

計算機システム概論
ネットワークとプロトコル、
LANとWAN (2)
2011/5/25

門林雄基

NAIST 奈良先端科学技術大学院大学

講義のポイント

2

- どのようにしてネットワークを使うのか？
 - ネットワーク利用のモデル

- ネットワークプログラミングとは？
 - プログラミングのモデル

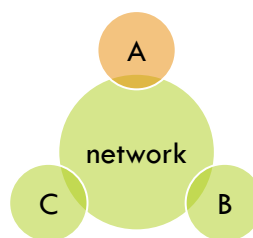
ネットワーク利用のモデル

ごく簡単に

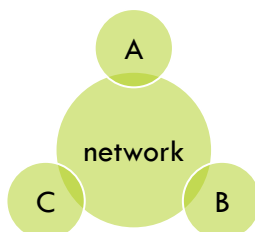
ネットワーク利用のモデル(1)

4

□ サーバ・クライアント



□ ピアツーピア



Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

ネットワーク利用のモデル(2)

5

- オープン
 - インターネット等

- クローズ
 - イン트라ネット (Intranet)
 - 構内網としての利用
 - VPN (Virtual Private Network)
 - オープンなネットワーク上に閉じたネットワークを構成

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

プログラミングモデル

網羅的かつ体系的に

プログラミングモデルの分類

7

- 1対1通信
 - Request-response protocol
 - Remote procedure call
- 1対多通信
 - Broadcast/multicast messaging
- 多対多通信
 - Message passing interface
 - File sharing
 - Tuple space
 - Message-oriented middleware

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

1対1通信

8

- 順にみていこう
- Request-response protocol
- Remote procedure call

- 同期式(基本)
- 非同期式(変形)

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

Request-response protocol

9

- 同期式

```

sequenceDiagram
    participant client
    participant server
    client->>server: Hello server
    server-->>client: Hello client
    client->>server: Get info
    server-->>client: Here is info
    
```

- 例: HTTP
 - Hypertext transfer protocol (RFC 2616)

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

Request-response protocol example: HTTP

10

```

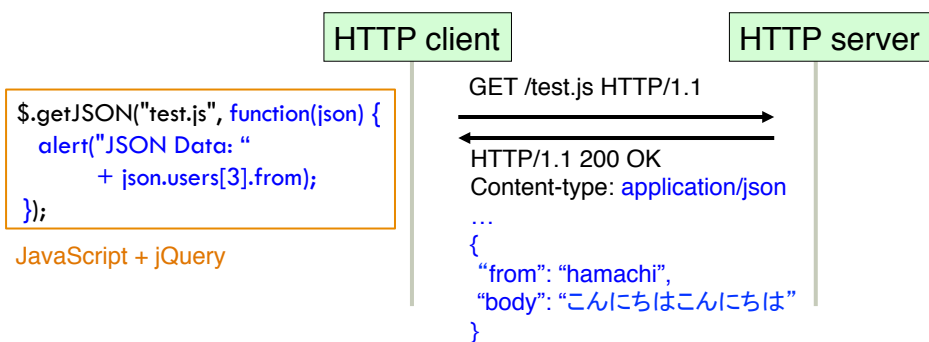
sequenceDiagram
    participant client as HTTP client
    participant server as HTTP server
    client->>server: GET / HTTP/1.1
    server-->>client: HTTP/1.1 200 OK  
Date: Wed, 02 Jun 2010 04:57:17 GMT  
Server: Apache-Coyote/1.1  
Content-Type: text/html;charset=utf-8  
Set-Cookie: JSESSIONID=5619B57CC; Path=/  
...
    client->>server: GET /leaderboard.html HTTP/1.1  
Host: www.theserverside.com  
Cookie: JSESSIONID=5619B57CC  
...
    server-->>client: HTTP/1.1 200 OK ...
    
```

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

Async request-response protocol

11

- 非同期式 (アプリケーション視点)
- AJAX: Asynchronous JavaScript + XML

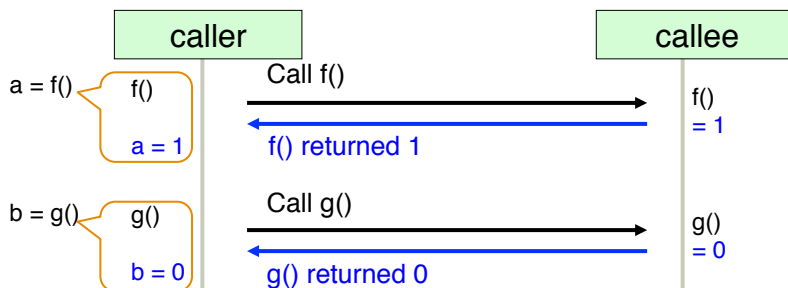


Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

Remote procedure call (RPC)

