

プログラミング環境特論

P2PコンピューティングとJXTA

佐藤 三久
筑波大学

プログラミング環境特論

内容

- ◆ グリッドとP2P
- ◆ P2Pとはなにか、P2Pでできること
- ◆ P2Pの歴史、プロダクト
- ◆ JXTAの概要、概念、要素
- ◆ JXTAのプログラミングの様子
- ◆ 終わりに

プログラミング環境特論

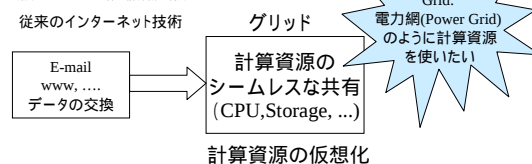
グリッドって？

- ◆ 世界中のPCをつないで計算 seti@home,...
 - これだけではない！
- ◆ スーパーコンピュータをつないで大規模計算
 - これだけではない！

プログラミング環境特論

“グリッド”とは

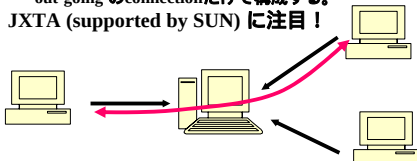
- ◆ グリッド技術とは広域の高速ネットワーク上において大量のデータ、計算資源、貴重な装置等を共有し、協調作業、資源の有効活用するネットワーク基盤技術（ソフトウェア、ネットワーク、ハードウェア）と、これを活用する応用技術
 - 大量のPC（クラスター）などの計算資源の相互に共有し、大量かつ大規模な計算を行う。
 - 大規模なデータの処理。例えば、加速器の制御データ、電波観測データの大量のPCなどの計算資源で、処理する技術が注目されている。
 - スーパーコンピュータ等の高速な計算機を融合し大規模な計算を行う（meta-computing）。
 - 遠隔の計算資源と手元のPCなどの計算機をシームレスに融合する技術（computing portal）。
 - 電子機器類、衛星などの異質な装置の連携共有
 - 電子システムなどの共同作業のサポート
 - 研究室などにある遊休の計算機の利用



プログラミング環境特論

P2Pコンピューティング

- ◆ P2P (peer to peer)コンピューティング
 - ファイル交換ソフト(Gnutella,...)で注目
 - 各PCが自立的につながる形態
 - 各PCは対等の関係(?)
 - 各PCは、ファイヤウォールの中でもつなぐことができることが重要
 - out-going connectionだけで構成する。
 - JXTA (supported by SUN) に注目！



プログラミング環境特論

P2Pコンピューティングの目的

- ◆ 潜在的には世の中に数百万台（以上？）のコンピュータがあり、これがネットワークに接続されている。
 - 社内、大学でもフルに活用されていない資源がたくさんある
 - 計算資源としてもちいることができないか？
 - ファイルを分散して格納できないか？
- ◆ 目的
 - ファイル共有
 - インスタントメッセージ
 - コラボレーションツール
 - 分散コンピューティング

プログラミング環境特論

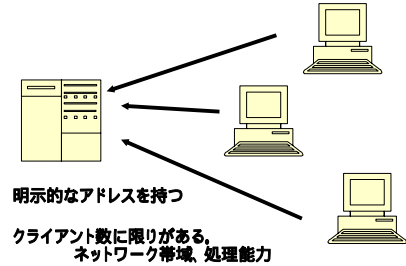
P2Pと“グリッド”の違い

- ◆ “グリッド”
 - (比較的) ノード数が少ない (100~1000?)
 - (比較的) 信頼できるノード
 - dedicated resourceのことが多い (多数のユーザがいる)
 - pushモデル(jobをsubmitする)
- ◆ P2P
 - ノード数が多い (数万にも及ぶ)
 - ノードは信頼できない **volatile**
 - 壊れているわけではない (もちろん、壊れることもある@)
 - 誰かが使っていることが多い
 - pullモデル
 - 自発的に参加

