過去～現在～これからのNinf-G

Ninf-G ver.4 の紹介

今後の開発計画について(Ninf-G ver. 5)

田中良夫、中田秀基（産総研）

注意事項

- Ninf-G ver. 4の話をしますが、プログラミングのAPI（GridRPC API, Ninf-G API)に変更はありません。
- プログラム自体はNinf-G ver. 2と ver. 4の間で互換性が保たれます。
- ただし、実行時に指定するコンフィグレーションファイルなどには違いがあります。
  - アッパーコンパチ
- Ninf-G ver. 5についてはプログラム自体についても100%の互換性は保証できないと思います。
  - 振る舞いが変わる可能性がある。

Ninfプロジェクト

- 1995年3月？スタート
- 当時のメンバ
  - 関口智嗣、佐藤三久、長嶋雲兵
- その後
  - 中田秀基、松岡聡、高木浩光、建部修見、
    合田憲人、藤沢克樹、竹房あつ子
- もっとその後
  - 田中良夫、首藤一幸、横川三津夫、武宮博、谷村勇輔
- 現在の開発体制
  - 田中、中田、竹房、谷村
  - (株)創夢(2002年より)　朝生、三浦、田代、井上
  - (株)エス・エフ・シー(2004年より)　岸本

Ninf の歴史

- Ninf-1(1996)
