Condorによる社内グリッドの構築

産業技術総合研究所
グリッド研究センター
中田秀基
hide-nakada@aist.go.jp

なぜ産総研のわたしがCondorの話を？

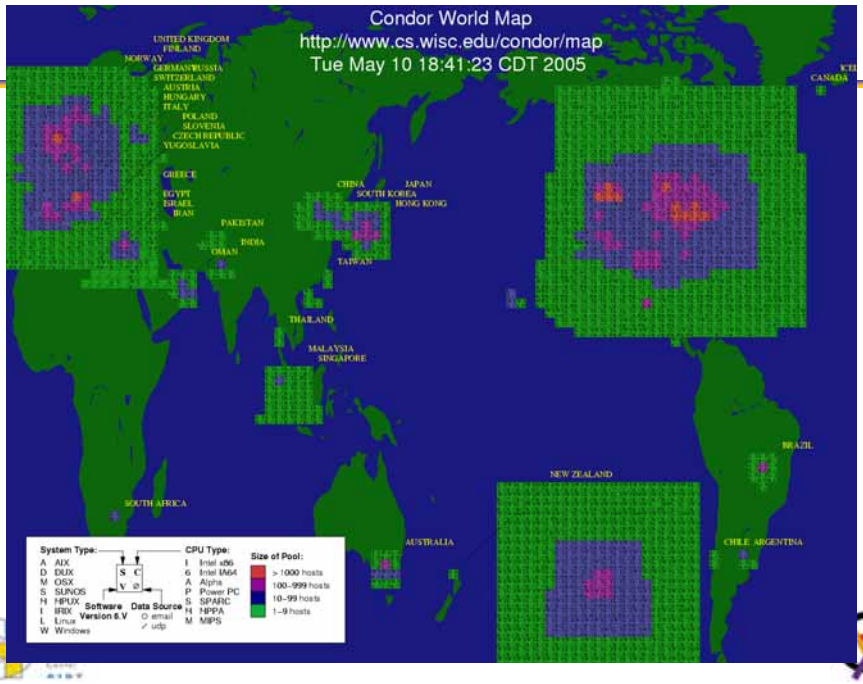
- 2004年 8月-12月 4ヶ月間 ウィスコンシンに滞在
  - Condor の機能拡張に従事
- 産総研発認定ベンチャーのベストシステムズが Condorのサポート提供を開始

ご質問

- 昨日のMiron教授のご講演をお聞きになりましたか？
- 昨日のMiron教授のご講演以前に「Condor」をご存知でしたか？

Condorとは

- 遊休計算機を利用して独立したジョブを大量に処理するシステム
  - c.f. Grid MP 、 SETI@HOME
- 並列ジョブにも対応したバッチキューイングシステム
  - c.f. PBS Professional, Sun One Grid Engine, LSF
- メタスケジューラ
  - c.f. Community Scheduler Framework (Platform computing)