プログラミング環境特論

従来の分散コンピューティングのモデル

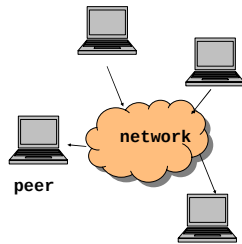
- ◆ クライアント・サーバモデル
 - WWW、FTP、e-mail



プログラミング環境特論

P2Pの分散コンピューティングのモデル

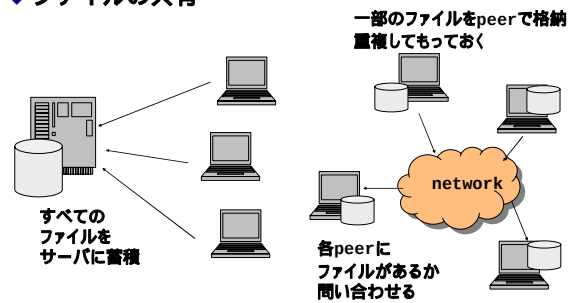
- ◆ サービスをネットワーク上のすべてのpeer間で分担する。
 - 1つのpeerでの障害で停止しない - 耐故障、信頼性
 - 末端の帯域まで活用できる - スケーラブル
 - 低コスト
- ◆ peerは何をするのか
 - クライアントとして
 - サービスを他のpeerに要求する
 - 要求の回答を受け取る
 - サーバとして
 - サービスの要求を受け取る
 - サービスを実行
 - サービスの結果を送る
 - サービスの要求を他のpeerに伝達する



プログラミング環境特論

P2Pで何ができるか (1)

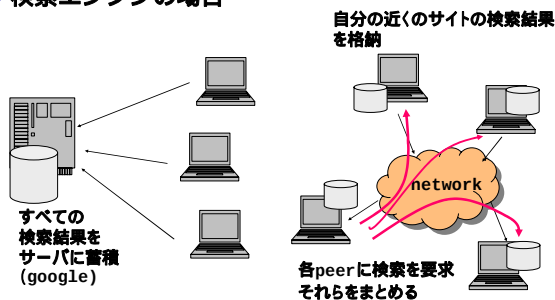
- ◆ ファイルの共有



プログラミング環境特論

P2Pで何ができるか (2)

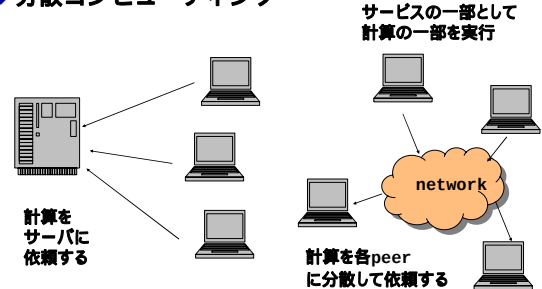
- ◆ 検索エンジンの場合



プログラミング環境特論

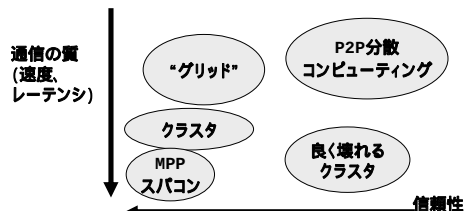
P2Pで何ができるか (3)

- ◆ 分散コンピューティング



P2Pで何ができるか (4)

- ◆ ちょっと、コメント
- ◆ P2Pは、スパコンにとって変わるものではない！
 - アプリケーションをえらぶ。coarse-grain...
 - 何をさせるかは問題！...でも、パラメータサーチはたくさんある



P2Pで解決しなくてはならないこと

- ◆ 静的なIPアドレスには限りがある
 - NATの問題
- ◆ マシンがプライベートアドレスの中にある
 - firewallの問題
- ◆ peerがvolatileであること
 - ネットワークが切れる
 - 違うルーティングをする
 - マシンが停止する
 - しかし、逆にこれらの問題点を解決することが高信頼なしシステムを構築するキーとなる

