buf = new byte[65536];

    public UDPServer() throws Exception {
        socket = new DatagramSocket(16384);
        socket.setReceiveBufferSize(16*1024*1024);
    }

    public boolean receive() throws Exception {
        DatagramPacket packet= new DatagramPacket(buf, buf.length);

        socket.receive(packet);

        byte[] reply = packet.getData();
        int len = packet.getLength();
        SocketAddress sockAddress = packet.getSocketAddress();
        DatagramPacket replyPacket = new DatagramPacket(reply, len, sockAddress);
        socket.send(replyPacket);

        return true;
    }

    public static void main(String [] args) throws Exception{
        UDPServer test = new UDPServer();
        while(test.receive() == true) ;
    }
}

```