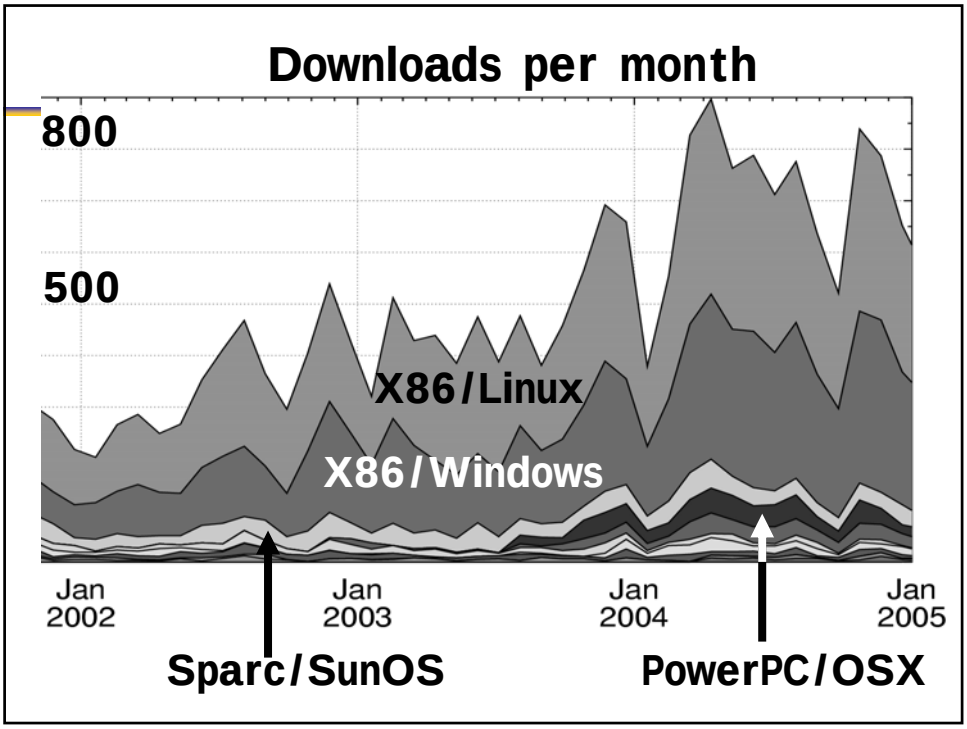


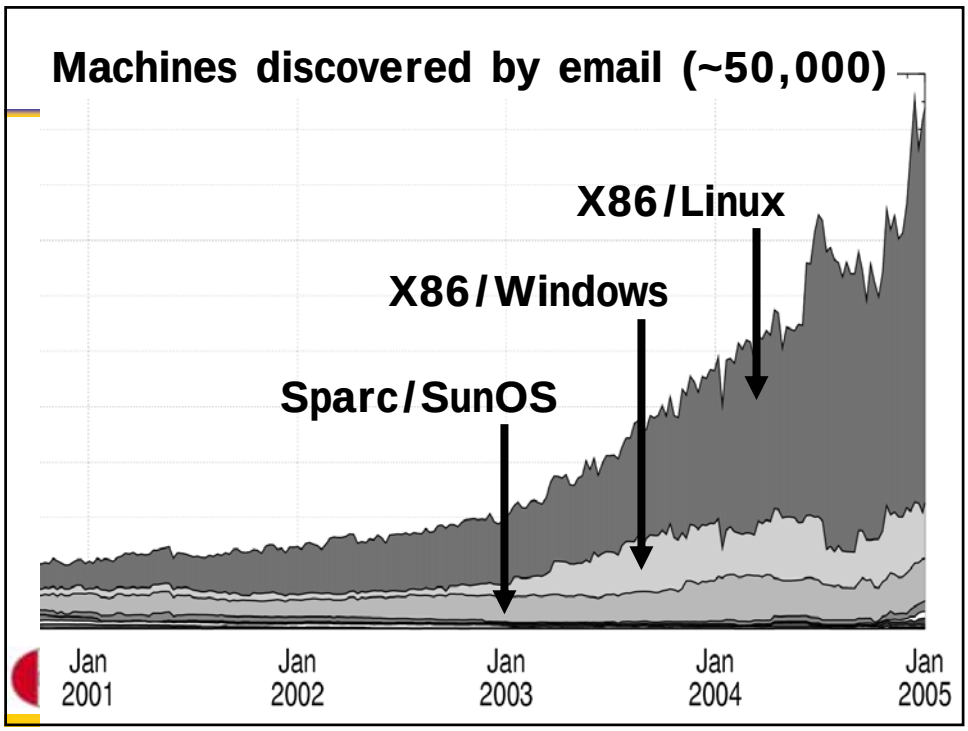
Condor の歴史

- 1983年 Prof. Miron Livny のPh.D Thesis
  - “The Study of Load Balancing Algorithms for Decentralized Distributed Processing Systems”
- 1985年 ウィスコンシン大学において、Condor Project 開始
- 1986年 最初のCondor システムが実装される
- 2005年で20周年(!)
- 予算: DoE、NASA、NIH、NSF、EU、INTEL、Micron、Microsoft、 UW Graduate School

サポートアーキテクチャ

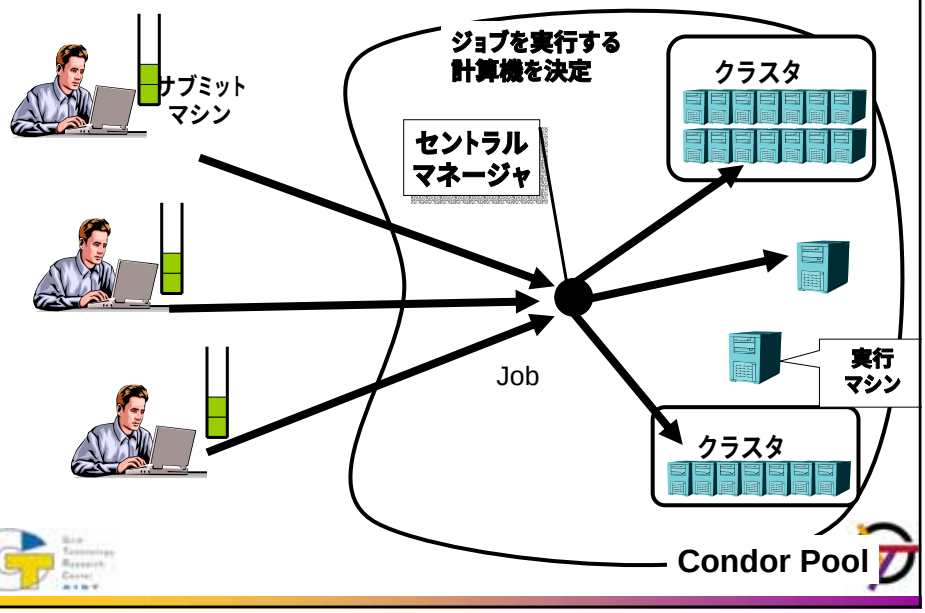
- **Sun SPARC Sun4m,Sun4c, Sun UltraSPARC**
  - Solaris 8, 9
- **Silicon Graphics MIPS (R5000, R8000, R10000)**
  - IRIX 6.5 (clipped)
- **Intel x86**
  - Red Hat Linux 7.1, 7.2, 7.3, 8.0, 9, Enterprise Linux 3, Debian Linux 3.1 (sarge)
  - Fedora Core 1, 2, 3
  - Windows 2000 Professional and Server, 2003 Server (Win NT 5.0), Windows XP Professional (Win NT 5.1) (clipped)
- **ALPHA**
  - Tru64 5.1 (clipped)
  - Red Hat Linux 7.1, 7.2, 7.3 (clipped)
- **PowerPC**
  - Macintosh OS X (clipped)
  - AIX 5.2 (clipped)
- **Itanium (IA64)**
  - Red Hat Linux 7.1, 7.2, 7.3 (clipped)
  - SuSE Linux Enterprise Server 8.1 (clipped)




Machines discovered by email (~50,000)
X86/Linux
X86/Windows
Sparc/SunOS
Jan 2001
Jan 2002
Jan 2003
Jan 2004
Jan 2005