12

- 同期式
- SunRPC
- SOAP, XML-RPC



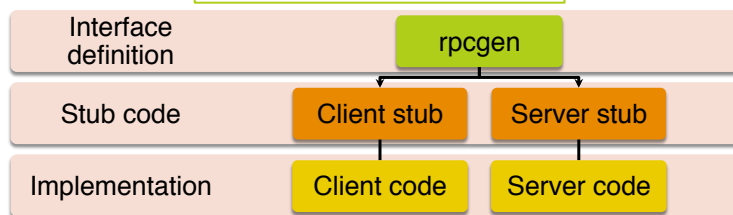
Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

Remote procedure call: SunRPC

13

```
program MESSAGEPROG {
  version PRINTMESSAGEEVERS {
    int PRINTMESSAGE(string) = 1;
  } = 1;
} = 0x20000001;
```

Source:
rpcgenプログラミングガイド



```
clnt = clnt_create(server, MESSAGEPROG,
  PRINTMESSAGEEVERS, ...);
result = printmessage_1(&message, clnt);
```

```
int *printmessage_1(char **msg, struct svc_req *req)
{
  static int result = 1;
  return(&result);
}
```

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

Remote procedure call: SOAP

14

- HTTP上でRPCを実現
- 実装言語を超える

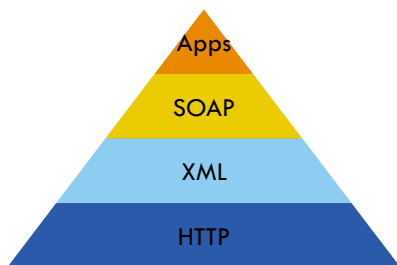
Perl

```
use SOAP::Lite +autodispatch =>
  uri => 'urn:',
  proxy => 'http://localhost/';

my $p = My::PingPong->new(10);
print $p->next, "\n";
```

Java

```
import org.apache.axis.client.Call;
import org.apache.axis.client.Service;
...
String endpoint =
  "http://ws.apache.org:5049/axis/services/echo";
Service service = new Service();
Call call = (Call) service.createCall();
call.setTargetEndpointAddress( new java.net.URL(endpoint) );
...
String ret = (String) call.invoke( new Object[] { "Hello!" } );
```



Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

1 対多通信

15

- Broadcast messaging
 - ▣ 全員にメッセージを送信
- Multicast messaging
 - ▣ 送信者はグループ宛に送信
 - ▣ 受信者はグループからメッセージを受信



Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

Multicast messaging example: jgroups

16

- 信頼性のある1対多通信
Reliable multicast communication

```
JChannel channel=new JChannel("/home/bela/udp.xml");
channel.setReceiver(new ReceiverAdapter() {
    public void receive(Message msg) {
        System.out.println("received msg from " + msg.getSrc() + ": " + msg.getObject());
    }
});
channel.connect("MyCluster");
channel.send(new Message(null, null, "hello world"));
channel.close();
```

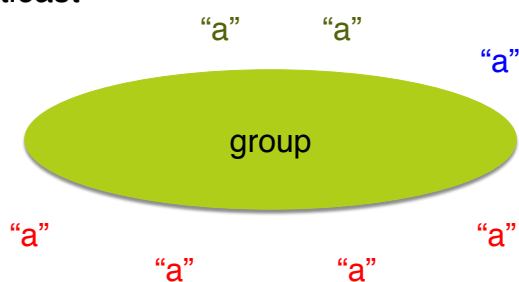
Source: www.jgroups.org

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

マルチキャストにおける多対多通信

17

- 特定の送信者
 - Source-specific multicast
- 送信者が誰でもよい
 - Any-source multicast



Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

多対多通信

18

- 順にみていこう
- Message passing interface
- File sharing
- Tuple space
- Message-oriented middleware

