Peer-to-peer の世界

首藤一幸
ウタゴエ(株) 取締役 CTO

UTA GOE
UNIVERSAL GRID

自己紹介 – ウタゴエ

- ソフトウェア・サービス開発企業
- 従業員 19名 (常勤 10名, 非常勤 9名)
- 2001年 1月 設立
- ウタゴエ →「コミュニケーション」技術
- コア技術の 1つが、peer-to-peer ライブ映像配信技術

UTA GOE
UNIVERSAL GRID

自己紹介 – ウタゴエ

- 最近の報道
  - 2007/1/15 23時 TV東京 WBS も。

日経 CNBC 2006/12/7(木)

PCつなぎ動画配信

伊藤忠、早大発VBに出資

設備費を安く

伊藤忠商事は複数のパソコン（PC）をつないで処理能力を高めるグリッドコンピューティングの技術を活用した動画配信サービスに乗り出した。この技術を開発した早稲田大学発ベンチャーのウタゴエ（東京・渋谷、園田智也社長）と資本・業務提携し、同社が運営するソーシャル・ネットワーキング・サービス（SNS）内の動画配信スペースを企業に売り込む。テレビ通販会社などの番組配信や動画チャットなどの利用を開拓する。

「オーシャン・グリッド」技術を利用したサービスのイメージ

通販などに利用

伊藤忠がウタゴエの第三者割当増資に応じ、発行済み株式の八・四%を取得した。動画を使ったネット広告などの配信サービスが広がるとみて、独自技術を持つウタゴエと提携、事業拡大を図る。

ウタゴエは二〇〇一年一月設立。グリッドコンピューティングの手法を使い光ファイバーで効率的に動画を配信する技術「オーシャン・グリッド」（OG）を開発した。利用者はPCに専用ソフトを取り込み、視聴を希望する動画を選ぶと、配信を管理するサーバーが同じ動画を見ている利用者のPCを割り出し、直近でアクセスしたPCに自動的に接続を指示する。複数のPCを順次中継しながら動画を流すため、一度に多数のアクセスが集中しても配信用設備は従来の数十分の一程度の経費で済む。

▼グリッドコンピューティング　複数のコンピューターをネット経由で接続、それぞれに業務を分散し並列処理することで一台ずつの性能は低くても高速計算が可能になる。利用者のパソコンを動画配信の「中継点」とし

OGを使ったウタゴエ

日経産業新聞 2007/1/9(火) 1面

内容

- P2P とは何か
  - 15 枚
- オーバレイネットワーク
  peer-to-peer の展開
  - 23 枚
- オーバレイについての発展的な話題
  - 11 枚
- 研究開発の実際
  Overlay Weaver の場合
  - 15 枚

UTA GOE
UNIVERSAL GRID

P2P とは何か

UTA GOE
UNIVERSAL GRID

P2P とは？

- 世間の認識
  - 「P2P」 =「ファイル共有ソフト」
    - Napster, Gnutella, WinMX, eDonkey, Kazaa, Winny, Share, …
  - 著作権法違反, 著作権侵害の温床
    - 音楽,映像コンテンツ, ソフトウェア, …をぶっこ抜き！
  - コンピュータウィルス (ワーム) の流通経路
    - 感染 → 個人情報 / 機密情報流出

- ある種のソフトウェアがそのように使われたことは事実。
- 技術者が何を言おうと、社会的に、単語「P2P」はまっ黒。
- ビジネス上、「P2P」という単語には大きなリスクが。

- ただし、語源「peer-to-peer」の技術的・社会的な特性や価値は、冷静に議論する価値あり。

UTA GOE UNIVERSAL GRID

P2P とは？

- peer
  【ラテン語「平等」の意から】
  1. 同僚; 仲間; **同等の者**.

- peer-to-peer → P2P
  同等な者どうしのやりとり

- 「各ノードの役割が同じ。例えば、あるノードは、サーバ/クライアント両方の役割を果たす。」
  - IRTF P2P Research Group の憲章より

UTA GOE UNIVERSAL GRID

P2P とは？

- でも
  - 同等でないノード (例: サーバ) が存在する P2P システムもある。
  - ノードどうしが直接通信しないのに P2P と呼ばれるシステムもある。

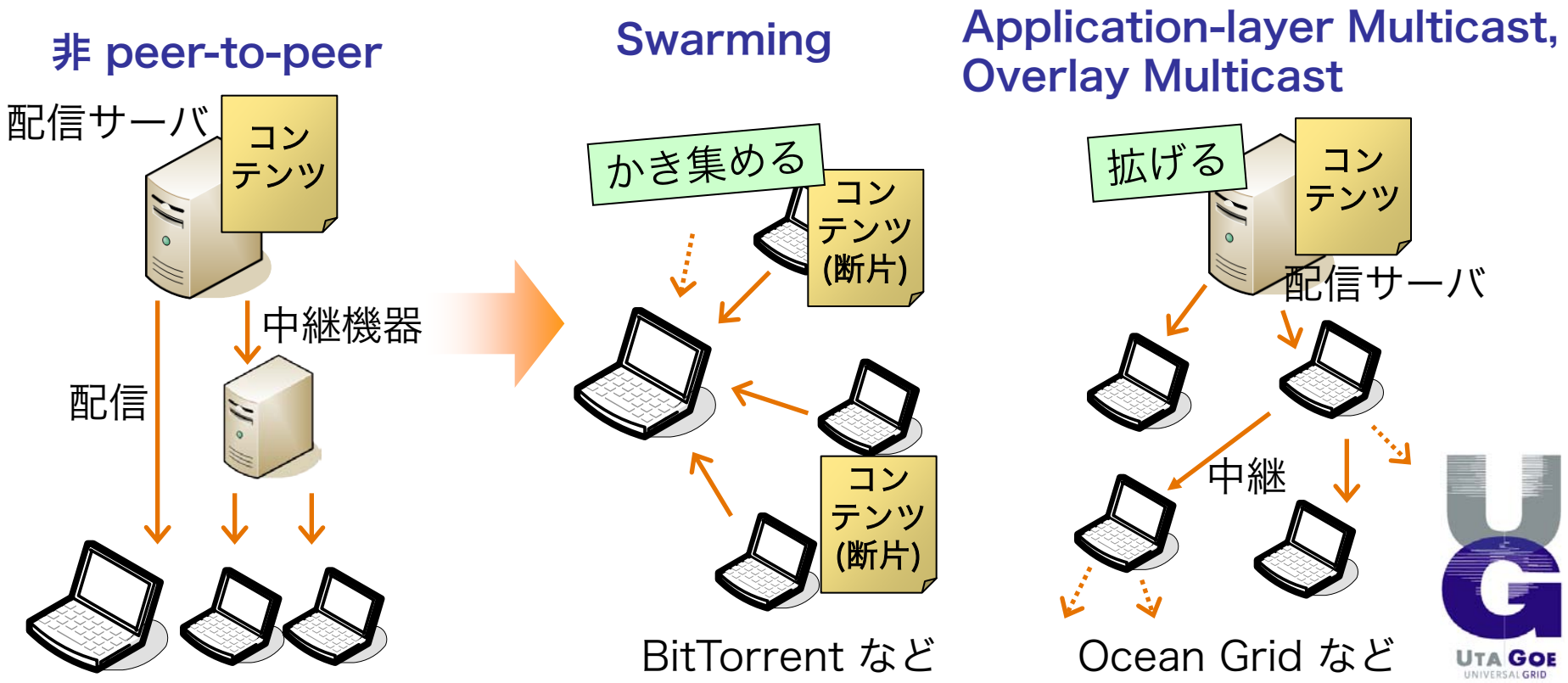
明確な定義はなく、様々な主張がある。

P2P と呼ばれるシステムを概観し、
P2P とは何かを考えていく。

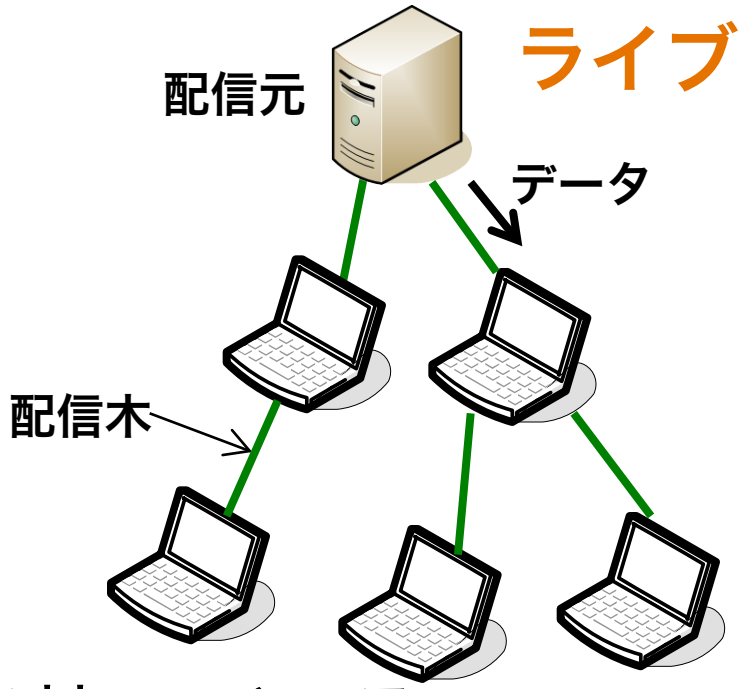
UTA GOE UNIVERSAL GRID

P2P コンテンツ配信

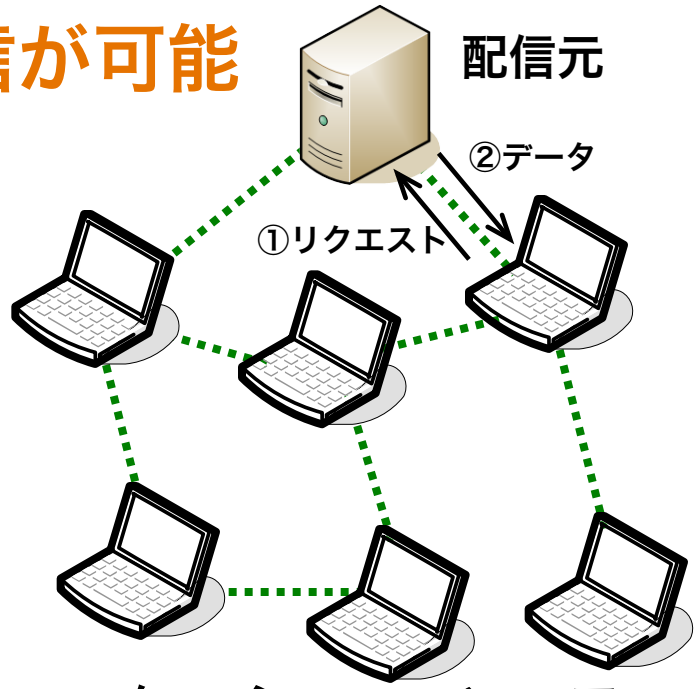
- 大容量コンテンツを配布 / 配信する仕掛け。
  - 例: BitTorrent, Kontiki (VeriSign社が買収), 当社 Ocean Grid, …
  - 含 ファイル共有
- 末端の PC が配信に協力する。