Condorの概要

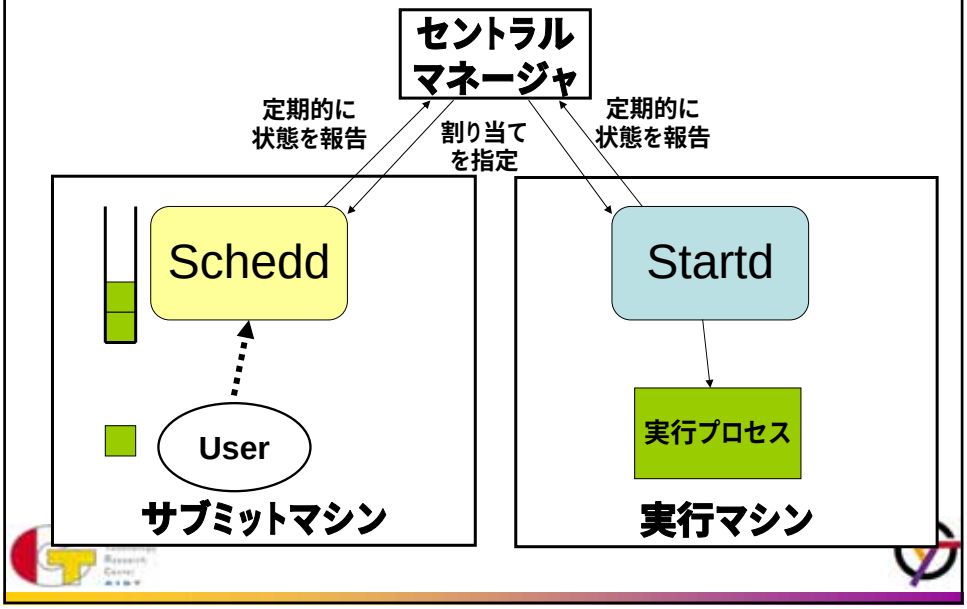
Condorの概要



Condorの特徴

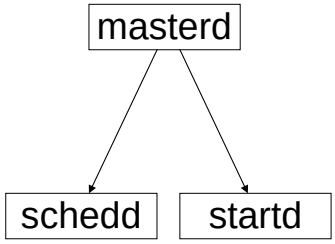
- スケーラブル
  - 1プールで1000ノード程度まで稼動実績あり
  - 複数のプールを組み合わせればさらに
- ステイブル
  - 自動復旧機構が組み込まれている
- フレキシブル
  - ClassAdとマッチメイキングによる柔軟なスケジューリング
  - ユーザのプライオリティによるフェアシェア

Condorの構造



スタビリティ

- マスターデーモン
  - 他のデーモン群を起動・監視
  - 落ちていたら再起動
  - 単機能なのでこのデーモンが落ちることはほとんどない

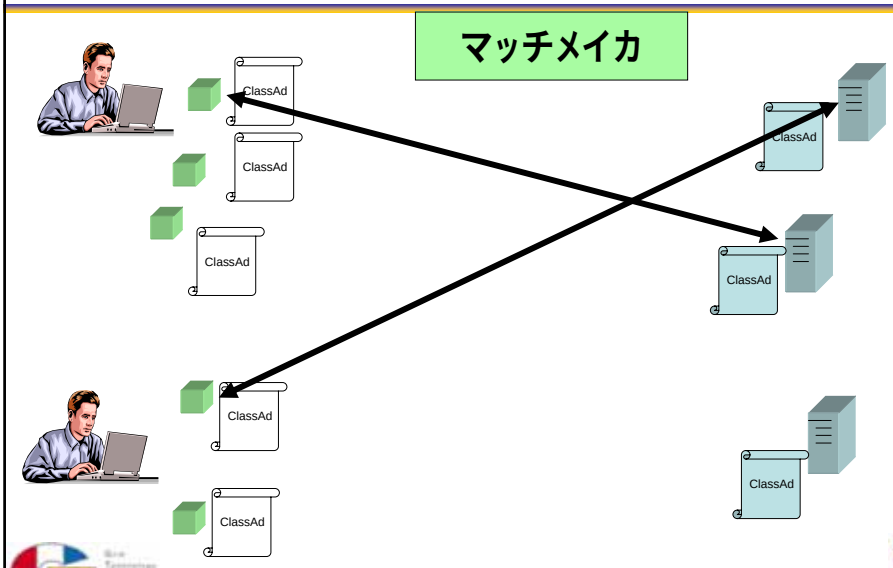


Condor ClassAd

- **Condorでやり取りする情報の表現形式**
  - 汎用性、拡張性に富む
  - サブミットマシン、実行マシンの両方が利用
- **基本的には属性名と属性値の集合**
  - 比較式を定義することも可能
- **セントラルマネージャはサブミットマシン、実行マシンから得たClassAdを用いてマッチメイキングを行う**

```
MyType = "Job"
TargetType = "Machine"
ClusterId = 292
User = "nakada@a02.aist.go.jp"
In = "/dev/null"
Out = "A.out"
Err = "/dev/null"
```

マッチメイキング



開発体制とサポート

Condor の開発体制

- **フリーソフトウェアだが、オープンソースではない**
  - 配布はバイナリ配布のみ
- **大学発のソフトウェアではあるが、学生が作っているわけではない**
  - 開発スタッフの大半は有給のエンジニア
  - 「研究」ではなく「開発」
  - その費用は主にさまざまなプロジェクトの予算から得られている
- **毎夜のビルドとリグレッションテスト**
  - すべての対応アーキテクチャマシンからなるCondorプールがある
  - Condor自身を用いてビルド・テストを行っている

