

GT3 によるソフトウェア開発

岸本 誠

(産総研GTRC上野オフィス
/(株)エス・エフ・シー)

主な内容

- GT3 GRAM 利用のノウハウ
- GT3 と GT4 の GRAM の速度比較
- ウェブサービス化の検討
- 開発に必要な情報

GRAM とは

Grid Resource Allocation and Management

- プロセス (ジョブ) の
 - 起動
 - 監視
 - 中断 (キャンセル)

Ninf-G・GridRPC と GRAM

- GridRPC - 計算グリッドの計算力を RPC モデルで利用
- Ninf-G ではリモートマシンで計算する実体を GRAM で起動

GT2 GT3 GT4 の比較 (1)

- ジョブ記述
 - GT2 RSL
 - GT3 XML化RSL
 - GT4 XML化RSL (変更点少)

GT2 GT3 GT4 の比較 (2)

- インタフェース
 - GT2 GateKeeper (daemon)
 - GT3 OGSi グリッドサービス
 - GT4 WSRF サービス+リソース

GT2 GT3 GT4 の比較 (3)

■ sandbox

- GT2 suid+exec
- GT3 UHE (Java Ap. WS コンテナ)
- GT4 sudo

GT3 GRAM 利用のノウハウ・サーバ

■ すぐ終るジョブ(ex test.xml)を繰り返すと

- メモリ不足で UHE が落ちる
- 対策

bin/launch_uhe.sh で起動している Java VM に "-Xmx????m -server" というオプションを追加

endorsed に xalan.jar を置く (GT3)

- GT3 添付の xalan.jar ファイル
- \${GLOBUS_LOCATION}/endorsed に置く
- Java VM を直接起動する時には -Djava.endorsed.dirs=... オプションを付ける
- これをやらないと
 - JDK 1.4.2_04 では大丈夫で _05 でダメという場合も

動かない (刺さる) 時の対策 (GT3)

■ UHE の資源を初期化する

--- 具体的には

- UHE のプロセス (UserContainer というクラスを main として実行中の Java VM) を kill する
- 以下のディレクトリを rm する
 - ▶ \${HOME}/globus/gass_cache (GASSの設定)
 - ▶ \${HOME}/globus/job (ジョブのstdout/err)
 - ▶ \${HOME}/globus/uhe-[FQDNホスト名] (WSコンテナのroot)
- 証明書等を消さないように注意

GT3 の GRAM が重い理由

- ユーザ毎に VM とコンテナが起動
 - 最初の 1 回に非常に時間がかかる
- クライアントでも notification のためにコンテナが起動
- managed-job-globusrun スクリプトで重い理由
- (Java VM を終了しないようにすればそこそこ速い)

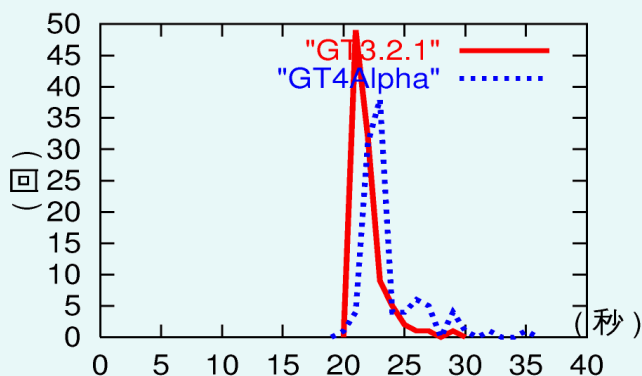
GT 3.2.1・4α 速度比較

- Pentium4 1.7GHz
- 768Mbyte
- JDK 1.4.2_05
- (同一マシンで計測)

3.2.1・4α どちらも all-source-installer でインストールしたままの環境で実行

GT 3.2.1 · 4 α 速度比較

managed-job-globusrun 1 回の実行時間



GT 3.2.1 · 4 α 速度比較

- クライアントでの **start** の実行からリモートでのジョブ起動まで (VM は再起動せず)
 - 3.2.1 --- 1~1.5 秒
 - 4 α --- 3~4 秒
 - (同一マシンで計測)

GT4 α impression

- サービスは動的に生成されない
- **sudo** によるジョブの実行
- 軽い - メモリをくわない
 - 安定
 - 実測では速くはなっていなかった
- (改良されているはずなのに)ほとんど速くはない理由は今後調査の予定

GRAM クライアント (GT3)

- Java 版クライアントは問題なし
- GT2 互換 API (RSL→XML 等)
 - あやしい
 - バグの報告中
- C バインディングは使えません (後述)

C バインディング (1) (GT3)

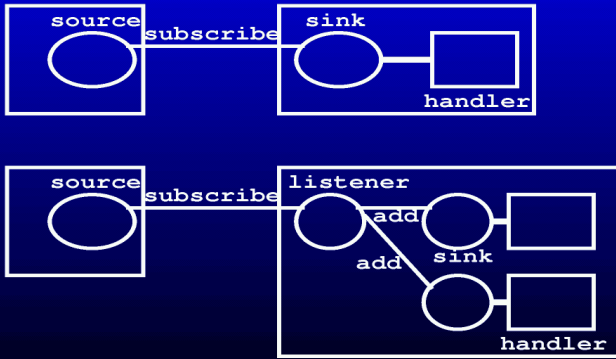
- **pthread** フレーバはデフォルトでは用意されていない
- バグが残っているが, 3.2.0 以降更新されていない
- 修正: **context diff** で 3000 行

C バインディング (2) (GT3)

C Programming Guide

<http://www-unix.globus.org/toolkit/docs/3.2/core/developer/cprogramming.html>
の **notification** の説明は現状と違っている

C バインディング (2) (GT3)



ウェブサービス (GT3~)

■ XML と HTTP ベースの RPC

■ 問題点

- 木構造のデータに縛られる
- パースが重い

■ 改善案

- シリアライズ → `byte[]`
 - ▶ 速い, が...
- パーサの特定構文専用化, 等
 - ▶ 既存の資産が使えない

わかりにくい Globus サイト

- `{www|www-unix|www-fp}.globus.org` ってどれがどれ?
- `globus_common` のドキュメント
 - <http://www.globus.org/common/>
 - <http://www-fp.globus.org/common/>
- <http://viewcvs.globus.org/viewcvs.cgi/>