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

Message passing interface

19

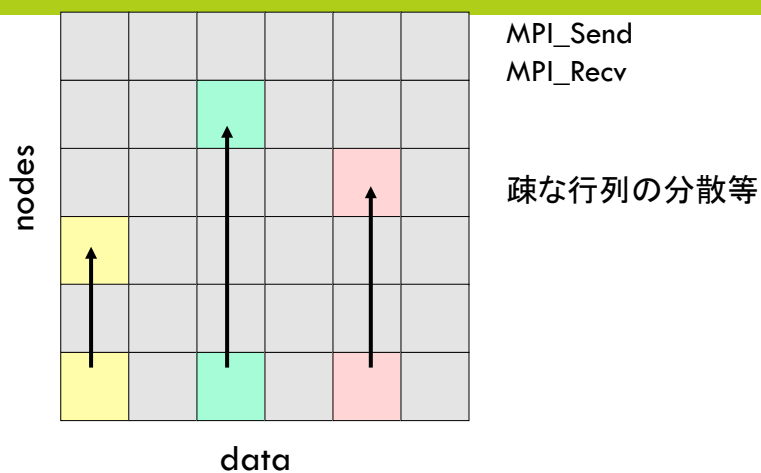
- MPI: Message Passing Interface
- 主として超並列計算用途

- Send / receive
- グループ内の通信
- Scatter / gather
- Reduce
- バリア同期

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

MPIにおける送受信

20



Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

同期／非同期での送受信

21

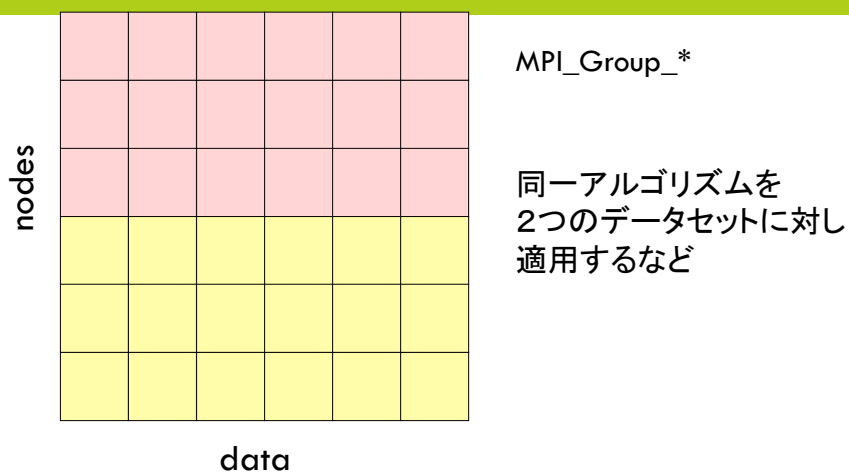
- 同期 (blocking)
 - ▣ MPI_Send
 - ▣ MPI_Recv