P2Pの歴史

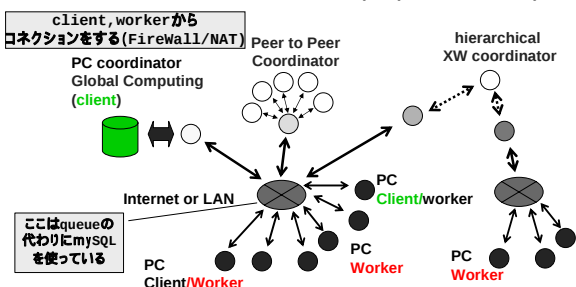
- ◆ usenet (日本だとWIDE)
 - mailやnewsのリレーによるネットワーク
- ◆ Instant messaging (ICQ, AOL, MSN,...)
 - ユーザの管理、登録はサーバがやるが、peer間のメッセージのやり取りは個々のpeer間で行う
- ◆ ファイル共有 (napstar, guntella, freenet, ...)
 - ファイルのある場所はサーバで管理 (napstar)
 - 検索も分散化 (guntella)
 - 問題点もたくさんある
 - tragedy of the common
 - Free ridingの問題
- ◆ 分散コンピューティング (seti@home, distributed.net, United Devices)
- ◆ Project JXTA

P2Pコンピューティングプロジェクト

- ◆ XtermWeb
 - MPIもあり
- ◆ BOINC
 - seti@homeから派生した (ツール化) プロジェクト
 - XtermWebと良く似ている
- ◆ コマーシャル
 - Entropia
 - UD GridMP
 - InterGrid
 - LSF...
 - 必要なのは、ジョブのdispatchだけでなく、認証、障害復帰、...

XtremWeb: General Architecture

- ◆ XtremWeb: Free, OpenSource, PC-Grid framework
- ◆ For research studies and production
- ◆ 3 entities: client/coordinator/worker (diff protect. domains)



Project JXTA

- ◆ Bill JoyとMike Claryらにより、Sun Microsystemsで始まったプロジェクト。
 - 現在ではJXTAコミュニティで運営されている
 - www.jxta.org
- ◆ P2P通信のコア機能とアプリケーション開発のフレームワークを提供
- ◆ プロトコル
 - Peer Discovery Protocol - peerがpeerサービスを発見できるようにする
 - Peer Resolver Protocol - peerで要求を送信・処理できるようにする
 - Rendezvous Protocol - peer間のメッセージ伝達の詳細を処理
 - Peer Information Protocol - 他のpeerから状態情報を得る
 - Pipe Binding Protocol - 仮想通信チャネルのサポート
 - Endpoint Routing Protocol - メッセージのルーティング
- ◆ 通信にはXMLが使われている
 - Javaでなくてもよい。

プログラミング環境特論

P2Pネットワークの要素

- ◆ 一群のデバイス (PC) がお互いのリソースとサービスを共有するためには、...
 - デバイスは他のデバイスの存在をどうやって知るのか
 - いろいろなデバイスをお互いに利益があるようにどのように組織すればいいのか
 - デバイスは自分のできることをどうやって知らせるのか
 - デバイスを一意に識別するにはどんな情報が必要か
 - デバイスはデータをどうやって交換するのか

プログラミング環境特論

P2Pネットワークの要素: peer

- ◆ Peer
 - P2Pネットワークのノード (PC, PDA, サーバ、...)
 - peerとは、有益な仕事を遂行でき、その結果をネットワーク上のほかのエンティティに直接または間接的に伝えることのできる任意のエンティティ
 - simple peer
 - firewall内にあってもよく、ほかのpeerのサービスをしたり、サービスを要求する
 - rendezvous peer
 - 他のpeerやpeer resourceを発見するためのネットワーク上の場所を他のpeerに提供する。peerに関する情報をキャッシュする。
 - firewallの外か、特定のポートは開いていなくてはならない
 - router peer
 - firewallやNATによってネットワークから隔離されたほかのpeerと通信するためのメカニズムを提供するpeer