  - セキュリティなし，Globusでは動作せず
- Ninf-G1(2000～2002)
  - 新しいコードベース
  - Globus 1.1.Xで動作
- Ninf-G2 (2002～)
  - またまた別のコードベース
  - Globus 2.X で動作
  - version 2.0.0は2004年3月に公開
  - 最新版は昨年8月に公開された2.4.0(安定&お勧め版)
- Ninf-G4 (2005～)
  - Ninf-G2と同じコードベース
  - Globus 2，4 で動作
  - 最新版は4.0.0 beta 2(ベータ版)
  - 正式版のリリースは2月24日(NAREGIシンポジウム)を予定

Ninf-G3 はどこへ行ったのか？

- Ninf-G2 for GT2
- Ninf-G3 for GT3
- Ninf-G4 for GT4

  - 実装したのだが，GT3が不安定でバグバグだったので公開にいたらず．

Ninf-G2とG4の違い

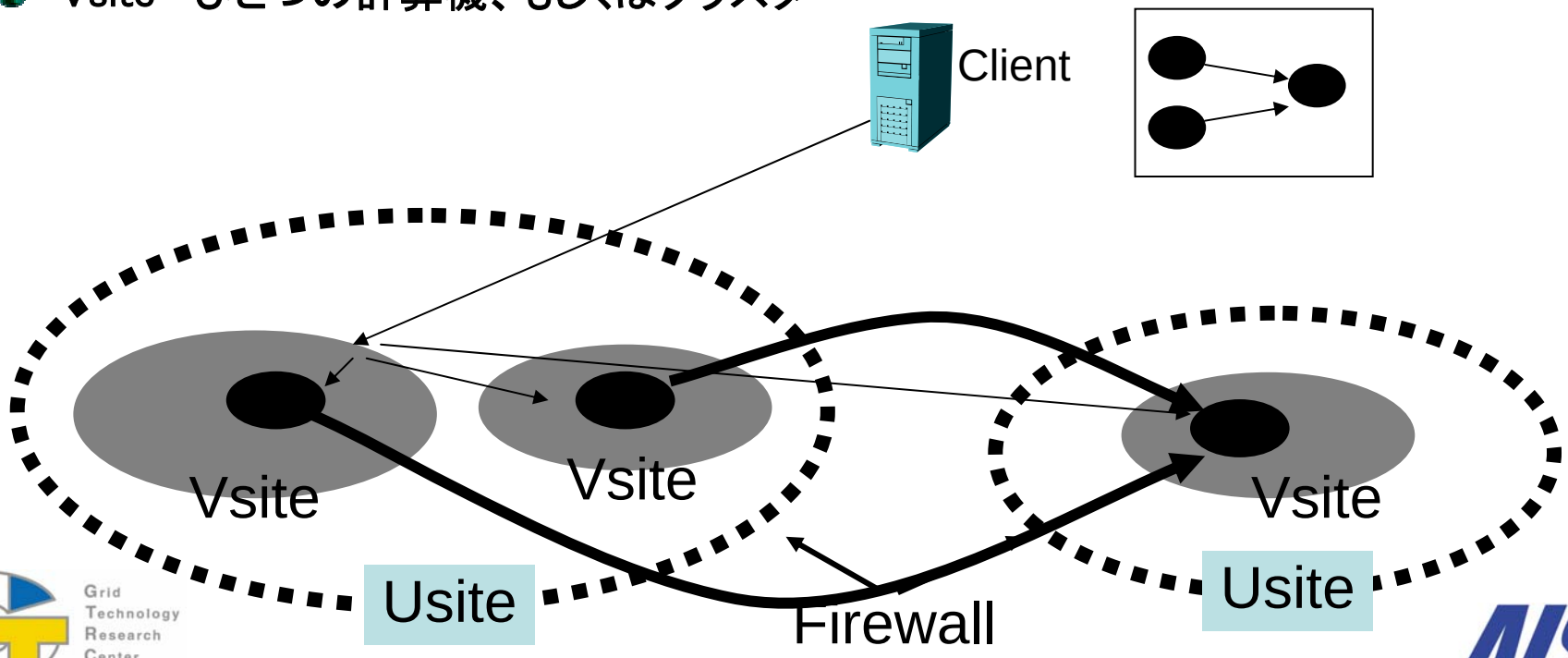
- さまざまなジョブ起動機構に対応しやすいよう，ジョブ起動部分の実装を変更
  - GT4に対応
  - UNICOREに対応
  - NAREGI SSに対応(予定)