P2P コンテンツ配信
アプリケーション層マルチキャスト (ALM)

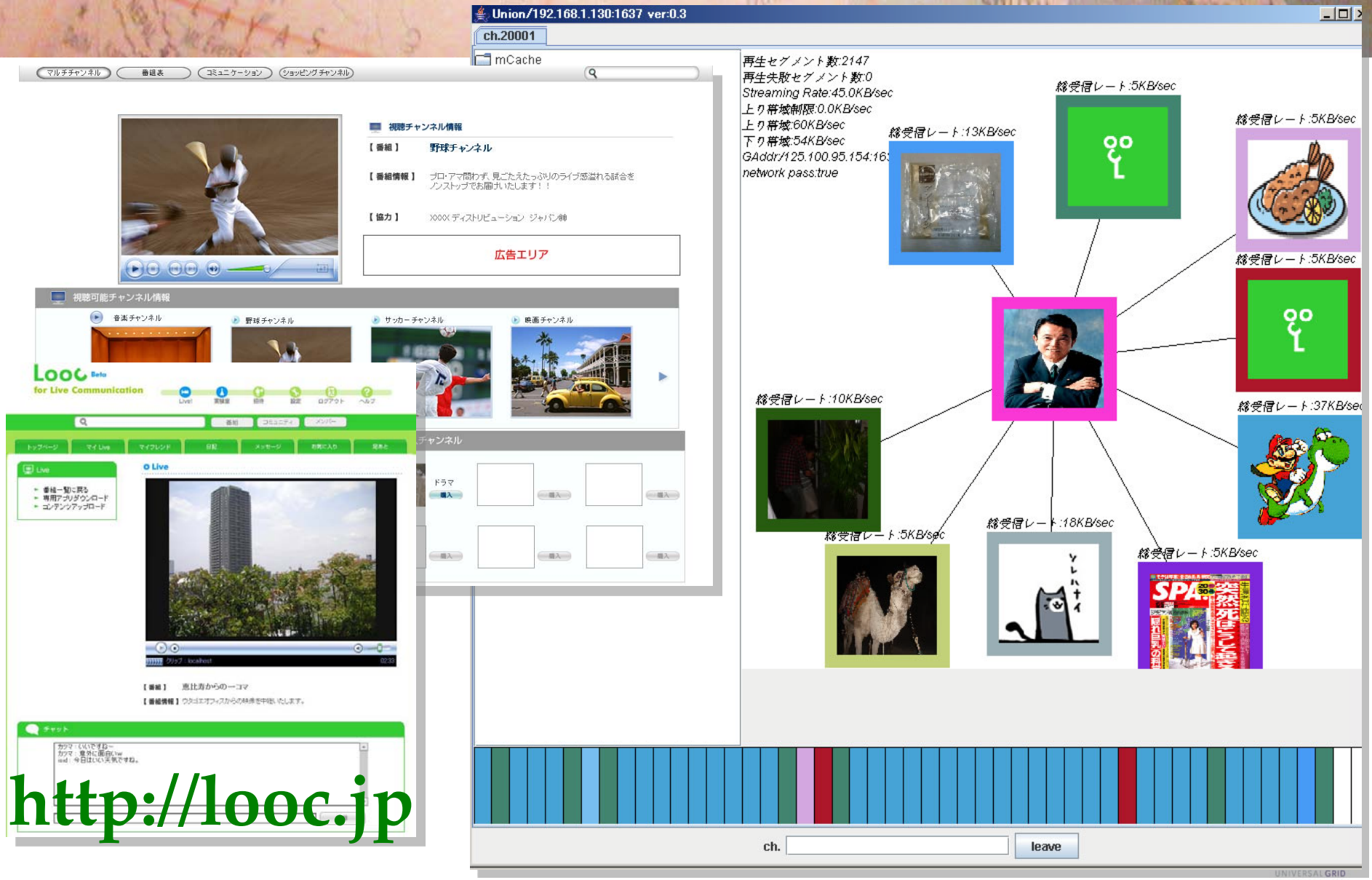


ライブ配信が可能



- ツリー ベース
  - 明示的に作ったトポロジがデータ流路。
  - 根から葉に向かってpush。
  - △ ノード故障時は、要 迅速な復旧。
  - ○ 末端まで確実に届く。配信遅延 小。
  - 広い上り帯域幅を使い尽くし得る。
- メッシュ ベース
  - 隣接ノードと緩やかな関係を保つ。
  - 隣接ノードからpull。
  - ○ 生来、ノード故障に強い。
  - △ 末端まで届く保証なし。要 補償策。
  - 狭い上り帯域幅を使い尽くし得る。

デモの代わりに


Union/192.168.1.130:1637 ver:0.3
ch.20001
mCache
再生セグメント数:2147
再生失敗セグメント数:0
Streaming Rate:45.0KB/sec
上り帯域制限:0.0KB/sec
上り帯域:60KB/sec
下り帯域:54KB/sec
network pass:true
総受信レート:5KB/sec
総受信レート:13KB/sec
総受信レート:5KB/sec
総受信レート:5KB/sec
総受信レート:10KB/sec
総受信レート:37KB/sec
総受信レート:5KB/sec
総受信レート:18KB/sec
総受信レート:5KB/sec
ch.
leave
視聴チャンネル情報
【番組】 野球チャンネル
【番組情報】 プロ・アマ問わず、見ごたえたっぷりのライブ感溢れる試合をノンストップでお届けいたします！！
【協力】 XXXXディストリビューション ジャパン㈱
広告エリア
視聴可能チャンネル情報
音楽チャンネル
野球チャンネル
サッカーチャンネル
映画チャンネル
Looc Beta
for Live Communication
Live
【番組】 恵比寿からの一コマ
【番組情報】 ウタエオフィスからの映像を中継いたします。
チャット
http://looc.jp

P2P グループウェア

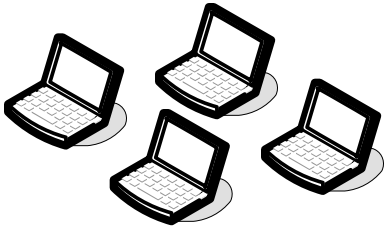
- 予定管理, 共有ファイル管理など、コラボレーションを支援するソフトウェア。
  - Microsoft Office Groove 2007, Ariel AirOne, ifreestyle, …
  - サーバなしで使い始められる。
  - PC間で直接、データをやりとりする。

Ariel MultiScheduler

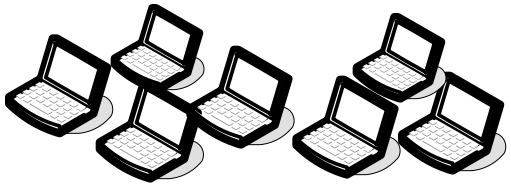
P2P 電話 - Skype

- 極力、サーバ (管理側提供マシン) を排してある。
  - 通常の IP 電話
    - 電話帳の保持, 通話の中継, … あらゆる処理をサーバが行う。
  - Skype
    - サーバが行う処理は、ユーザがログインする際の認証、だけ。
    - ユーザの検索処理すら、ユーザ側マシンが協調して行う。

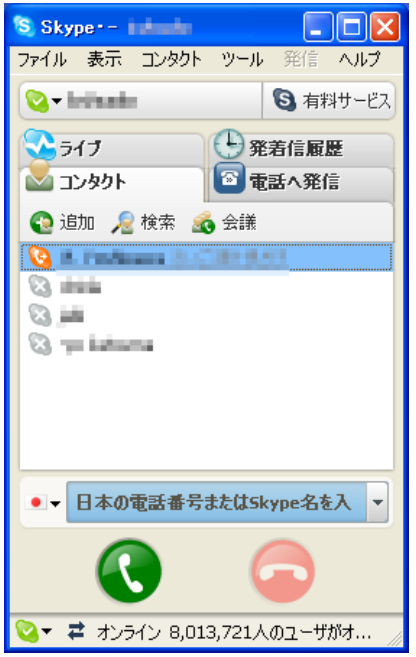
管理側 サーバ
•ログイン時の認証を行う。



スーパーノード
•選ばれし強力な ユーザ側 PC。
•ユーザ検索, 通話の中継, … を担当。



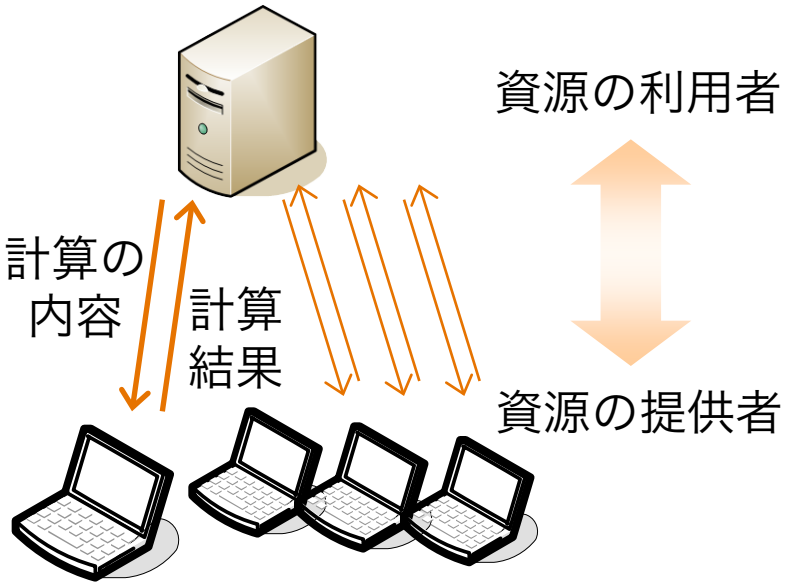
一般のノード
•通話は、極力、直接の通信で行う。
•場合により、スーパーノードを利用。