Condorの開発体制 (2)

- テスト、ビルド専門のスタッフが数名(!)
- ドキュメント遷任のスタッフも！

サポート

- 2つのメイリングリスト
  - condor_admin@cs.wisc.edu
    - 開発者によるサポート
  - condor_users@cs.wisc.edu
    - ユーザメイリングリスト
- Condor チームと契約してサポートを購入することが可能
  - インストール、設定、アプリケーションの開発補助
  - カスタマイズの依頼も可能
- 大学のチームのサポートではいやだという向きには。。
  - 国内ではベストシステムズ(株)によるサポートが購入可能

Condor Week

- Condorのユーザーミーティング@マディソン
- 2000年からほぼ毎年
  - 参加者100人程度
- 2004年は マディソンのほかに英国エディンバラでも UK Condor Weekが開催
  - 200人弱？

導入事例

- 遊休計算機の有効利用
- 専用クラスタの利用
- 並列ジョブスケジューラとしてのCondor
- メタスケジューラとしてのCondor

Condorが有効な場面

- 比較的小さいジョブが沢山実行したい
  - 夜間使用されていないPCやWSがある
  - 専用のクラスタがある
- 1日ー1週間かかるようなジョブをいくつか実行したい
  - 計算機関占有できる計算機がない
- MPIなどで書かれた並列ジョブを実行したい
  - クラスタを使用
- PBSなどで管理された複数のクラスタを一括管理して有効利用したい
- 遠隔地のGlobusなどで管理されている複数のクラスタで大量のジョブを分散実行したい

Condorによる遊休計算機の有効利用

- Condorの開発開始時の主目的
- キーボードのアイドル時間、CPUのロードアベレージ、時刻などから遊休状態を認識
  - スクリーンセーバとして実装するよりも精度の高い認識が可能
- 遊休状態が終了したらジョブを追い出す
  - ユーザが戻ってきた
  - 始業時間になった
  - 別の方法でジョブが投入された

"The philosophy here is that we would like to encourage you to use as many cycles as possible and to do research projects that can run for weeks or months. But we want to protect owners, whether or not they are heavy users."

導入事例：Micron

- 半導体ベンダ
  - DRAMのシェアは世界で第二位
  - デジカメ用のCMOSイメージャ
  - フラッシュメモリ
- 大量のWindowsマシンを使用
- 半導体設計のためのシミュレーション、テスト等に使用
- CMOSイメージャのテスト

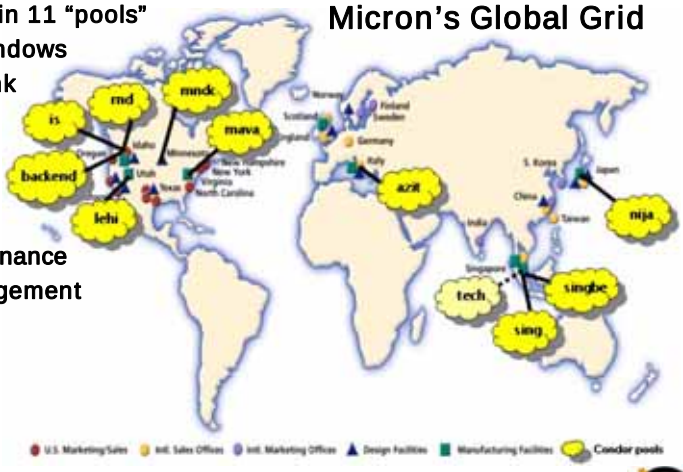
Micron のグリッド

10k+ processors in 11 “pools”
Linux, Solaris, Windows
~29th Top500 Rank
4.2 TeraFLOPS

Built in-house
Open Source Grid
Centralized governance
Distributed management

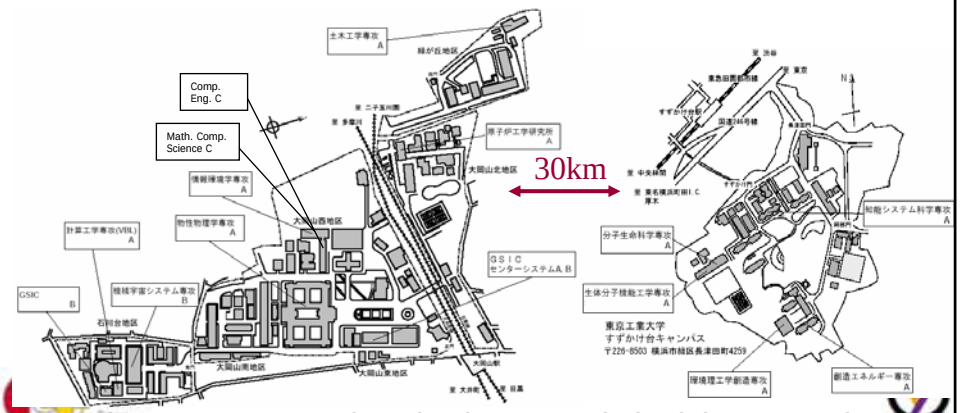
16+ applications
Self developed

Micron’s Global Grid



導入事例：東工大 Titech Grid

- 学内12 サイトに分散したクラスタ群
- 総CPU数 800



Oo-okayama campus (10 sites)　　Suzukakedai campus (5 sites)