プログラミング環境特論

P2Pネットワークの要素: peer group

- ◆ peer group
 - peer groupとは、共通の利益・目的に資するように作られたpeer群にサービスを提供するグループ。peer groupは、そのグループのpeerメンバーに対して、P2Pネットワーク内の他のpeerがアクセスできないサービスを提供できる。
 - どのように分割されるか
 - 関係するpeerがグループとして協力しあうアプリケーションに基づいて分割
 - 関係するpeerのセキュリティ要件について定義
 - グループのpeerメンバーに関する状態情報の必要性について定義

プログラミング環境特論

P2Pネットワークの要素: ネットワークトランスポート

- ◆ ネットワークトランスポート
 - TCP, UDP
 - http, smtpでもよい
 - エンドポイント
 - ネットワーク上で伝送される送信元、あて先
 - パイプ
 - 2つ以上のエンドポイントを接続する一方方向の非同期な仮想的なチャネル
 - 入力パイプ、出力パイプの2つがある
 - メッセージ
 - エンドポイントからエンドポイントまでパイプで伝送されるデータのコンテナ

プログラミング環境特論

P2Pネットワークの要素: サービス

- ◆ サービス
 - 有益な仕事
 - ファイル転送、状態情報の提供、計算の実行
 - peer サービス
 - ネットワーク上の特定のpeerが他のpeerに提供する機能
 - 特定のpeerに固有のもので、そのpeerがdownしている時には使えない
 - peer group サービス
 - peer groupがそのグループのメンバーに提供する機能
 - peer groupの1つが生きていれば、提供できる

プログラミング環境特論

P2Pネットワークの要素

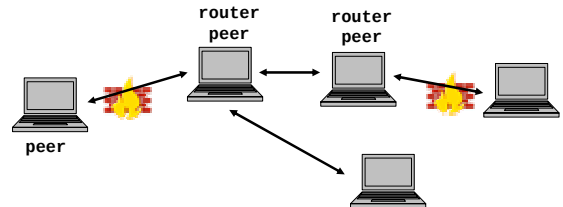
- ◆ アドバタイズメント (告知)
 - peerまたはpeer groupによって、P2Pネットワークの一部として提供されているエンティティ、サービス、リソースの構造化された表現のこと
 - JXTAでは、XMLで記述されている
- ◆ プロトコル
 - ネットワーク上のpeerを見つける
 - peerによってどのようなサービスが提供されているかを知る
 - peerから状態情報を取得する
 - peer上のサービスを呼び出す
 - peer groupを作成する。参加する。離脱する。
 - peerにデータ接続を行う
 - 他のpeerに当てられたメッセージをルーティングする
- ◆ エンティティネーミング
 - peer, peer group, pipe, contentsに対して、一意に識別できる識別子を与える

P2P通信

- ◆ どのようにしてP2Pネットワーク上のpeerとサービスを見つけるのか？
- ◆ アドバタイズメントの発見
 - 基本構成要素はアドバタイズメントとして表現できる。つまり、アドバタイズメントの発見として扱うことができる。
 - キャッシュから発見する（一定時間でキャッシュをフラッシュする）
 - 直接発見。peerに関してマルチキャストして見つける。
 - rendezvous peerを利用して発見する。rendezvous peerは、
 - ・他の自分の知っているpeerに発見要求をフォワードする
 - ・キャッシュされている内容を返す
 - ・要求がループしたり、過剰にならないように制御する必要がある

P2Pネットワークの構築

- ◆ simple peerはfirewall内にあつたりするために、router peer(rdv peer)を使う
 - firewall: 着信を拒否、or 特定のポートのみに制限。
 - NAT (network address translation): 静的NATと動的NAT



JXTAの長所と欠点

- ◆ XMLを用いることによって、プロトコルを高レベルに抽象化している
- ◆ WSDLのようなサービス記述は提供していない。これはアプリケーション側の問題としている。
- ◆ TCPに限定していない。複雑になることもあるが、柔軟性がある。
- ◆ Jiniとの違い
 - Javaに依存していない。広域のネットワークでも可。
 - 本当か？