P2P 分散コンピューティング

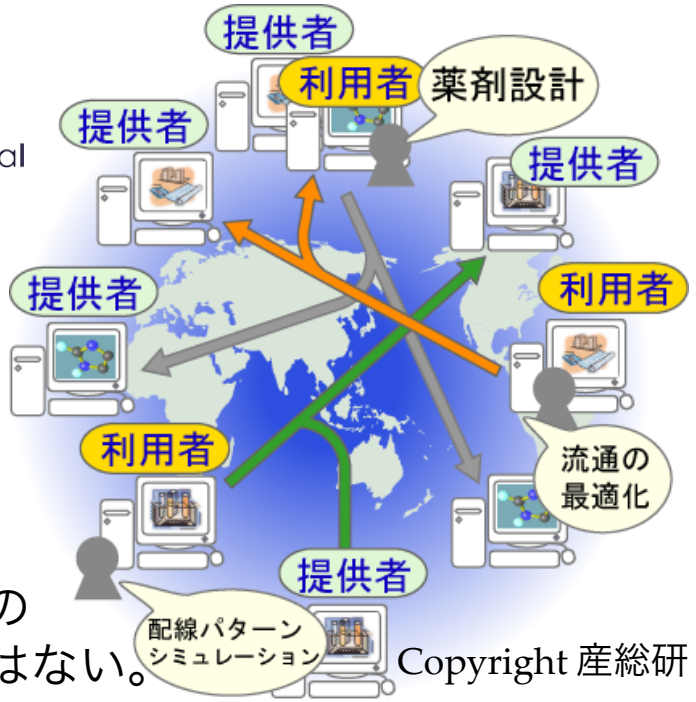
- 手持ちの PC では足りないような大きな計算を行う。
  - 計算プロジェクト: SETI@home, distributed.net, …
  - ソフトウェア: BOINC, United Devices 社 Grid MP, Entropia 社 DC Grid, XtremWeb, …
  - 別名: public-resource computing, volunteer computing, desktop Grid, …
- これを「P2P」と呼んだのは Intel 社？ 2000年 P2P-WG 設立の頃。

peer-to-peer
分散コンピューティング



P3 (産総研)

Personal Power Plant



提供者、利用者の
関係が固定的ではない。

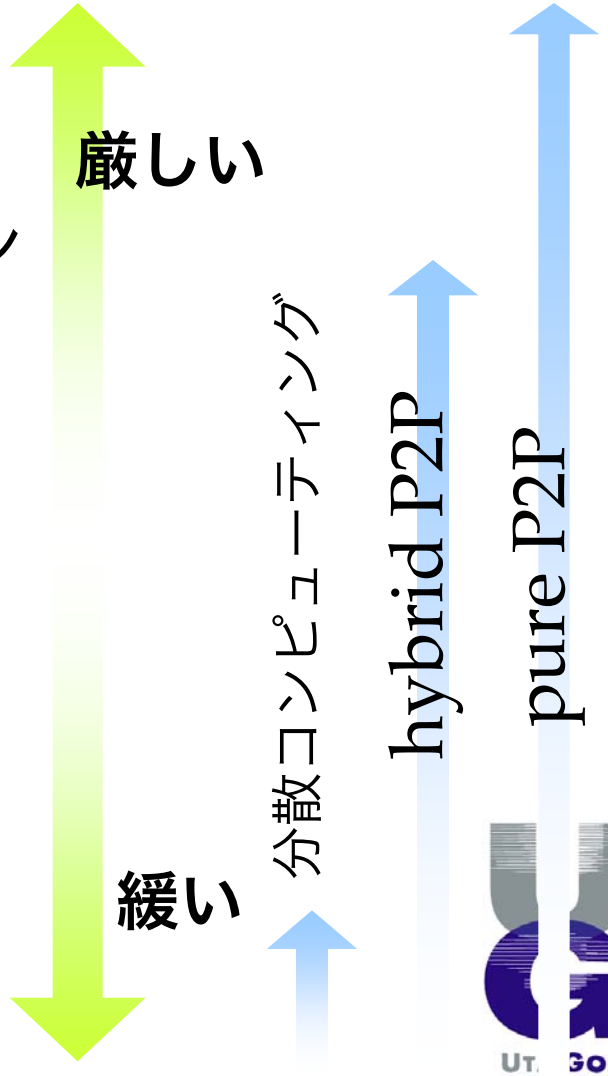
P2P ○○

- P2P コンテンツ配信
  - 含 ファイル共有
- P2P グループウェア
- P2P 電話
- P2P 分散コンピューティング
- P2P 掲示板
  - 新月, Winny 2, …
- P2P Instant Messenger
  - threedegrees, …

UTA GOE UNIVERSAL GRID

P2P とは？

- 集中的なサーバが存在しない。
- 各ノードの役割が同じ。例えば、あるノードはサーバ / クライアント両方の役割を果たす。
  - IRTF P2P Research Group の憲章
- システムの中で固定的なサーバを介さずにピア間で直接通信する場面があるシステム
  - 日経バイト誌 2004年 8月号
- 末端の機器が重要な役割を果たす分散システム



P2P とは？

- 「P2P」の緩い定義例：

末端の機器が重要な役割を果たす
分散システム

- 「重要な」
  - 重要か否かは、状況や主観に依る。
  - 技術用語の定義に使える言葉ではない。

- 「P2P」の意味・定義は、それが産まれた状況や、その時点での人々の主観を抜きには考えられない。

P2P とは？

- 「peer-to-peer」
  - 遅くとも 1980年代には使われていた言い回し。
  - 技術用語だった。
    - cf. UUCP 関連文書
    - cf. Macintosh の LocalTalk 関連文書
- 「P2P」
  - 2000年前後に産み出された言葉。
  - ウェブの反動？
    - ダムクライアントであるウェブブラウザ ⇔ P2P
  - 社会的な意味を期待されていた？いる？
    - SETI@home に対するワクワク感、興奮。