導入事例：東工大 Titech Grid

- 主にSCore MPI のクラスタとして使用
  - ユーザが直接ログインしてプロセスを起動することも可能
  - 占有することはできない
- MPIジョブのないときだけCondorのジョブが実行される
  - MPIジョブが投入され、ロードアベレージが上がるとCondorのジョブは停止する
  - MPI ジョブは比較的頻繁に投入されるため、Condorが連続して使用できる時間は短い

導入事例：東工大 Titech Grid

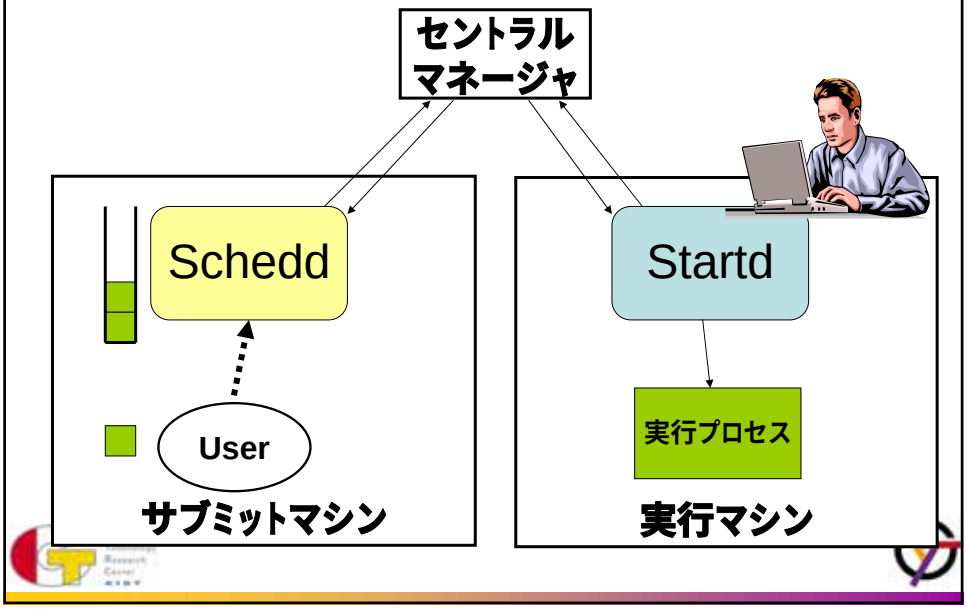
- **東工大 太田元則助教授による研究**
- **たんぱく質分子のMDによるフォールディングを初期値を変更して200回実行**
  - 統計的に処理することでフォールディングにいたる経路を議論
  - 1シミュレーションに数日
    - こんなに連続して CPUが使用できることはTitech Gridではほとんどない
  - にもかかわらず、200ジョブを一ヶ月で実行できた

チェックポイントとマイグレーション

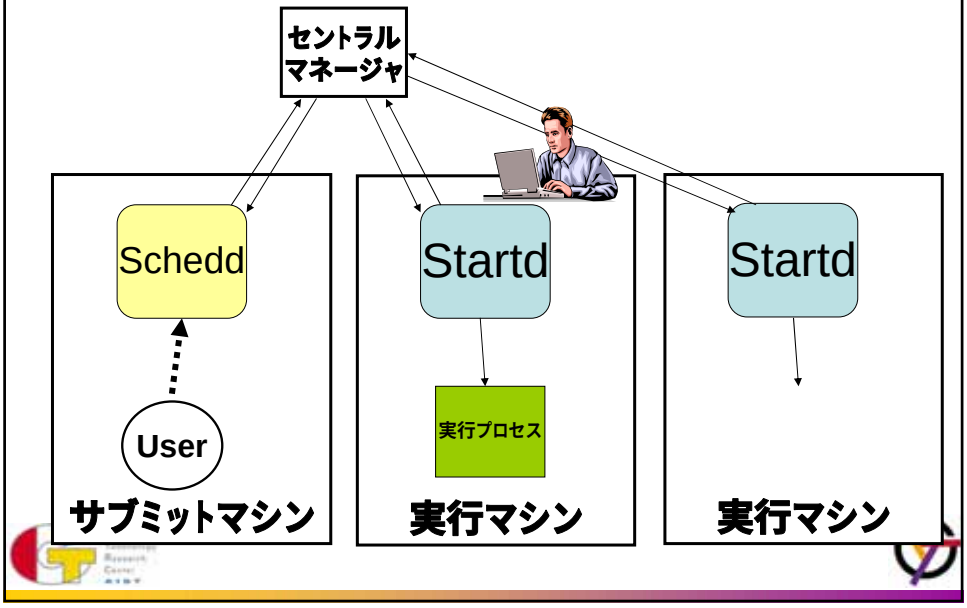
- **実行中のプログラムを状態を損なうことなく停止**
- **同じマシンもしくは別のマシンで実行を再開**

- **ファイルをオープンしていても大丈夫**
  - ファイルの同じ場所を参照したまま再開される
  - 注：Seekなどでファイルのあちこちを書くようなものだと、うまくいかない場合もある
- **ネットワーク接続するプログラムはうまくいかない**