GlobusToolkit 4の概要

- WSRF(Web Services Resource Framework)と呼ばれるWebサービスの一種で実装
  - Apache Axisをベース
- Globus Toolkit 2とは基本的に互換性がない
  - 機能的には共通
    - GRAM, MDS..
- JavaにはGRAM API があるがCにはなぜかない
  - C言語でGRAMを呼び出す機能を実現したコマンドは用意されている
  - 内部的にはツールが生成したスタブコードを直接利用している
    - ツールが変更された場合には影響を受ける
    - 低レベルなためコードのメンテナンスが困難
- C言語のライブラリからジョブ起動をするのは困難

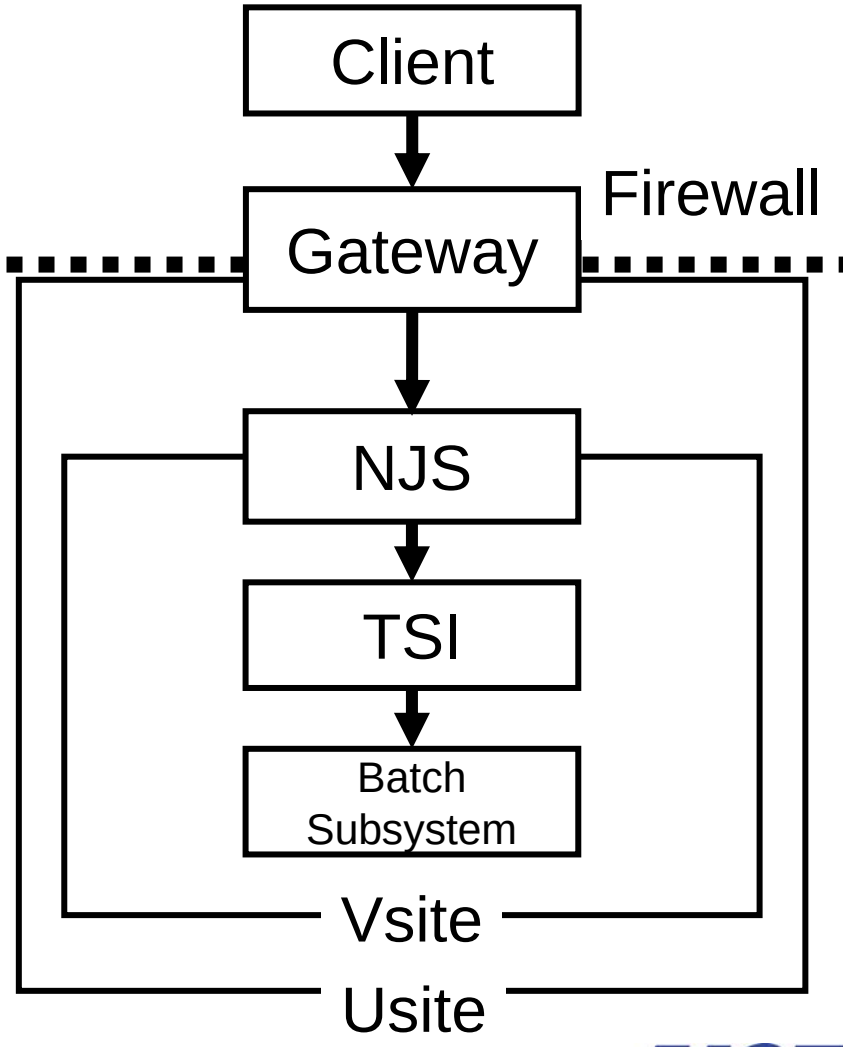
UNICOREの概要

- 欧州富士通研究所が中心となって開発
- 複数のコンピュータセンタに設置された計算機をシームレスに使用
  - ファイアウォール対応
- ワークフローをJavaのオブジェクト(AJO: Abstract Job Object)で表現、実行
- Usite– ひとつのファイアウォールで囲まれた単位
  - コンピュータセンター
- Vsite– ひとつの計算機、もしくはクラスタ



UNICOREの概要(2)

- Gateway
  - Firewallで稼動
  - すべての通信はこのプロセスを経由
  - SSLベースのセキュリティ
  - Javaで実装
- NJS (Network Job Supervisor)
  - ワークフローエンジン
  - AJOを解釈して実行
  - Javaで実装
- TSI (Target System Interface)
  バッチシステムとの間を仲介
  - Perlで実装



UNICOREの問題点

- ワークフローをJavaのオブジェクトで表現
  - Java標準機能を用いてワークフローに署名
- 通信プロトコルUPLもJavaのオブジェクトを前提
  - Java独自のシリアライザを利用する