UTA GOE UNIVERSAL GRID

「P2P」の社会的な意味

技術的な性質 → 社会的な意味

- 末端ノード間の直接通信
- 末端ノードが重要な役割を果たす
- 各ノードの役割が同じ

- 人と人を直接結びつける技術 (P2P: person-to-person 説)
- 個人中心の社会観
- 階層的ではなくフラットな個人間ネットワーク

UTA GOE UNIVERSAL GRID

Web 2.0 との類似性・相違

- Web 2.0 = **参加するウェブ**
- それは「P2P」に期待されてきた社会的意味
  - 末端 / 個人を empower

- Web 2.0 なサービス
  - セマンティクス (意味) は P2P。
  - ただし、実装は集中型。非 peer-to-peer。

  - 実装は peer-to-peer でも何でもいい、とも言える。
    - 「あちら側の世界」(ウェブ進化論, 梅田望夫氏) は必要条件か？

  - というわけで、次は、最近の技術的な展開を review

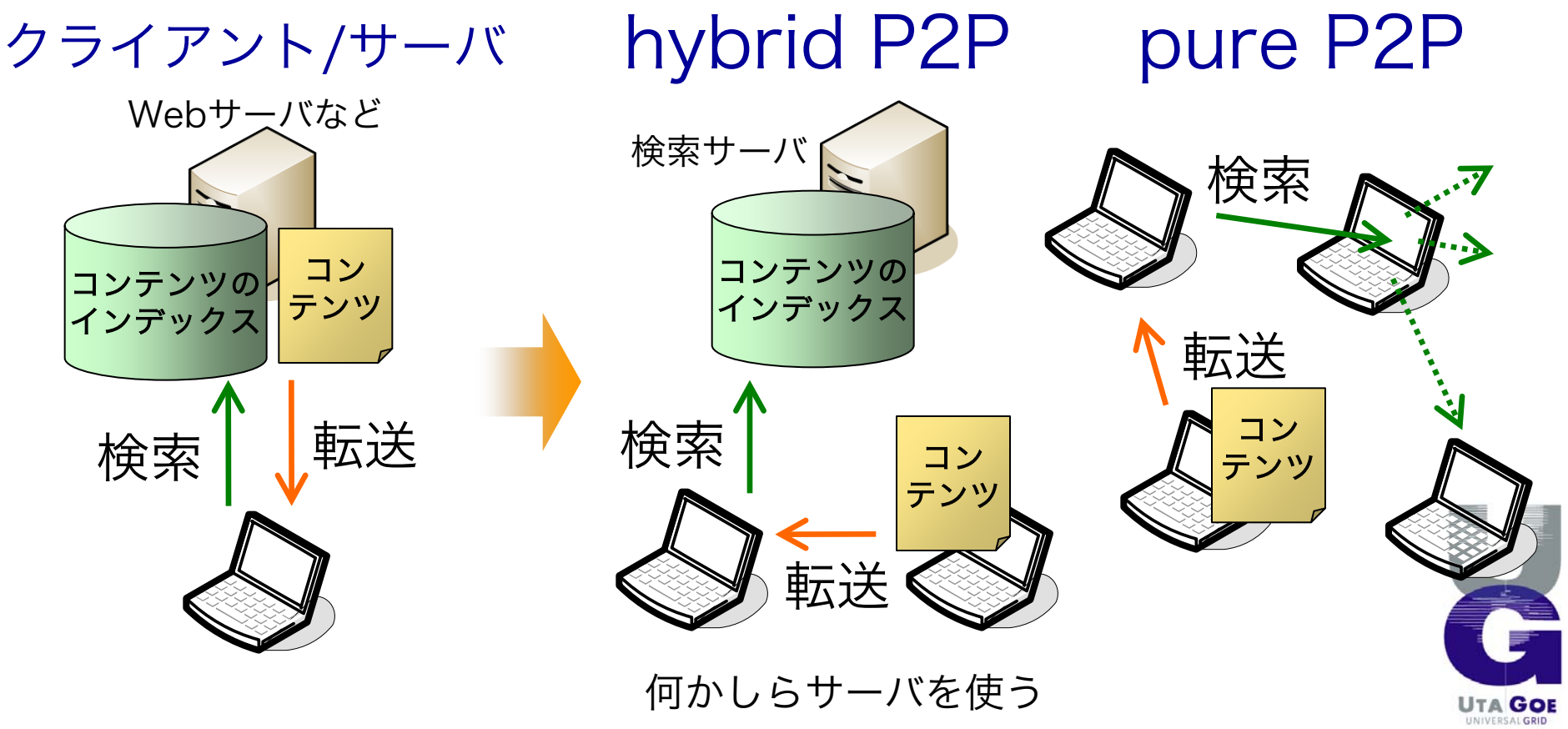
オーバレイネットワーク

peer-to-peer の展開

UTA GOE
UNIVERSAL GRID

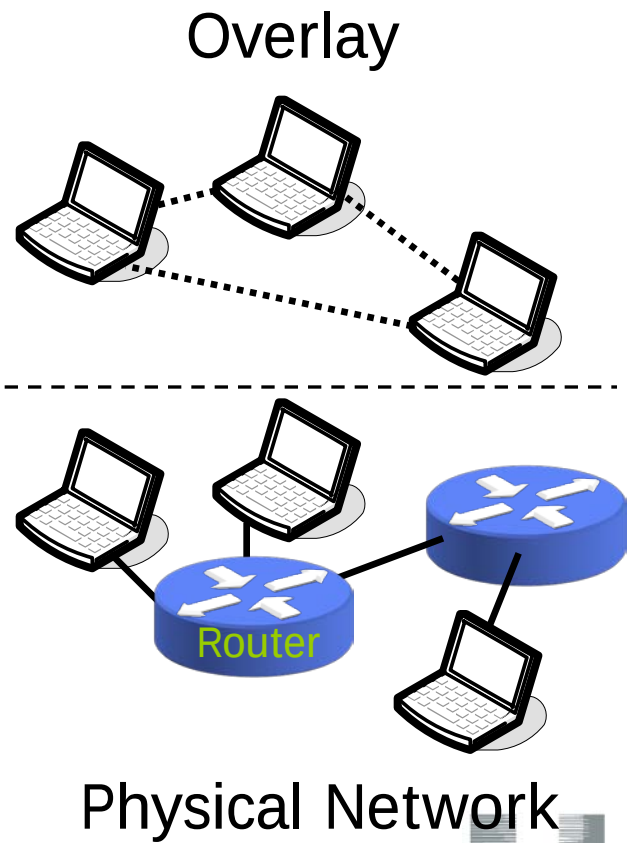
前置き: hybrid / pure P2P

- P2P ファイル共有ソフトの例



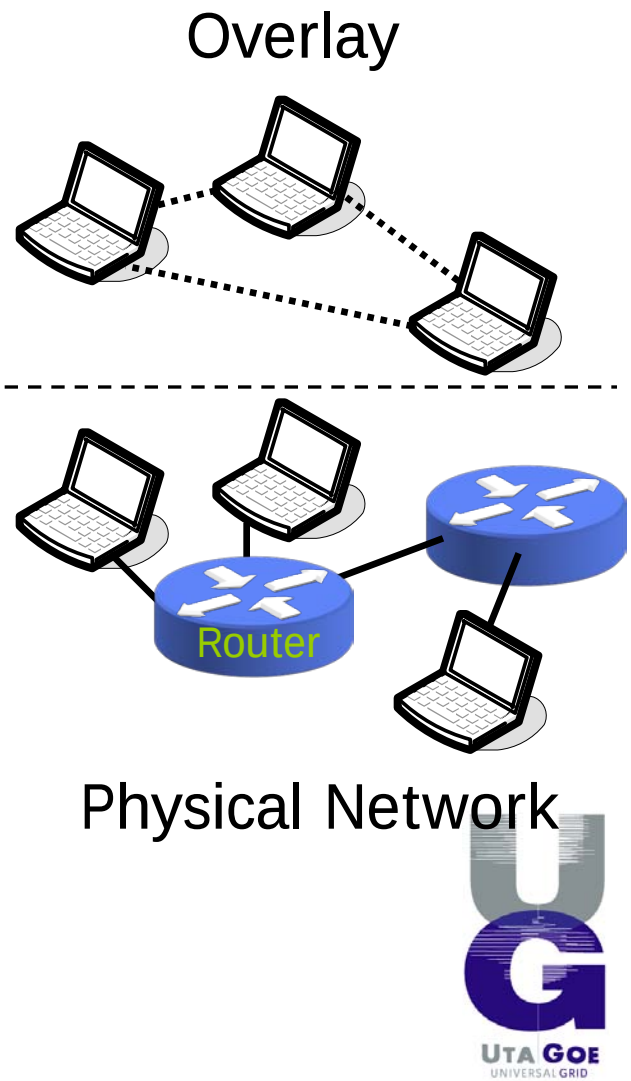
オーバレイ (overlay)

- pure P2P → 非集中 (decentralized)
- 集中サーバなしに何らかの機能を果たすため、アプリケーションレベルのネットワークを構成する。
  - 例: ファイル共有ソフトのネットワーク
    FastTrack, eDonkey2K, Gnutella, …
    ↑ ノード数 100万以上
  - 機能： 発見, マルチキャスト, メッセージ配信, …



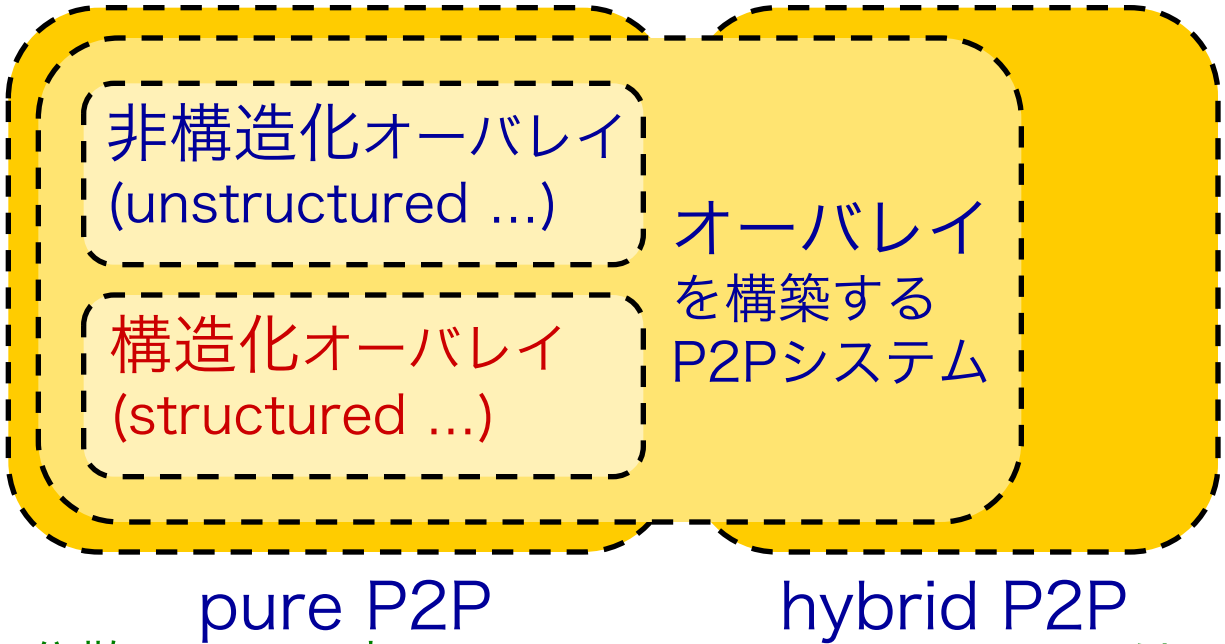
オーバレイ (overlay)

- そのトポロジは下位ネットワーク (インターネット) の物理的トポロジとは独立
  - オーバレイネットワーク or ネットワークオーバレイと呼ばれる。
  - 注）オーバレイ：ネットワークに被せられた別トポロジのネットワーク一般を指す。P2P 関係だけの用語ではない。



オーバレイとP2Pの関係

- pure P2P → 要オーバレイ
  hybrid P2P → オーバレイを構築したりしなかったり
  - アプリケーション層マルチキャスト (ALM) は、
    hybrid P2Pであってもオーバレイを構築する。
- pure P2Pシステムが構築するオーバレイには、
  非構造化 (unstructured) と構造化 (structured) の２種類がある。



- 分散ハッシュ表 (Distributed Hashtable, DHT) は、
  構造化オーバレイの応用のひとつ。

Unstructured と Structured

- Unstructured オーバレイ
  - 例： Gnutella ネットワーク, Winnyネットワーク
  - 誰を隣接ノードとするか、トポロジに制約がない。
  - 存在するオブジェクトは、発見できる可能性がある。
  - 一般に、効率は良くないが、柔軟な検索が可能。

- Structured オーバレイ
  - 例： DHT（分散ハッシュ表）のネットワーク
  - 誰を隣接ノードとするか、トポロジに制約がある。
  - 存在するオブジェクトは、（たいてい）発見できる。
  - 一般に、効率は良いが、柔軟な検索が苦手。

Unstructured と Structured

- 当初、pure P2P のファイル共有ネットワークは unstructured オーバレイだった。
  - Gnutellaネットワーク, Winnyネットワーク
- 最近は、structured オーバレイの応用も始まっている。
  - eDonkey2k, eMule の Kadネットワーク
  - BitTorrent、Azureus のトラッカーなし動作

具体例

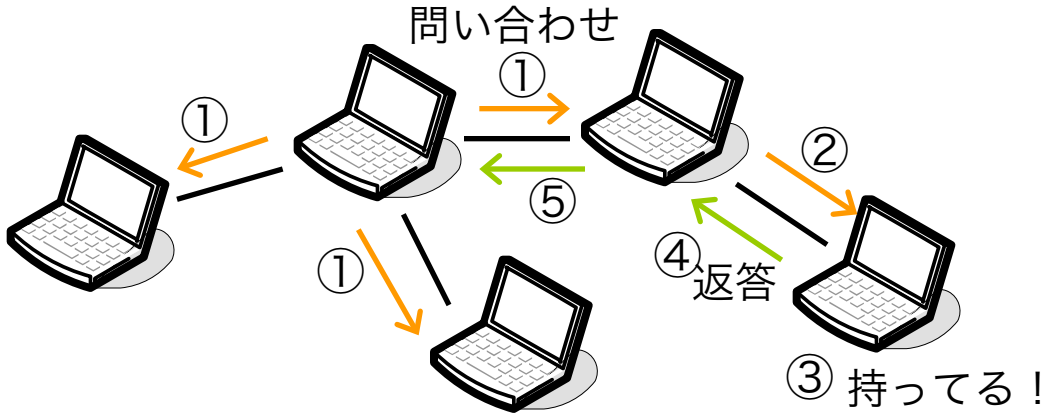
- Unstructured オーバレイ
  - Gnutella プロトコル (0.4)

- Structured オーバレイ（アルゴリズム）
  - Chord
  - Pastry
  - Kademlia

Unstructuredオーバレイの例:

Gnutella プロトコル (0.4)