チェックポイント



マイグレーション



サポートアーキテクチャ

- **Sun SPARC Sun4m,Sun4c, Sun UltraSPARC**
  - Solaris 8, 9
- **Silicon Graphics MIPS (R5000, R8000, R10000)**
  - IRIX 6.5 (clipped)
- **Intel x86**
  - Red Hat Linux 7.1, 7.2, 7.3, 8.0, 9, Enterprise Linux 3, Debian Linux 3.1 (sarge)
  - Fedora Core 1, 2, 3
  - Windows 2000 Professional and Server, 2003 Server (Win NT 5.0), Windows XP Professional (Win NT 5.1) (clipped)
- **ALPHA**
  - Tru64 5.1 (clipped)
  - Red Hat Linux 7.1, 7.2, 7.3 (clipped)
- **PowerPC**
  - Macintosh OS X (clipped)
  - AIX 5.2 (clipped)
- **Itanium (IA64)**
  - Red Hat Linux 7.1, 7.2, 7.3 (clipped)
  - SuSE Linux Enterprise Server 8.1 (clipped)

導入事例：東工大 Titech Grid

- **たんぱく質分子のMDによるフォールディングを初期値を変更して200回実行**
  - 1シミュレーションに数日
  - 個々のジョブはいくつものCPUを渡り歩いて終了
    - 延べ10-100 CPU
  - プログラムはAmber
    - Condorのチェックポイントライブラリとリンクしただけで、ソースコードは変更していない

専用のクラスタの利用

- Condorのメリット
  - 柔軟な制御
    - 特定の計算機ではその持ち主を優先するなど
  - フェアシェア
    - ユーザ優先度に基づいてリソースを割り当て
    - 優先度 1の人は 2の人の2倍リソースを使用できる

導入事例：The Hartford

- 創業約200年の保険会社
- 2003年の収益180億ドル
- 従業員30000人
- 債務のモデリング解析にCondorを使用
  - 商用ソフトウェアと比較の上で Condorを導入
- Condor プール
  - 200 CPU
  - 1月あたり10万ジョブ以上

導入事例：Core Feature Animation

- **CG製作会社**
  - X-MEN, Dr. Dolittle, Johnny Mnemonic, The Time Machine, などなど
- **CondorをCGのレンダリングに使用**
  - Macintosh を使用
  - デッドラインに応じたスケジューリングポリシを設定
    - シーンの優先順位に応じた資源の配分

導入事例：Oracle by Optena

- **リグレッションテストに使用**
  - プログラムを変更した際に副作用が出ていないかを確認するテスト
- **3000CPU以上のCondor Pool**
- **Condorを使用することで**
  - 1回に1週間かかっていたリグレッションテストが毎日150回できるように
  - システムの使用率が 15パーセントから80パーセントへ

並列ジョブスケジューラとしてのCondor

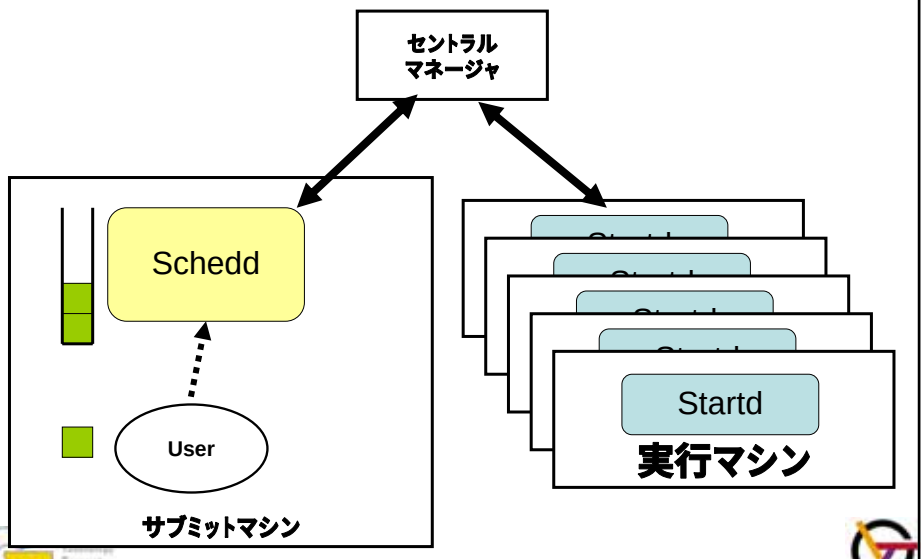
- PVMとは相性がよい
- MPIとはいまいち相性がよくない
  - 現在公開されているバージョンはMPICH 1.2.4 以下でしか動かない (1.2.6 が最新)
- 次期メジャーリリース(6.8系)では並列ジョブのサポートを大幅に強化予定
  - MPICH, LAM, SCore
  - 任意の並列アプリケーションを実行可能