JXTA shell

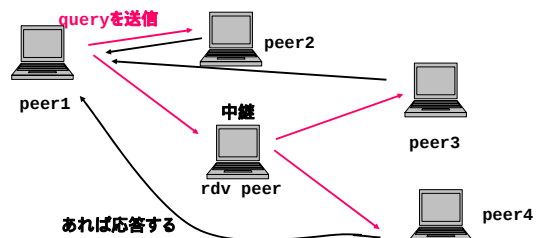
- ◆ 一つのアプリケーション
- ◆ JXTA環境のセットアップに用いる
- ◆ JXTAアプリケーションの実行環境を提供する。
 - もちろん、なしでもOK

JXTAの要素・プロトコル

- ◆ プロトコル
 - Peer Discovery Protocol - peerがpeerサービスを発見できるようにする
 - Peer Resolver Protocol - peerで要求を送信・処理できるようにする
 - Rendezvous Protocol - peer間のメッセージ伝達の詳細を処理
 - Peer Information Protocol - 他のpeerから状態情報を得る
 - Pipe Binding Protocol - 仮想通信チャネルのサポート
 - Endpoint Routing Protocol - メッセージのルーティング

Peer Discovery Protocol

- ◆ アドバタイズメントを発見するプロトコル
 - アドバタイズメントの発見を要求するメッセージ
 - 発見に回答するメッセージ



アドバタイズメント

- peer間で交換されるデータの基本単位
- peer, peer group, pipe, ...
- たとえば、peerだと

```
<?xml version='1.0'?>
<!DOCTYPE jxta:PA>
<jxta:PA xmlns:jxta="http://jxta.org">
<ID>
<jxta:url:xuid-59616261646162614A7874615032563F6B1622F89E24>
</ID>
<SID>
<jxta:jxta-NetGroup>
</SID>
<Name>
msat01
</Name>
<Svc>
<NCID>
<jxta:url:xuid-DEADBEEDFEAFBABAEEDBBE000000005>
</NCID>
<Param>
<jxta:RA xmlns:jxta="http://jxta.org">
</jxta>
<jxta:APA xmlns:jxta="http://jxta.org">
</jxta>
<jxta:url:xuid-59616261646162614A7874615032563F6B1622F89E24>
</jxta>
</Param>
tcp//127.0.0.1:9702
</ID>
</jxta>
relay://xuid-59616261646162614A7874615032563F6B1622F89E2403
</jxta:APA>
</SID>
</jxta:RA>
</Param>
</Svc>
</NCID>
```

Peer Discovery Protocol

- ### ◆ Query Messageの例

```
<?xml version="1.0" encoding="utf-8"?>
<jxta:DiscoveryQuery>
  <type> ... </type>
  <Threshold> ... </Threshold>
  <PeerAdv> ... </PeerAdv>
  <Attr> ... </Attr>
  <Value> ... </Value>
</jxta:DiscoveryQuery>
```

Peer Discovery ProtocolのAPI

```
// listenerを定義しておく
class exampleListener implements DiscoveryListener {
    public void discoveryEvent(DiscoveryEvent e){
        ... 発見されたら、呼ばれる...
    }
    ...
    theEnvironment = getEnv();
    ShellObject theShellObject = theEnvironment.get("stdgroup");
    PeerGroup currentGroup = (PeerGroup)theShellObject.getObject();
    ...
    DiscoveryService discovery = currentGroup.getDiscoveryService();
    ...
    //eventを受け取る側
    DiscoveryListener listener = new exampleListener();
    discovery.addDiscoveryListener(listener);

    //発見のリクエストを出す側
    discovery.getRemoteAdvertisement(null, type, null, null, thr, null);
}
```