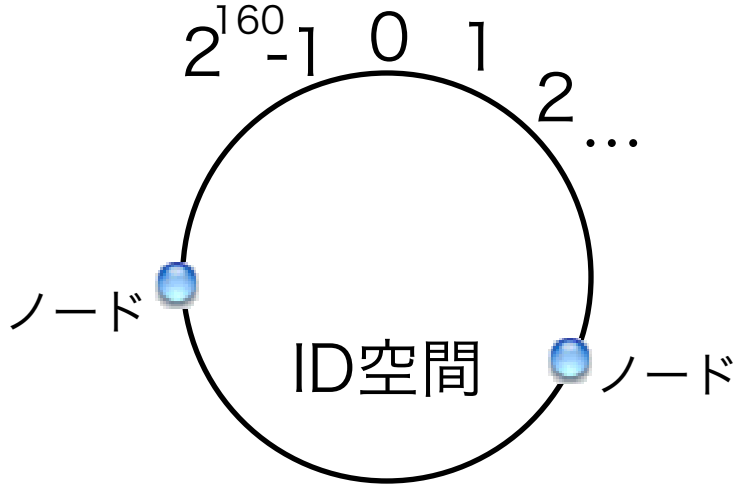
- 各ノードは、一定数の隣接ノード (neighbor) を持つ。
  - どのノードを隣接ノードとするか、および、隣接ノードの数はアプリ依存。例えば 4。
- 検索時、各ノードは問い合わせを全隣接ノードに転送する → flooding
  - TTL（time-to-live, 最大ホップ数）は 7。
  - 返答は、転送経路に沿って返される。

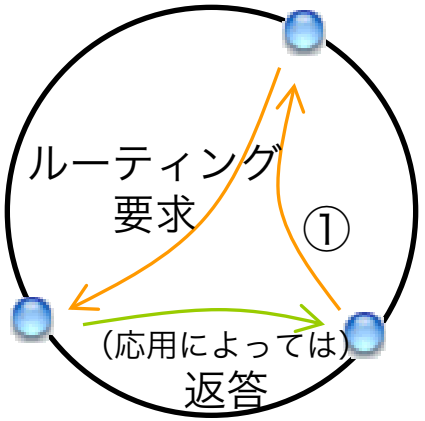


- Gnutella プロトコル 0.6 は、super-peers の概念を採り入れた。

Structured オーバレイの基本

- ノード（計算機）とオブジェクトの両方にIDが振られる。
  - IDはたいてい整数値。160ビットだったり 128ビットだったり。
  - オブジェクト：任意の文字列だったり、ファイルだったり、プロセスだったり。
- ノードは、ID空間中のある範囲を受け持つ。
  - だいたい、ノードのIDと数値的に近い範囲を受け持つ。
- IDを宛先としてルーティングが行われ、その行き着く先は受け持ちノードとなる。

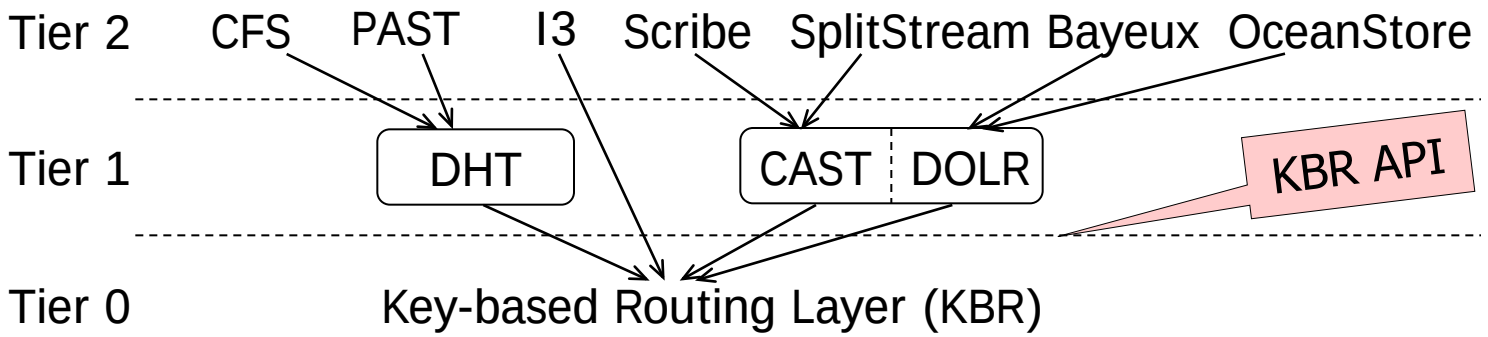




Structured オーバレイの基本

- 肝はルーティング（アルゴリズム）
- 資源にかかる負担が、ノード数 n として O(log n)。
  - ルーティング時のメッセージ数など。
  - cf. unstructured オーバレイ上の flooding
- 様々なアルゴリズムが提案されてきた。
  - CAN, Chord, Tapestry, Pastry, Kademlia, Koorde, Broose, Accordion, ORDI, DKS, D2B, Symphony, Viceroy, …

- Structured オーバレイ上に、いろいろなサービスが載る。
  - 分散ハッシュ表 (DHT), マルチキャスト, メッセージ配送, …



Cited from [Dabek03]

分散ハッシュ表 (DHT)

- Structured オーバレイ上に載るひとつのサービス。pure P2P な DB。ハッシュ表。
  - put (key, value)
  - get (key)
  - キー・値ペアの削除

- 処理
  - put: keyをキーとしてルーティングを行い、目的ノードにキー・値ペアを保持させる。
  - get: keyをキーとしてルーティングを行い、目的ノードが保持している key に対応する値をもらう。

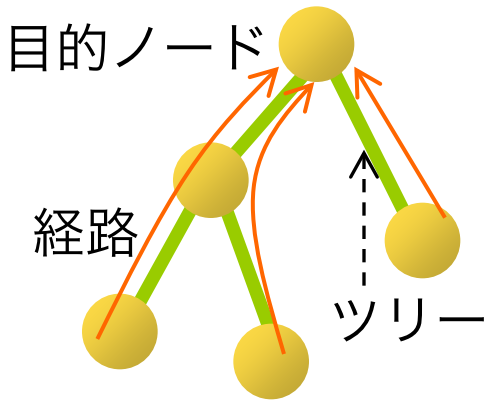
  - 注) key が ID になっていない場合、ハッシュ値を求めて ID にしておく。例えば（暗号学的ハッシュ関数） SHA1 を通すと、160 ビットの ID を得られる。

DHT の応用

- もろもろの名前解決や位置解決
  - ホスト名 → IPアドレス, 名前 → 電話番号, 曲名 → 楽曲ファイルやそのURL などなど。
  - DNS を作ってみました、という研究もある。
  - IP電話ソフト Skype のコンタクトリスト管理に DHT のようなアルゴリズムが使われているとのこと。
    「DHTである」という証拠はない？

Structured オーバレイ上の アプリケーションレベル マルチキャスト (ALM)

- ある ID に対して複数のノードからルーティングすると、それら経路の集合はツリーを成す。
  これを配送木として利用する。
- チャネルが ID で表されるので、多チャネル。
  ≠ ブロードキャスト
- 処理
  - あるチャネルにsubscribeするノードは、チャネルの識別子をキーとしてルーティングをする。
  - 経路上のノードは、1ホップ手前と1ホップ先のノードを、それぞれツリーの子と親として記憶する。
  - 各ノードが、ツリー上の親子関係に沿ってトラフィックを転送する。



UTA GOE UNIVERSAL GRID

マルチキャストの応用

- 同報通信を行うもろもろの応用
  - グループチャット・音声 / ビデオ会議
  - 放送
  - VPN
    - L2 のブロードキャストに使える。
    - 例??? x-kad: Kademliaというアルゴリズムを応用した VPN
  - 分散処理
    - コードの配布やプロセッサ間同期、ブロードキャスト
- 当然、エニーキャストも載る。
  - グループ中のノードどれかにメッセージ配信
  - 応用：サーバ群の負荷分散

UTA GOE UNIVERSAL GRID

Structured オーバレイの（ルーティング）アルゴリズム

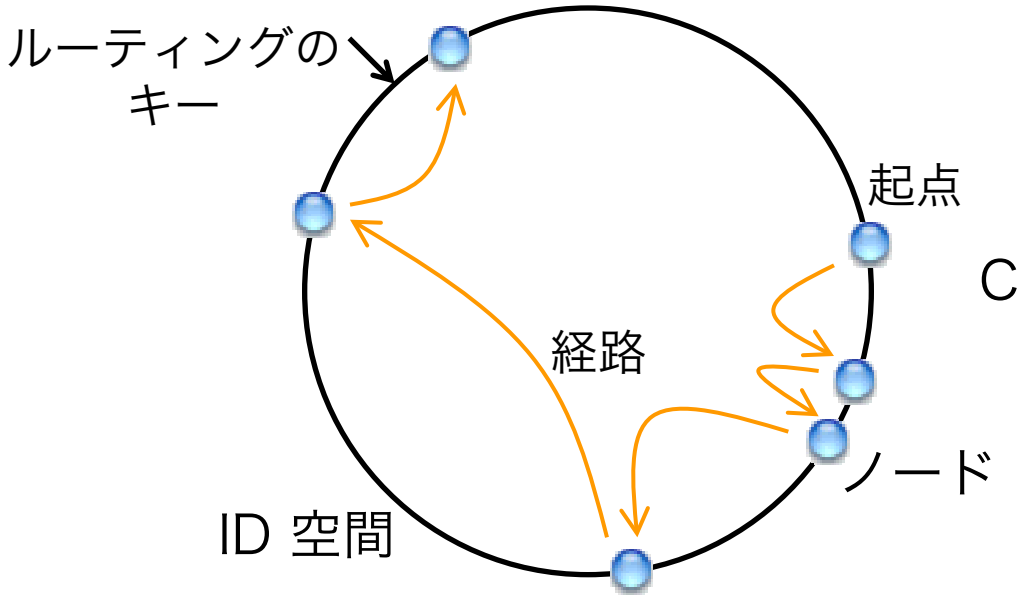
- 宛先 ID に（数値的に）より近い ID を持つノードに、要求をまわしていく。
  - いつかは最も近い ID を持つノードに辿り着く。

- 近づき方の、アルゴリズムごとの違い
  - Chord
    - ID 空間を時計まわりに近づいていく。
  - Pastry, Tapestry
    - ID を、上位桁から順に、b (例 4) ビット単位で揃えていく。
    - Pastry は、最後の 1ホップは違う方法で近づく。
  - Kademlia
    - ID を、上位桁から順に、1 ビット単位で揃えていく。