導入事例：ーー

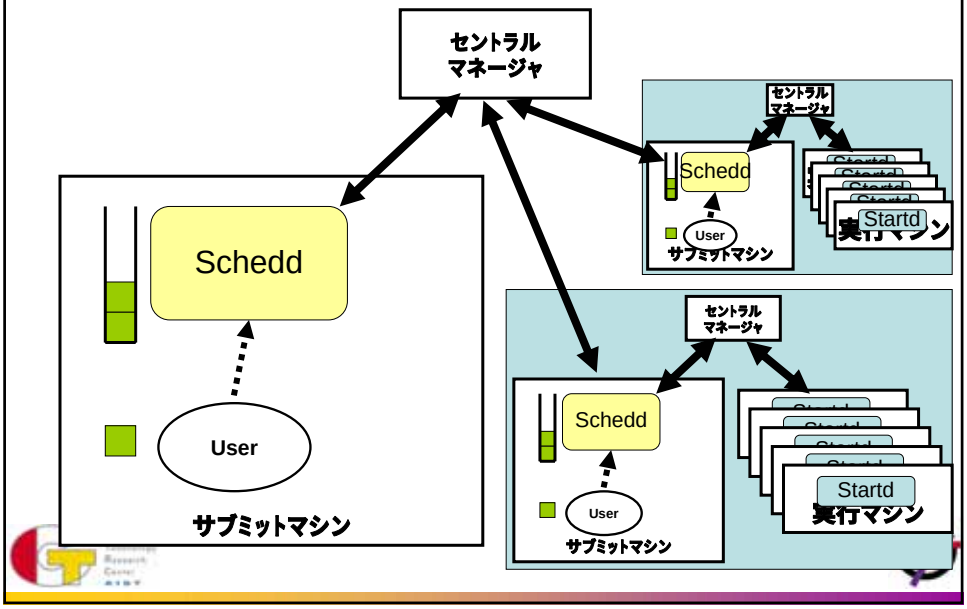
メタスケジューラとしてのCondor

- 個別のスケジューラで管理された複数の計算機群を「メタ管理」
- ローカルのクラスタ
  - PBS, LSF, etc..
- 遠隔地のクラスタ
  - Globus、NordGrid、Unicore, etc.

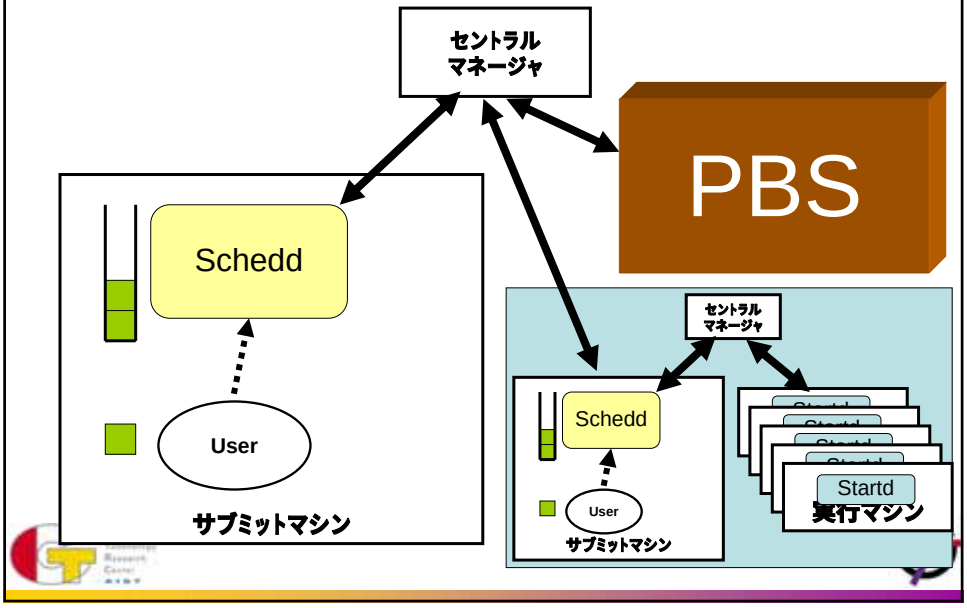
メタスケジューラとしてのCondor



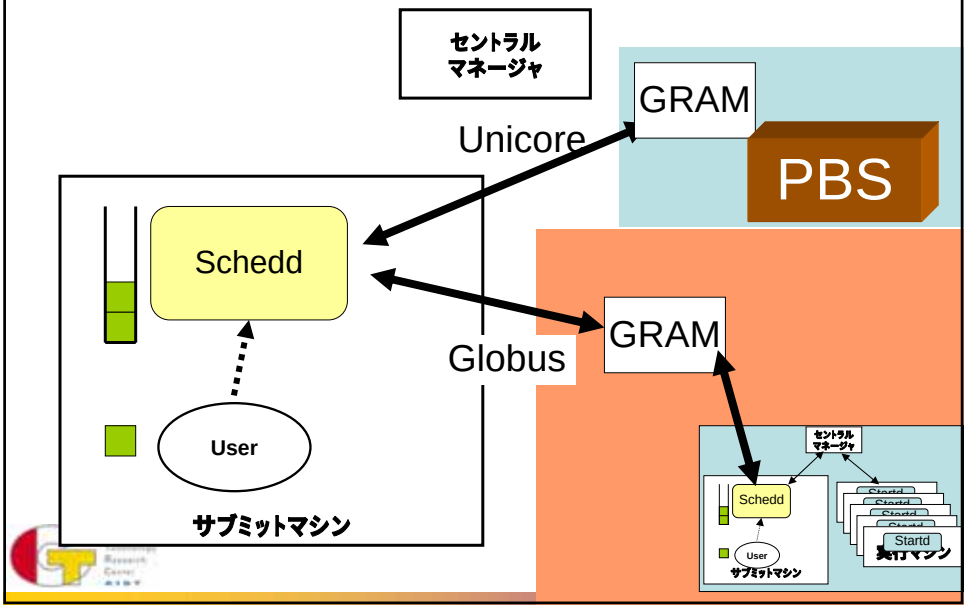
メタスケジューラとしてのCondor



メタスケジューラとしてのCondor



メタスケジューラとしてのCondor



導入事例

- EGEE
  - EUのプロジェクト
  - gLite – Condor-C
- CMS Data Grid
  - CERNの加速器を使った実験データを処理するためのデータグリッド

Condorのその他の機能

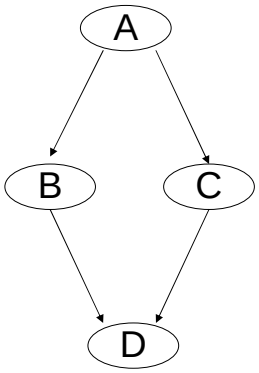
- Condor MW
- Condor DAGMan
- セキュリティ
- NAT越え