Peer Discovery ProtocolのAPI

- ```
advertisement.setName(...);
advertisement.setDescription(...);
...
//... DiscoveryService discoveryを取得
// localに公開
discovery.publish(advertisement,DiscoveryService.GROUP,
 1000,1000); //type, lifetime,などを設定

// globalに公開
discovery remotepublish(advertisement,
 DiscoveryService.GROUP,1000);
```

## Peer Resolver Protocol

- Discovery Protocolはこれを使って行っている
- Resolver照会メッセージ
- Resolver応答メッセージ

```
<?xml version="1.0" encoding="UTF-8"?>
<jxta:ResolverQuery xmlns:jxta=http://jxta.org>
 <HandlerName> ... </HandlerName>
 <Credential> ... </Credential>
 <QueryID> ... </QueryID>
 <SrcPeerID> ... </SrcPeerID>
 <Query> ... </Query>
</jxta:ResolverQuery>
```

handlerName  
でマッチする

いろいろなフォーマット

いろいろなフォーマット  
を設定できる。

## Pipe Binding Protocol

- バイブバインド照会メッセージ
- バイブバインド応答メッセージ

パイプアドバタイズメント  
から入力パイプを生成し  
メッセージの到着をまつ

送りたい側のパイプ  
アダプタイズメントを  
照会するために、照会  
メッセージを送る



照会メッセージに対して  
自分のパイプアダプタイズ  
メントを返答メッセージでかえす。

応答メッセージから、返されたパイプアドバタイズメントをとりだし、その情報を使って出力パイプを作成する

#### プログラミング環境特論

### Pipe サービスのAPI

```
// 入力側

public PipeServer implements PipeMsgListener {
 //メッセージが到着したら呼ばれる
 public void PipeMsgEvent(PipeMsgEvent e){
 Message msg = e.getMessage();
 msg.getString()でメッセージをとりだす;
 }
 ...
 PipeAdvertisement pipeAdv =
 AdvertisementFactory.newAdvertisement(...); //すでにつくってあるもの
 DiscoveryService discovery = peerGroup.getDiscoveryService();
 discovery.publish(pipeAdv, DiscoveryService.ADV);
 discovery.remotepublish(pipeAdv, DiscoveryService.ADV);
 ...
 PipeService pipeserv = peerGroup.getPipeService();
 inputPipe = pipeserv.createInputPipe(pipeAdv, this);
 // pipeは、pipeアドバタイズメントとListenerからつくる
```

#### プログラミング環境特論

### Pipe サービスのAPI

```
// 出力側

public PipeClient implements OutputPipeListener {
 // バインドが確立したらよばれる
 public void outputPipeEvent(OutputPipeEvent e){
 outputPipe = event.getOutputPipe();
 // これを使って書き込み, たとえば ...
 pipeService pipeserv = peerGroup.getPipeService();
 Message msg = pipeserv.createMessage();
 msg.setString("なんか");
 outputPipe.send(msg);
 }
 ...
 PipeAdvertisement pipeAdv =
 AdvertisementFactory.newAdvertisement(...); //入力と同じもの
 ...
 PipeService pipeserv = peerGroup.getPipeService();
 pipeserv.createOutputPipe(pipeAdv, this);
 // pipeは、pipeアドバタイズメントとListenerからつくる
```

#### プログラミング環境特論

### サービスとピアグループ

- ◆ サービスはピアグループ内で作られる
- ◆ サービス
  - モジュールクラスアドバタイズメント
  - モジュール仕様アドバタイズメント
  - モジュール実装アドバタイズメント
- ◆ 詳しくは、...

#### プログラミング環境特論

### おわりに

- ◆ JXTAの印象
  - いろいろと野心的であるが、まだ未完成
  - 例えば、pipeの通信は信頼性が保障されていない。
- ◆ P2Pの将来
  - 本当にスケーラブルなのか、10000もサポートできるフレームワークはできるのか？
    - 特別なアプリであれば可能であるが、...
  - 有効なアプリケーションはあるのか？
  - 汎用になればなるほど社会的な問題がおきてくる
    - free ridingの問題 ...
    - P2Pに参加するmotivationの問題
    - ...