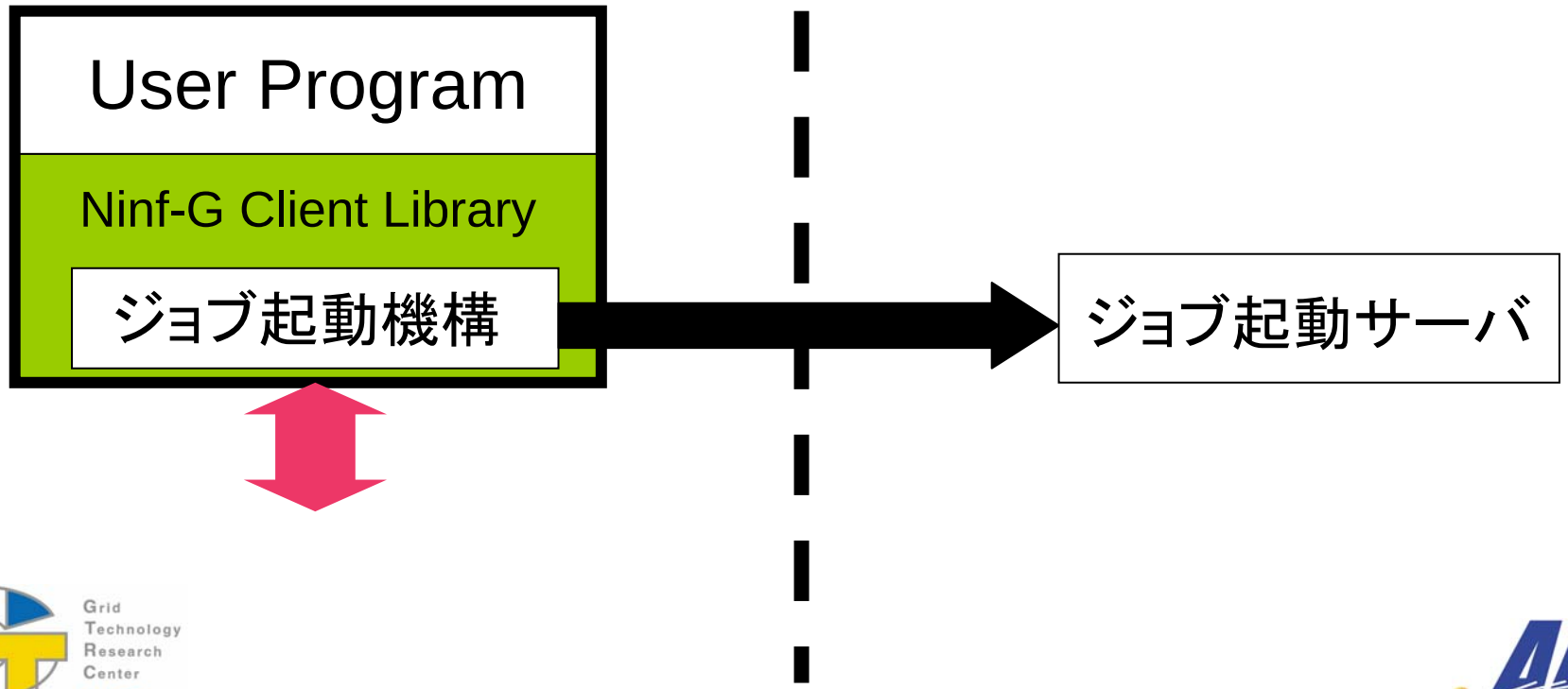
→ Java以外の言語でクライアントを書くことは、不可能ではないにしろ、非常に困難

問題点のまとめ

- Ninf-G2ではジョブ起動部がクライアントライブラリの一部としてリンクされる
  - C言語でジョブ起動部を実装する必要がある
- GT4やUNICOREではC言語でジョブ起動部を記述することが難しい
  - GT4 – C言語APIがない
  - UNICORE – Java以外でかけない

Ninf-G4のアーキテクチャ

- ジョブ起動機構をクライアントライブラリから外に
  - ジョブ起動部分をクライアントライブラリとは別の言語で実装することが可能に
  - クライアントライブラリ本体を修正することなく、他のリモートジョブ起動機構に対応可能



プロトコルの要件

- 言語独立性
  - 記述言語に固有のシリアライズ法は使用しない
    - 例： Java のシリアライズ、PythonのPickle
  - アーキテクチャ、コンパイラに暗に依存したバイナリデータ構造は用いない
    - 例: バイトオーダ、int のビット数
- 簡潔性
  - ジョブ起動モジュールを任意の言語で容易に書けるようにする

プロトコルの設計

- **通信フォーマット**
  - 簡単な文字列ベースプロトコル
  - 改行で区切られる
- **同期プロトコル ＋ 通知**
  - リクエストに対してレスポンスが1対1に対応
  - レスポンス以外に非同期の「通知」が行われる
- **通信路**
  - 標準入出力、エラー を使用
    - 入出力をリクエスト、レスポンスに
    - エラーを通知に
    - 外部起動モジュール側で新たにソケットをオープンする必要がない

リクエスト・リプライのプロトコル

- JOB_CREATE
  - ジョブを生成、起動
  - 引数として属性鍵、属性値のペアを持つ
  - ジョブIDを後に通知
- JOB_STATUS
  - 引数はジョブID
  - ジョブの状態を返す
- JOB_DESTROY
  - 引数はジョブID
  - ジョブ、およびジョブ関連情報を破棄する
- EXIT
  - 外部ジョブ起動モジュールを停止する