Condor MW

- マスターワーカ型アプリケーションの記述を助ける枠組み
  - C++の抽象クラスを提供、ユーザはこれを実装すればよい
- ワーカをCondorで沢山の計算機にばら撒く
  - マスタはサブミットしたマシンで稼動

Condor DAGman

- 複数の依存関係を持つジョブ群(ワークフロー)を依存関係の順番に実行
- 各ジョブごとにCondorのサブミッションファイルを記述
- ジョブ間の依存関係を簡単なテキストで記述



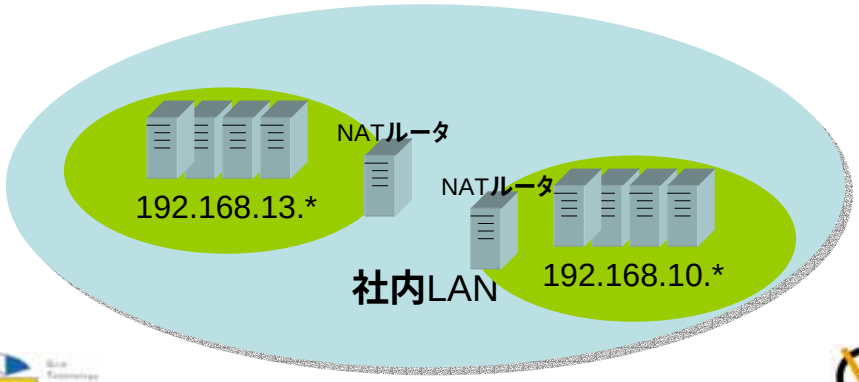

```
JOB A A.condor
JOB B B.condor
JOB C C.condor
JOB D D.condor

PARENT A CHILD B C
PARENT B C CHILD D
```

セキュリティ

- デフォルトではIPアドレスに基づいた制約のみ
  - 設定ファイルで通信相手のIPアドレスを指定
- 設定するだけでKerberosやPKIを用いた高度なセキュリティも利用可能
- 基本的には通信路のセキュリティと秘匿化だけで、メモリ上のデータの保護などは考えられていない

NAT越え

- 複数のプライベートアドレスに分散した計算機をひとつのCondorプールとして構成することが可能になる
- 次期メジャーリリースで組み込まれる予定



まとめ

- Condor は
  - 遊休計算機を利用したデスクトップグリッドシステム
  - 並列ジョブにも対応したバッチシステム
  - 複数のバッチシステムをたばねるメタスケジューリングシステム
- It's Free!
  - オープンソースではない(いまのところ)
    - セキュリティ
    - ビルド環境

Condorが有効な場面

- 比較的小さいジョブが沢山実行したい
  - 夜間使用されていないPCやWSがある
  - 専用のクラスタがある
- 1日ー1週間かかるようなジョブをいくつか実行したい
  - 計算機関占有できる計算機がない
- MPIなどで書かれた並列ジョブを実行したい
  - クラスタを使用
- PBSなどで管理された複数のクラスタを一括管理して有効利用したい
- 遠隔地のGlobusなどで管理されている複数のクラスタで大量のジョブを分散実行したい

参考URL

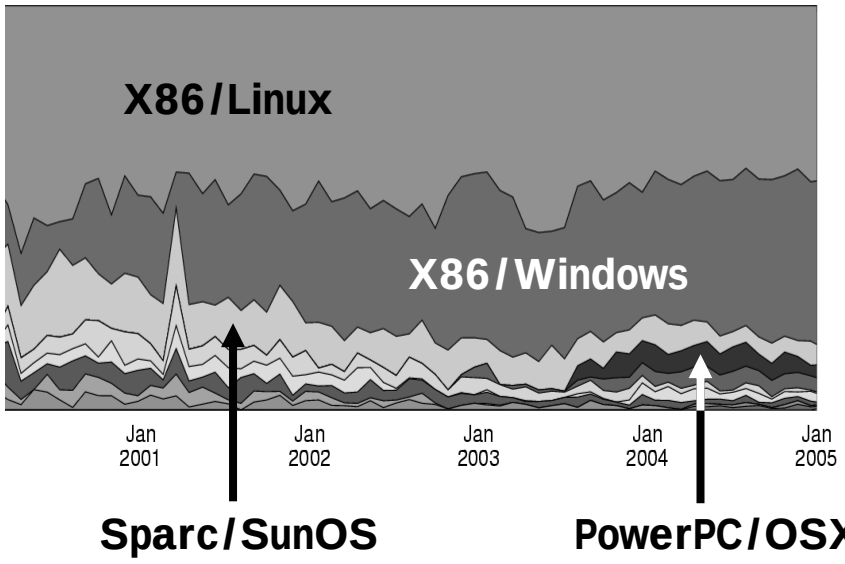
- http://www.cs.wisc.edu/condor
  - Condor Project Home Page
- ダウンロード、メイリングリストの登録などすべてここから
- Condor Weekのスライドも

参考URL(2)

- http://www.bestsystems.co.jp
  - ベストシステムズ
  - 産総研発認定ベンチャー
  - Condorチームと契約して日本語でのサポートを提供

?

Percent of downloads per month



Condorのポリシ

"The philosophy here is that we would like to encourage you to use as many cycles as possible and to do research projects that can run for weeks or months. But we want to protect owners, whether or not they are heavy users."