- 非同期 (non-blocking)
 - ▣ MPI_Isend
 - ▣ MPI_Irecv

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

グループ内の通信

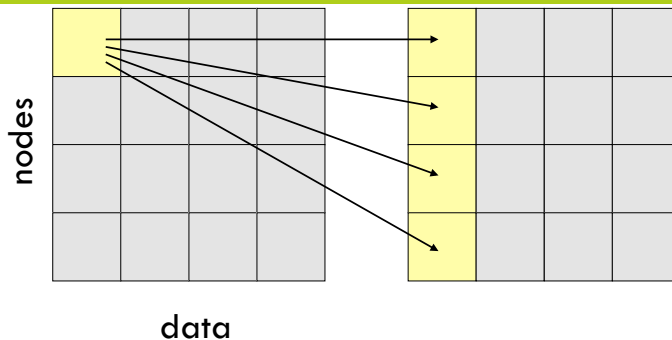
22



Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

MPIにおける集合通信(1): Broadcast

23



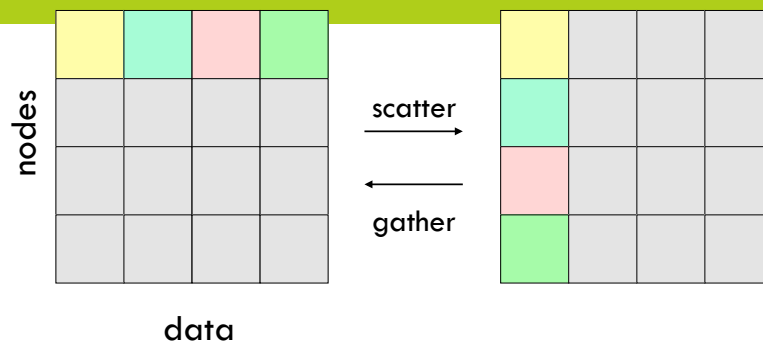
MPI_Bcast

用途例: 初期パラメータの分配など

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

MPIにおける集合通信(2): Scatter / gather

24



MPI_Scatter

MPI_Gather

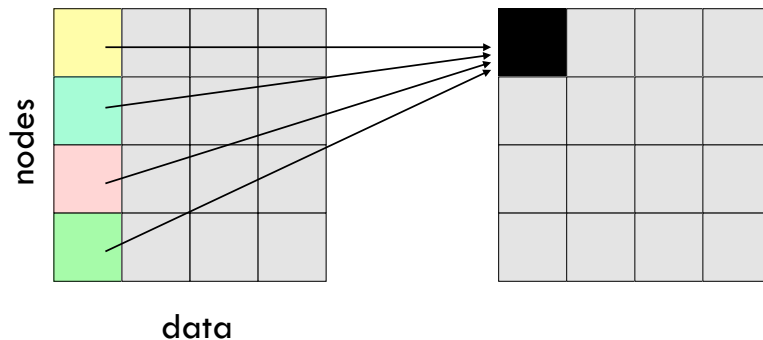
用途の例: ベクトルのノード群への分割など
行列積の例 ([uniprocessor](#), [MPI](#))

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

Global reduction operations: Reduce

25

applies similarly to all columns



MPI_Reduce

MPI_SUM, MPI_PROD, MPI_MAX, MPI_MIN, ...

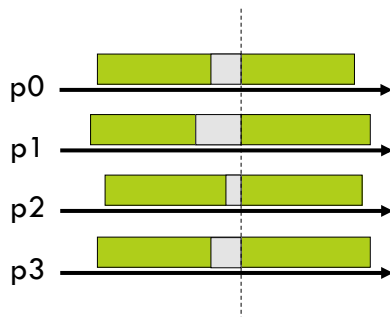
用途例: ベクトルと行列の積

Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

バリア同期

26

- MPI_Barrier
- 全ノードが MPI_Barrier を呼び出すまでブロック

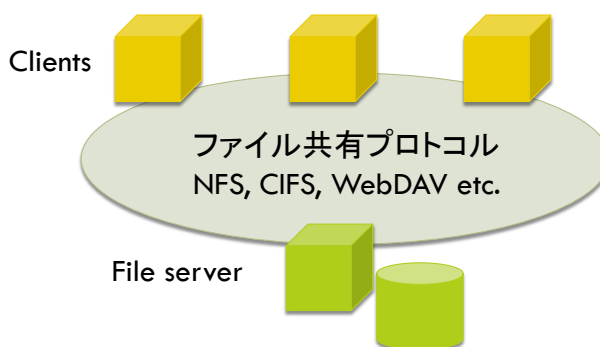


Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

File sharing

27

- 大規模データセットの簡単なやりとり等
- スクリプト言語で十分な場合も多い



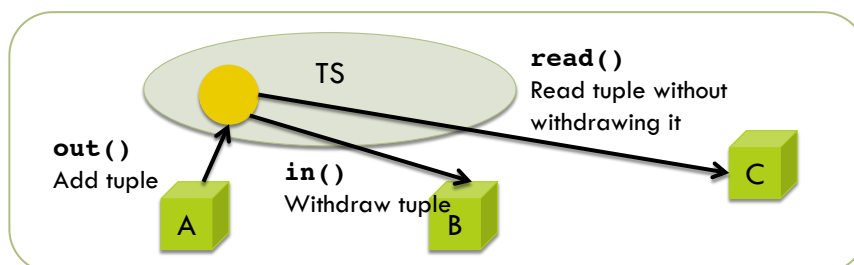
Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

Tuple space

28

- Linda
- JavaSpaces

```
import net.jini.space.JavaSpace;
...
MessageEntry msg = new MessageEntry();
msg.content = "Hello there";
Lookup finder = new Lookup(JavaSpace.class);
JavaSpace space = (JavaSpace) finder.getService();
space.write(msg, null, 60*60*1000);
```