UTA GOE UNIVERSAL GRID

Chord

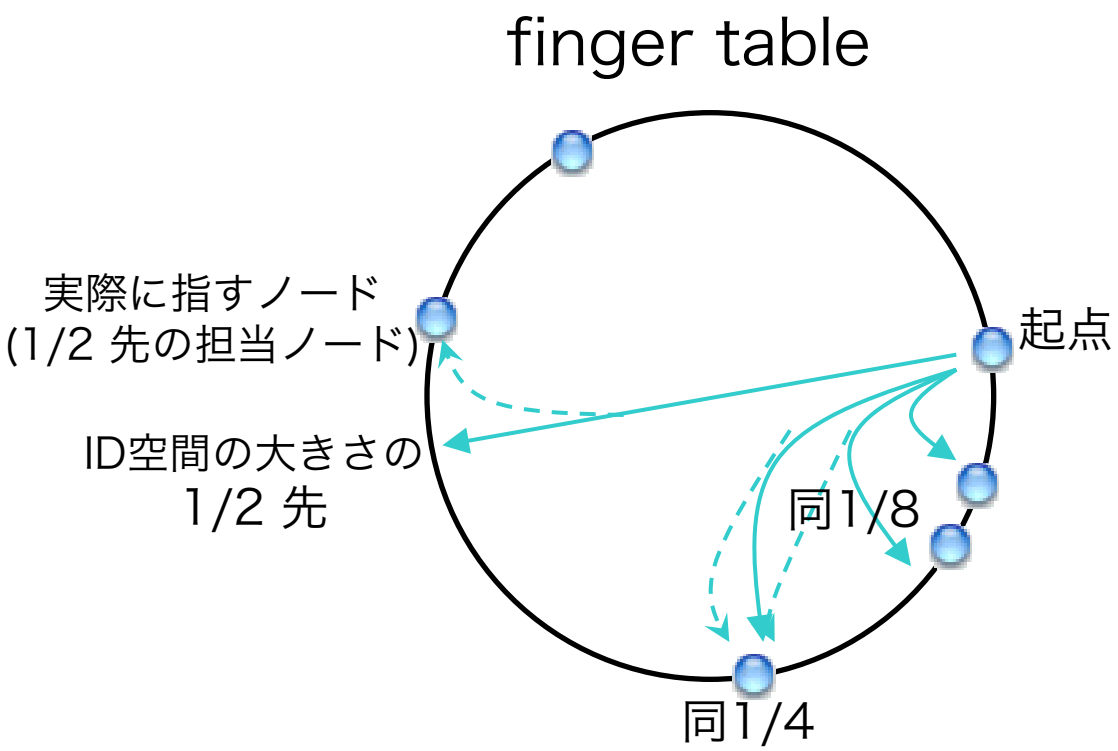
- Consistent Hashing 上の線形探索 + finger table
  - finger table: ショートカットリンク
- Consistent Hashing 上の線形探索
  - ID空間を時計まわりに進む。
    - 各ノードは、次のノード (successor) へのリンクを持つ。
  - 下図の場合、5 ホップ
  - オーバレイ上のノード数を n とすると、ホップ数は O(n)。残念！

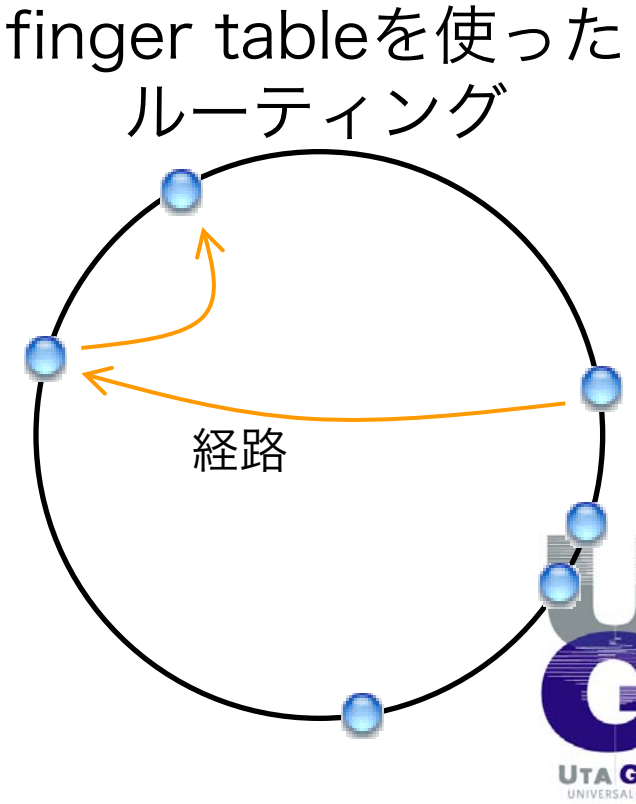


Consistent Hashing上の
線形探索

Chord

- finger table
  - ショートカットのためのリンク
  - 下図のルーティングの場合、2ホップ。
  - ホップ数 O(log n)。万歳！





Pastry, Tapestry

- Plaxtonらの方法 を用いる
  - 上位桁から bビット単位で揃えていく。
- 例
  - b = 4 とする → 16進数の一桁ごとに揃えていく。
  - ルーティングの宛先 ID: 437A (16進数) とする。
    - 本当は、160ビット (Tapestry) や 128ビット (Pastry)。
  - 手順
    - ID が 4XXX であるノードにまわす。
    - ID が 43XX であるノードにまわす。
    - ID が 437X であるノードにまわす。
    - …
  - そういうIDを持つノードが経路表に載っていなければ、次善のノードにまわす。
    - 43XX がなければ、44XX にまわす。

UTA GOE UNIVERSAL GRID

Plaxtonらの方法の経路表

- 例: ID が 10DF であるノードが持つ経路表
  - この場合、サイズは 16 列 x 4 行。
  - n 行目には、n – 1 桁目まで自分と ID が一致するノードが入る。
  - ID とコンタクト先（IPアドレス等）の組が入る。空の場合もある。

	0	*1*		…	*D*	*E*	*F*
1桁目		自身			DXXX	EXXX	FXXX
2桁目	自身				1DXX	1EXX	1FXX
3桁目					自身	10EX	10FX
4桁目		10D1			10DD	10DE	自身

Pastry と Tapestry の違い

- Pastry
  - 前頁の経路表に加えて、ID が数値的に近いノードの集合 leaf set も保持する。
    - 8ノードだったり 32ノードだったり。
    - 次ホップを決める際、まずは leaf set を見る。
- 違い
  - エントリが空だった場合の、次善のノードの決め方。
    - Pastry: とにかく ID が数値的に近いノード。
    - Tapestry: 表を右に見ていく。自身にぶつかったら、下段に降りる。
  - おまけ：広域分散ストレージ OceanStore のルーティングアルゴリズムは、Tapestry + bloom filter

オーバレイへの参加・経路表の保守

- 参加の際は、自身および他ノードの経路表を更新（作成）する。
- 新ノードの参加や既存ノードの脱退に応じて、各ノードは経路表を保守する。

- ただのルーティングよりは複雑。
  - アルゴリズムごとの詳細は、ここでは割愛。

オーバレイへの参加・経路表の保守

- 各アルゴリズムが、プロトコル・手順を規定している。
  - 参加の際は、まず、自ノードの ID を宛先としてルーティングする。
  - 参加の時点で、関係する全ノードの経路表を更新し切ってしまうか否か：
    - 更新し切る
      - Chord （論文Fig.6のアルゴリズム）, Pastry, Tapestry
    - 保守によって、じょじょに更新される
      - Chord（通常のアルゴリズム）, Kademlia
  - 保守
    - 定期的に通信して行う
      - Chord, Pastry, Tapestry
    - 定期的な保守が不要
      - Kademlia: ルーティング目的の普段の通信を手がかりに保守する。

オーバレイについての発展的な話題

- ネットワーク的な近接性の扱い
- 非均質性の扱い
- Complex network の工学応用？
- 研究開発ツール・ソフトウェア

UTA GOE
UNIVERSAL GRID

ネットワーク的な近接性の扱い

近接性: proximity

- 「素早く見つけたい」「速くダウンロードしたい」「途切れなく視聴したい」「確実に届けたい」...

- ネットワーク的に近い方が
  早い / 速い / 確実。
  だから、近い相手を選びたい。

- 近いとは
  - 遅延が小さい and/or 帯域幅が広い
  - ID の数値的な近接と混同しないように注意する。

UTA GOE UNIVERSAL GRID

ネットワーク的な近接性の扱い

- Unstructured オーバレイの方が得意だと言われている。
  - 隣接ノードや通信相手を選ぶ際の制約がないから。
- Structured オーバレイに採り入れることも可能。
  - 採り入れ方の分類
    - Proximity Neighbor Selection (PNS)
    - Proximity Route Selection (PRS)
    - Proximity Identifier Selection (PIS)
  - アルゴリズムによって、それぞれとの親和性が異なる。
    - Pastry: 前述の表に加えて、neighborhood setを持つ。
      最初の論文では「実装してない」とのこと。
    - Tapestry: 経路表を複数枚持つことで、PNS & PRS が可能。
- Iterative と Recursive ルーティングでは、近接性の影響が異なる点にも注意。

近さを知る方法

- 基本的には、計測。
  - 遅延 & 帯域幅

- 推定する手法も提案されている。
  - 一部分の計測結果をもとに、ノード間距離を推定する。
  - Vivaldy
    - ２次元空間＋高さにノードをマップ
  - Lighthouse

非均質性の扱い

非均質性: heterogeneity
⇔ homo-…

- オーバレイに参加するノードは様々。
  - 処理能力
  - ストレージの大きさ
  - ネットワーク帯域幅
    - 例: xDSL の上りは狭い。P2P で放送をする際に…
  - 通信の信頼性
  - ファイアウォール内 / 外
  - 継続して稼動する時間の長さ
    - 例: 常時電源 ON, たまに起動, …
- 全ノードにまったく同じ処理を求めると、最も低い / 狭い / 弱いところに全体の性能が制約されかねない。

UTA GOE UNIVERSAL GRID

非均質性への対処方法

- スーパーノード / ピアの導入
  - 限られたノードでオーバレイを構成する。
    他のノードはスーパーノードの提供するサービスを利用する。
  - cf.
    - ファイル共有ソフト Kazaa の Gnutella プロトコル 0.6
    - JXTA 2.x
  - いわば、0 か 1

- （全ノードの）適応 (adaptation)
  - 能力に応じて、隣接ノード数などを変える。
    - cf. HeteroPastry
  - いわば、連続量