通知

- 作成通知
  - CREATE_NOTIFY
  - ジョブ作成リクエストの成否を通知
  - 成功時にはジョブIDを返す
- 状態通知
  - STATUS_NOTIFY
  - ジョブの状態の変化を通知

設定ファイルの拡張

- Ninf-Gの設定ファイル
  - 各サーバに固有の情報を指定

```
<SERVER>
hostname server.example.org
heartbeat 120
tcp_nodelay true
</SERVER>
```

- 外部ジョブ起動サーバに渡す任意の情報を書けるように拡張
  - クライアントライブラリ本体はそれらの情報を理解せずにジョブ起動サーバに引き渡す
  - 拡張性を確保

UNICORE用設定ファイルの例

```
<SERVER>
hostname server.example.org
invoke_server UNICORE
invoke_server_option vsite VsiteName
invoke_server_option usite UsiteName
invoke_server_option keystore keyfile
invoke_server_option passfile passfile
</SERVER>
```

ダミーホスト名
プログラム中で用いる

UNICORE用
ジョブ起動モジュール
使用を指定

起動対象となる
Vsite,Usiteを指定

認証用キーストア
ファイルを指定

キーストア
ファイルのパス
ワードを収めた
ファイル

提供予定の外部起動モジュール

- GT2向け外部起動モジュール
  - C言語で実装
- GT4向け外部起動モジュール
  - Pythonによるスクリプトで実装
  - C言語で記述されたコマンドラインインターフェイスを使用
- UNICORE向け外部起動モジュール
  - Java言語で実装
- NAREGI-SS向け外部起動モジュール実装中
  - Java言語で実装

Ninf-G4のまとめ

- GridRPCシステムNinf-Gの外部起動モジュール機構
  - 容易に外部起動機構を追加可能
- 単純なインターフェイスプロトコル
  - スクリプト言語でも容易に外部起動モジュールを実装可能
- クライアント設定ファイル
  - 設定情報をそのまま外部起動モジュールに流す
  - Ninf-Gクライアントライブラリに変更を加えずに外部起動モジュールの追加が可能

今後の開発計画について(Ninf-G ver. 5)

様々な多様性への対応

- 基盤システムの多様性への対応
  - 基盤ソフトウェア
    - Globus, Unicore, その他、
    - Globusなしでも大丈夫
  - （セキュリティ）ポリシ
    - 認証（GSI，SSL）
    - firewallがきつくても何とかなる
- 実行環境の多様性への対応
  - グリッド上
  - LAN内
  - 単一システム内
- アプリケーションの多様性への対応
  - 実行時間（数秒～数ヶ月）
  - 計算の粒度（細粒度～粗粒度）
  - 実行規模（サイト数・CPU数）
- 長時間実行への対応
  - 障害検知・復旧

Ninf-G Version 5 の構想

- Ninf-G5: 環境・ポリシ・アプリケーションの要求に応じた方法で遠隔手続き呼出を行なうGridRPCシステムの研究開発
  - Ninf-G ver. 2, ver.4 の基本機能の提供に加え. . .
  - LAN環境あるいは単一システム内では不要なオーバーヘッドを削減することにより高速な遠隔手続き呼び出しを実現する。
    - グリッドではGSI/Globus、クラスタ内ではrsh/ssh など
  - ジョブの要求に応じてクライアント・サーバ間の通信プロトコルを切り替えることにより、アプリケーションの性質に応じた長時間実行への対応を可能とする。
    - クライアントとサーバ間の接続を維持・切断
  - クライアントのチェックポイントをとることにより、今までのNinf-Gでは解決できていなかったクライアントがSingle point of failureとなる点を解決する。
  - プロトコル変更を伴う大幅な変更（実質書き直し）
  - 平成18年度末には Ninf-G version 5.0 の公開を予定

ラウンドテーブル

- 開発者とのフランクなQ&A、意見交換
- （恥ずかしがらずに）何でも聞いてください
- 登壇者
  - 田中良夫（産総研）
  - 武宮博（産総研、アプリ）
  - 朝生正人（創夢、開発）
  - 岸本誠（エス・エフ・シー、開発）