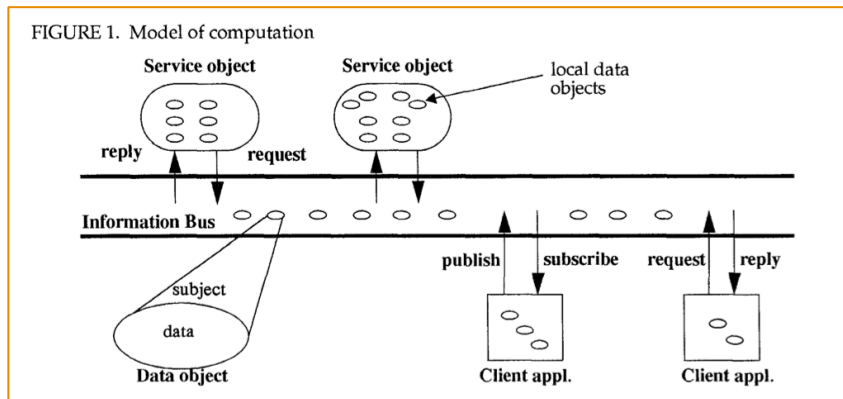


Copyright(C)2011 Youki Kadobayashi. All rights reserved. 11/05/25

Message-oriented middleware

29

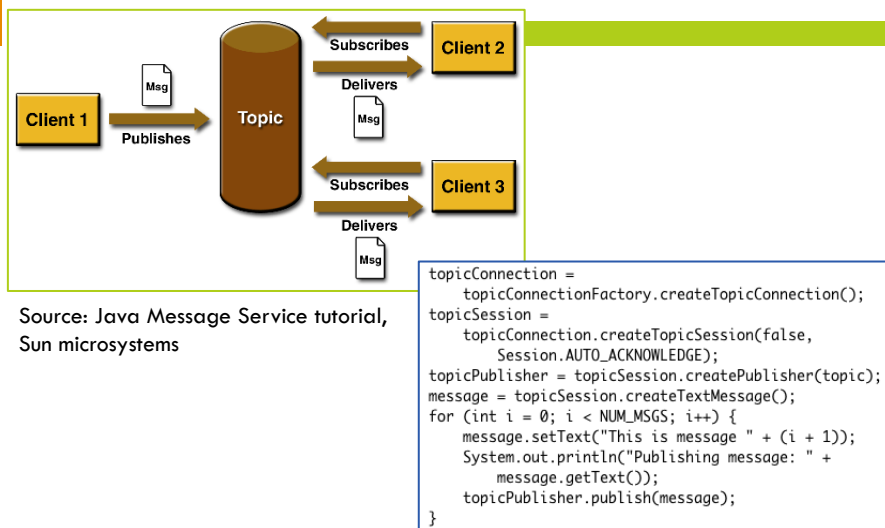
□ 非同期式



Source: The Information Bus: Architecture for Extensible Distributed Systems, SIGOPS'93.

MOM example: Java Message Service

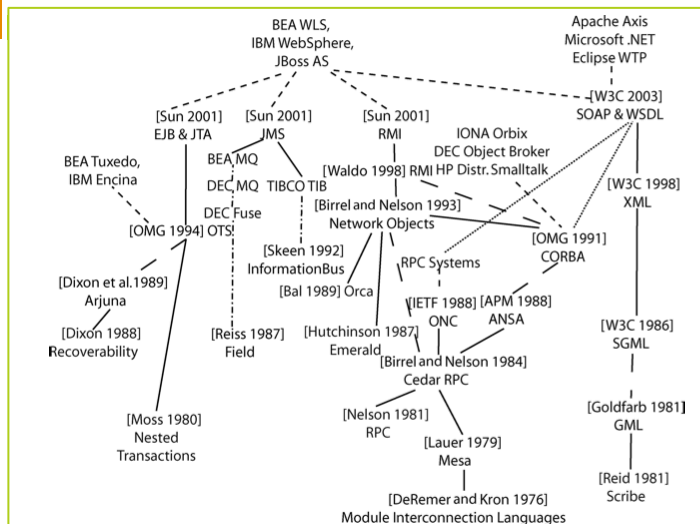
30



Source: Java Message Service tutorial, Sun microsystems

ネットワークプログラミングの系譜

31



The Impact of Research on the Development of Middleware Technology, ACM Trans. on Software Engineering Methodology, August 2008.

ネットワークプログラミング鳥瞰図

32

□ 気になったものを調べてみよう

1対1／多対多			
同期／非同期			
使用言語			
データ型			
データサイズ			
処理速度			