  - Unstructured オーバレイへの導入の方が素直ではある。

UTA GOE UNIVERSAL GRID

Complex Network の工学応用？

- Complex Network
  - regular グラフでも random グラフでもないグラフ
  - Small World Network
    - Low diameter
      - Diameter (直径): 任意の２ノード間の距離の平均
    - High clustering
      - clustering coefficient (クラスタリング係数): あるノードの隣接ノード群がつながっている割合
    - cf. Milgram の手紙転送実験
  - Scale-free Network
    - 次数（枝数）が k であるようなノードの出現頻度が、k^-γ に比例。⇒ べき乗則
    - つまり、ハブが存在する。
    - cf. ウェブページのリンク数

UTA GOE UNIVERSAL GRID

Complex Network の工学応用？

- Complex Network の性質
  - 次数が小さいのに直径（≒ ホップ数）が小さい。

- ネットワーク分析の分野で注目されているが、作る際（工学）にも活かせるだろう。
  - 例
    - 流通ネットワークの構築：ハブの設置
    - Structured オーバレイに適用：Symphony
      - ショートカットパスの数を、ID 空間サイズに依らず、一定とする。⇔ Chord
      - それでもホップ数はわりと小さいまま。

UTA GOE UNIVERSAL GRID

Complex Network

- Gnutella ネットワークは
  以前から Scale-free Network ？
  - 多分、super-peers 導入前。それにもかかわらず。

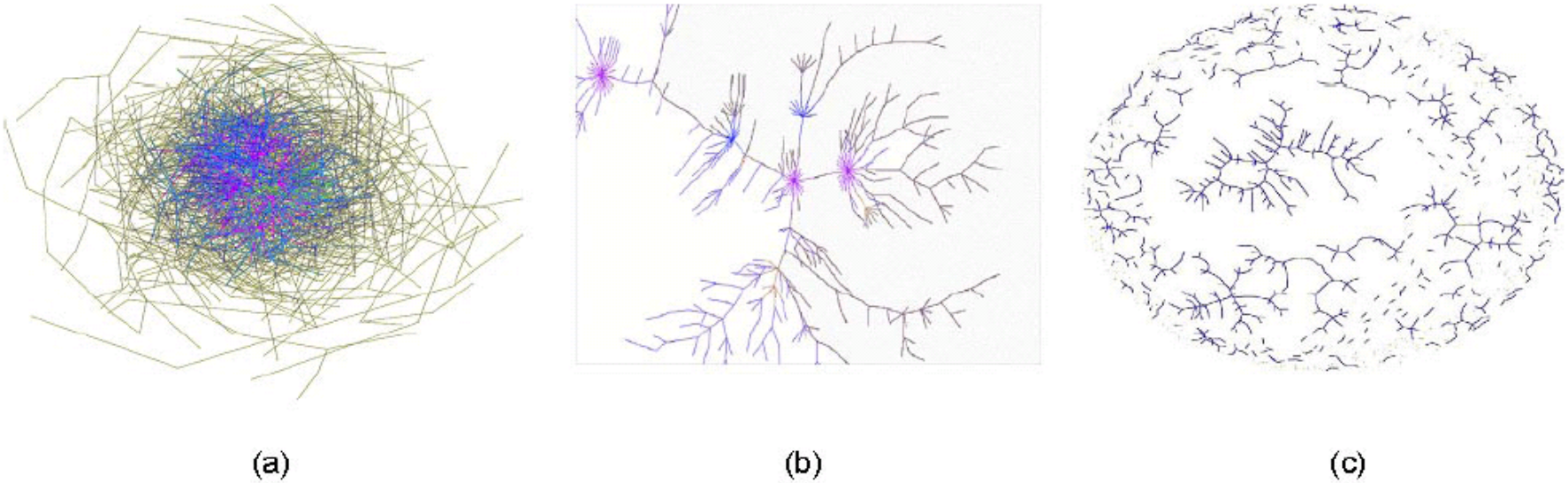


Figure 13. Left: Topology of the Gnutella network as of February 16, 2001 (1771 peers); Middle: Topology of the Gnutella network after a random 30% of the nodes are removed; Right: Topology of the Gnutella network after the highest-degree 4% of the nodes are removed.

Stefan Saroiu, P. Krishna Gummadi and Steven D. Gribble, “A Measurement Study of Peer-to-Peer File Sharing Systems”, Proc. MMCN’02, 2002. の図13

UTA GOE UNIVERSAL GRID

研究開発ツール・ソフトウェア

- p2psim
  - ルーティング方式のシミュレータ
  - Structured オーバレイのアルゴリズムを多数提供
    - Chord, Accordion, Koorde, Kelips, Tapestry, Kademlia
- MACEDON
  - オーバレイアルゴリズムの研究プラットフォーム
    - structured / unstructured 問わず。
  - C / C++ に似た専用言語でアルゴリズムを記述し、それを MACEDONのコンパイラで処理し、動作する C++ コードを得る。⇒ コード量が少なくて済んでいる。
  - DHT アルゴリズムとして Chord, Pastry を提供
    - 本来 160 / 128 ビットであるID が、なぜか 32 ビット。
- Mace
  - MACEDON の後継プロジェクト

UTA GOE UNIVERSAL GRID

研究開発ツール・ソフトウェア

- DHT のライブラリ
  - Bamboo DHT
    - PlanetLab 上で運用されている。⇒ OpenDHTプロジェクト
    - アルゴリズムは Pastry ベース。
    - Javaで書かれている。
  - Chimera, Tapestry
    - Tapestry の研究グループ自身による実装。
    - Tapestry は Java で書かれている。
    - Chimera は、Tapestry のコンパクトな C 実装。
  - FreePastry
    - Java で書かれている。
  - Khashmir
    - Python で書かれた Kademlia 実装。
    - BitTorrent （本家クライアント）が使っている。
  - Kenosis, SharkyPy, DKS, OPeN, …

研究開発の実際

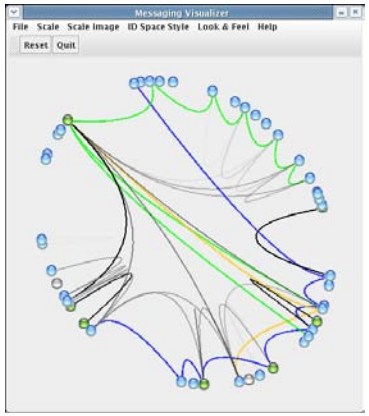
Overlay Weaver の場合

UTA GOE
UNIVERSAL GRID

Overlay Weaver

- DHTライブラリ
  - 記述言語は Java。
  - ２万ステップ程度。
    - ３万数千行。
  - Apache License 2.0。
    - いろいろな目的に使いやすい？

可視化ツール



- 特徴
  - 本当は DHT だけじゃない。
    アプリケーション層マルチキャストも。
  - ルーティングアルゴリズムを差し替え可能：
    Chord, Kademlia, Koorde, Pastry, Tapestry
  - コーディングなしで試用や実験が可能。
    - サンプルツール (DHTシェル等) を使った運用
    - エミュレータを使った実験: メッセージ数やホップ数の計測
  - 動作状態を可視化して楽しめる。デモできる。
  - XML-RPC 経由で DHT を使える。
    - Bamboo, OpenDHT と同じプロトコル → 同じクライアントを使える。

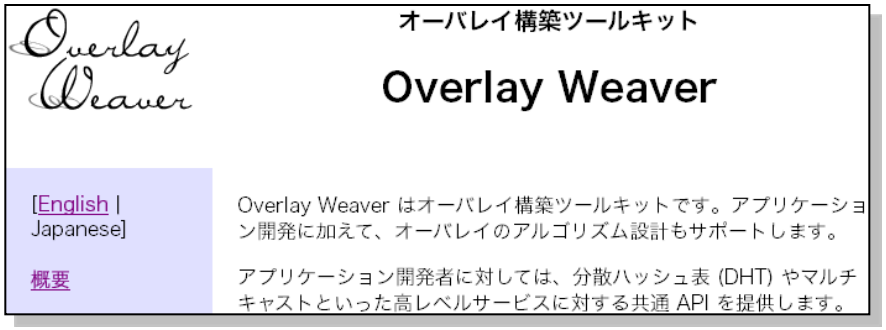
UTA GOE UNIVERSAL GRID

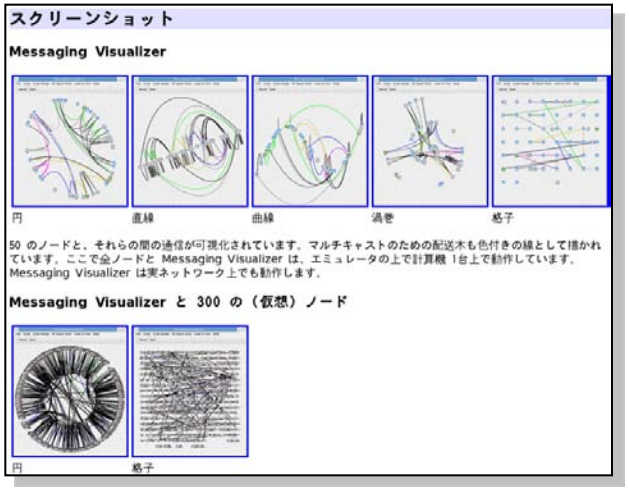
オープンソースソフトウェアとしての Overlay Weaver

- http://overlayweaver.sf.net/ (SourceForge)
  - 2006年１月17日、リリース。
  - Apache License 2.0

- 状況 (2007/1/6 21時)
  - ダウンロード 4,354件。
  - メーリングリスト登録数
    - 英語 55名, 日本語 69名
  - Mixi コミュニティのメンバ数 121名

- 想定ユーザ
  - アルゴリズム設計者 だけでなく
  - アプリケーション開発者
    - 例: RDF文書のP2Pストレージ by 的野 (産総研)
  - 卒論・修論での活用、大歓迎





ウェブサイト

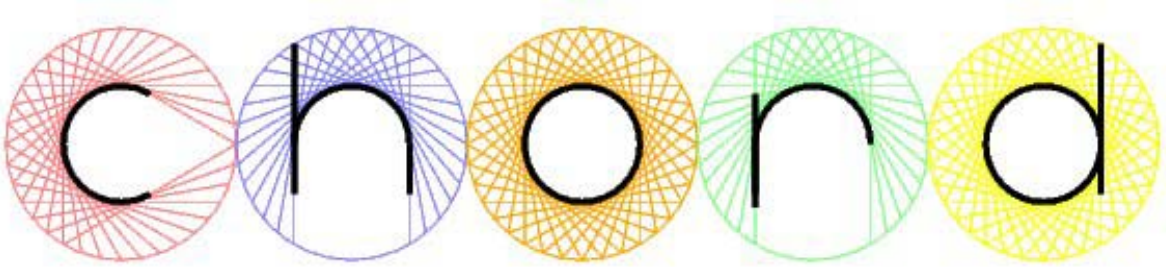
UTA GOE UNIVERSAL GRID

オーバレイ織り器

Overlay Weaver

- structured オーバレイ関係の名前には、糸を織るというアナロジが見られる。
  - Chord -- 弦
  - Tapestry -- 壁掛けの織物
  - Pastry -- 練り粉, パイの皮 ??
- Overlay Weaver
  -- (structured) オーバレイの織り器
  - Weave -- 織る, 編む
  - Weaver -- 織り手, 織工, 編む人

Weaver



Tapestryのロゴ

大規模環境は入手/アクセス困難

- ターゲットは、数千 〜 数百万台。
  - そんな実験環境は、ない。
  - なので、1台あたり複数ノードを担当させる。
    - 多重化: せいぜい、〜 数十ノード / 台
  - それでも、数十台 〜 数万台は使いたい。

- 買う / 借りる
  - ところにより、数百台あるでしょう。
    - TSUBAME, StarBED, AISTスーパークラスタ, …
      問題: 貸し出されてない, 賃料が高い, 商用利用の制限, …
    - Googleの中の人「申請すれば一万台(?)という環境を用意してもらえる。2,3ヶ月かかるけど。」
  - 当社: いったん 1X 台用意した…が、サービス用に切り売りして、消えていった。
    - マシンはともかく、設置場所, 消費電力が問題。

大規模環境は入手/アクセス困難

- 共有テストベッド: PlanetLab
  - 2台 供出すれば、約 700台を使える。
    VM 払い出し方式。
  - 企業からの参加は、最低でも $10,000 / 年。高い。
  - 知人「SIGCOMM の締切前は重くて使いものにならない」

- インターネットエミュレータ: ModelNet
  - PC 40台でたかだか 1,000ノードのエミュレーション
  - 構築が困難 (e.g. FreeBSD限定, …)
    - 知人「環境構築を外注したら、2社にギブアップされた」

UTA GOE UNIVERSAL GRID

大規模環境は入手/アクセス困難

- いきなりリリース！！
  - 魅力的な機能を持たせて、配布する。
    - 例: 映画のファイルが無料で手に入る！
    - 流行らせることの方が困難だっつーの…
  - Winny 方式。

- そうもいかないので、1台あたりなるべく多くのノード数を動かせるように作る。
  - Overlay Weaver: 1台で 4,000ノード
- 困難とは言っても、分散処理 / 高性能計算よりはマシ。
  - CPU, メモリなどを食うため、多重化が困難。

UTA GOE UNIVERSAL GRID

実験に時間がかかる

- Overlay Weaver で私がやってる実験
  - 注: データ / query 数をノード数に比例させてある。
  - 100ノード: 5分
  - 1,000ノード: 43分
  - 4,000ノード: 2時間半
    - 当初は 11時間かかってた。cf. SACSIS 論文
  - × 6アルゴリズム × 2 (iterative/recursiveルーティング)
    - × いろいろなパラメータ

- システムソフトウェアにそういう傾向がある？
  - JITコンパイラのときも、SPEC JVM98 の実行に数十分かかってた。
- このためだけにも PCクラスタが欲しい。
  - not 並列処理
- イベント駆動シミュレーションが可能なように作っておくと、実験時間を短縮できる。
  - ただし、時間・イベントを抽象化した (アルゴリズム) 記述が必要となり、記述性が下がる。スレッドを自由に作るようなことはできなくなる。e.g. Bamboo

大規模実験を想定した作り方

- 失敗例
  - 各ノードで GUI の操作が必要。
    - 過去にやりました… あとで CUI を用意。

- 考慮すべき点
  - ノードの制御
    - プログラムから集中的に行えるようにする。
      - 要コマンドインタフェース
  - ファイル
    - ノード数に比例した数の設定ファイル、ログファイルは避けたい。

UTA GOE UNIVERSAL GRID

10台で動作しても 100台では…

- 動作するとは限りません。
  - 100ノードで動作 → 1,000ノードで×
  - 1,000ノードで動作 → 4,000ノードで×

- 構造化オーバレイでの処理コストは
  たいてい O(log n), O(log^2 n) だが
  - ひそかにそうでない処理が紛れ込みがち。
    - 例: 定期的に○○ (通信)
      → 頻度・量がノード数やデータ量に比例 O(n) !
    - 論文に “periodic” を見たら、要注意。
    - 複数ノードが同期して、負荷が集中する危険もある。
      → 乱数でずらす。cf. インターネットの各種プロトコル

UTA GOE UNIVERSAL GRID

そもそも その規模で何が判るの？

- 首藤「4,000ノード実験ができますた！」
- 誰か
  「1,000 と 4,000 で判ることが違うの？」
  「何ノードなら充分なの？」
  「多ければいいの？」
  「10ノードじゃダメなの？」

- 対策
  - 性質と規模の関係を明らかにする。
    理論上は比較的容易だが…
    4,000 は 1,000 よりも本当に数百万に近いのか???
  - 現実の数字をベースに、何か言えるようにしておく。
    - 例: 国内の世帯数は 4,000万だからほげほげ
  - ツッコまれないようにする。
    - 規模の大きさなんかアピールしない。
    - 「4」とかいうキリの悪い数字は使わない。

UTA GOE UNIVERSAL GRID

パラメータが超多い

- パラメータ
  - 通信のタイムアウト, ルーティングのタイムアウト, コネクション/ソケットプールのサイズ, UDP hole punchingの頻度, ルーティングのTTL, 次ホップ候補の返答数, IDのビット数, 経路表からノードを落とすまでの通信失敗回数, 通信失敗を忘れるまでの時間,並行queryの並行性 (Kademlia), 経路表の大きさ (Kademlia), 経路表の更新頻度 (Chord,Koorde,Pastry), stabilize頻度 (Chord), successor listの長さ (Chord), digitを何ビットとするか (Koorde,Pastry,Tapestry), de Bruijnエッジの本数 (Koorde), leaf setサイズ (Pastry), 経路表エントリ上書きの確率 (Pastry,Tapestry), などなど
- 論文に書いてあれば、基本的にその通りにしている。他は、わりとエイヤッと。
- 現在研究しているchurn 対策関連だけでも、整数パラメータ 3つ、on/offパラメータ 1つがある。
  - 組み合わせの数はすぐに数十に。
    - Overlay Weaverの場合、さらに × 11。
    - 実験 1本あたり、1,000ノードなら 43分。
  - 要、ある種の探索。
  - ヒント?: ATLAS (自動チューニング), 並列化コンパイラにおける最適化手法適用順の探索, …

UTA GOE UNIVERSAL GRID

マルチスレッド,非同期処理は難しい

- 苦労した例
  - ロックの粒度が性能を大きく引っ張った。
    - ルーティングアルゴリズム Koorde で、DHT getの成功率が大きく低下。リリース前試験で判明。どの変更が影響したか不明。困った。
    - プロファイリングには表れない。
    - ソースコードの diff を元に、目視で原因を推測
      → なんとか発見して、解決。
    - まだ気づいていない問題が残っているかもしれず…
      - ロックを取得している個所すべてを目視確認？？？
  - UDP hole punching 処理でデッドロック。
    - ロック獲得順に起因する、ありがちなミス。
- 性能チューニングが困難。

UTA GOE UNIVERSAL GRID

異常系の奥が深い

- 通信タイムアウト: 失敗とみなすまでの時間
  - 長いと、利用者の待ち時間増加につながる。
    短いと、false positive が増える。
  - 現在は 3秒。
  - 賢い方法 in 論文 “Handling Churn in a DHT”
    - TCPのように、過去のRTTを元に決める。
      - ただし、recursiveルーティング限定。
    - 推測する。e.g. Vivaldi

- 通信失敗した相手を経路表から落とす条件
  - いきなり落とす? 2度失敗したら落とす? …
  - churn 時のルーティング遅延, 成功率に影響する。

UTA GOE UNIVERSAL GRID

日本に研究者コミュニティがない

- オーバレイ、peer-to-peer はどこに？
  - 情処 OS研究会
    - 関連発表はほとんどない。
  - 情処 ネットワーク生態学研究グループ
    - ネットワーク分析の人たち。
  - 情処 DSM研究会
    - NEC加藤さんはここで発表してるけど、
      中心メンバ/テーマは教育・計算機センター運営。
  - WIDEプロジェクトIDEON WG
    - すばらしい方々だけど、誰でも参加できるわけではない。
  - JXTAのメーリングリスト
    - 今年に入ってから27通だけ。首藤はJXTAから離れた。
  - P2P勉強会・DHT勉強会
    - 草の根。年に1,2回。知り合いベースのプログラム構成。
  - オーバレイネットワークシンポジウム (12月, 東大本郷)
    - by NICT, 東大 中尾先生, 阪大 宮原先生。招待ベース。政治的会合？
- 単に、研究してる人がいないだけかも。
- Google Groups に overlay-network-ja MLを作った。約 90名。

UTA GOE UNIVERSAL GRID

- P2P とは何か
  - 緩い定義から厳しい定義まで
  - Web 2.0 は P2P に期待されてきた社会的意味の具現化
- オーバレイ
  - unstructured / structured オーバレイ
- オーバレイについての発展的な話題
- 研究開発の実際
  - 大規模分散システムならではの悩みがある

UTA GOE
UNIVERSAL GRID

自己紹介 – 首藤

•略歴
- •1973年 産まれた
- •1998年 早稲田大学の助手
- •2001年 (独) 産業技術総合研究所
- •2006年 ウタゴエ

- ソフトウェアエンジニア
  - Peer-to-peer / Grid / 分散処理
    - スレッド移送 (1997〜)
    - Overlay Weaver – オーバレイ構築ミドルウェア (2005〜)
  - Grid とか
    - Access Grid (2001〜), 全世界カラオケ (2003)
      - 2003年 12月の丸山先生レクチャー ネタ
  - Java
    - shuJIT – Java JIT コンパイラ (1998〜)
  - 情報セキュリティ
    - 共通鍵暗号の強度解析 (1996)
    - AES 候補の高性能実装 (1998), AES 15候補の性能評価 (1999)
- 2006年 4月、ウタゴエに参加。

UTA GOE
UNIVERSAL GRID