

SC2003テクニカル セッション報告

丸山直也

naoya@matsulab.is.titech.ac.jp

東京工業大学



発表概要

- SC2003概要
- テクニカルセッション報告
 - グリッド関係のセッションからいくつかを選択


- disclaimer
 - SCの参加は初めてです
 - 普段はグリッドにあまり関係のないテーマに取り組んでいます



SC2003概要

- <http://www.sc-conference.org/sc2003/>
- 11/15-21
- Phoenix, Arizona
- 総参加者数7641
- テクニカルセッション
 - テクニカルペーパー
 - マスターワークス
 - ポスター
 - ワークショップ
 - 招待講演
 - BoF
 - チュートリアル

などなど



テクニカルペーパーセッション

- Chair: Diane Rover, Iowa State Univ.
- Vice Chair: Barton Miller, Univ. of Wisconsin, Madison
- 207本の投稿中、60本受理
- 3日間、20セッション
- テーマ
 - クラスタリング、ネットワーク、グリッド、ツール、可視化、性能、数値計算、Gordon Bell Award、など



Cluster-Based Servers

- Hong Tang (University of California, Santa Barbara), et al., *“An Efficient Data Location Protocol for Self-organizing Storage Clusters”*
- Changxun Wu (Johns Hopkins University), *“Handling Heterogeneity in Shared-Disk File Systems”*
- Kiran Nagaraja (Computer Science, Rutgers University), et al., *“Quantifying and Improving the Availability of High-Performance Cluster-Based Internet Services”*



An Efficient Data Location Protocol for Self-organizing Storage Clusters

- ストレージクラスタにおける柔軟なデータ配置方法を提案
 - ノードの動的な追加・削除 効率の良いメタデータ管理、マイグレーション技術が必要
 - ファイルシステム中のファイル
 - サイズ小: 大部分のファイル、一部の領域
 - サイズ大: 一部のファイル、大部分の領域
 - ファイル配置方法
 - ハッシュ: ○位置検索コスト低、○メモリ使用量少、×利用効率悪、×再構成時マイグレーション必要
 - ブルームフィルタ: ○再構成時マイグレーション不要、○利用効率良、×位置検索コスト高、×メモリ使用量多
 - サイズによって配置方法を変更
 - 小: ハッシュ値
 - 大: ブルームフィルタ
- webのインデックシングを行うアプリケーションで効率改善を確認



Handling Heterogeneity in Shared-Disk File Systems

■ ディスク共有ファイルシステム

- 共有ディスク、ディスクのメタデータを管理するサーバー、それらをつなぐSANから構成
- クライアントはサーバーにメタデータを問い合わせ、それに基づき共有ディスクに対してI/Oを実行
- メタデータ管理サーバーは複数
 - ファイルシステムを分割してメタデータを分散管理
- 問題
 - アクセスパターンによりサーバーの負荷が不均一
 - サーバー自身のハードウェア性能が不均一

■ メタデータ管理のロードを均一可

- ファイルIDのハッシュ値をとりメタデータ管理サーバを決定
- ハッシュ関数の値域をサーバ間で分割
 - 領域均等分割で開始 実行時のファイルアクセスレイテンシを元に動的に再構成
- ○アプリケーション、ハードウェアに関する事前知識不要
- アプリケーション、ハードウェアの不均一性を考慮して静的に分割した構成と同程度の性能を達成

Quantifying and Improving the Availability of High-Performance Cluster-Based Internet Services

■ クラスタベースのサービス

- ノード間の協調動作 ○ロードバランス向上、など、×可用性低下
- 目標: 協調動作を許し、かつ可用性を維持
- 解決法: さまざまなCOTS + ソフトウェアによるFT技術
- 問題?
 - FT技術を適用したときに可用性がどの程度向上するか?

■ 貢献

- クラスタベースのサービスにおいて、FT技術の効果を定量的に推定
 - 故障をソフトウェア故障発生器を使ってエミュレートし、故障による可用性低下、FT技術を適用した場合の可用性向上率を**実環境**で計測
 - 上記基礎データをもとに任意のワークロードにおける性能、可用性を予測

種々のFT技術適用の指針



Grid Support

- Dong Lu (Northwestern University), et al., “*Synthesizing Realistic Computational Grids*”
- Xin Liu (UCSD), et al., “*Traffic-based Load Balance for Scalable Network Emulation*”
- Ali Raza Butt (Purdue University), et al., “*A Self-Organizing Flock of Condors*”



Synthesizing Realistic Computational Grids

- シミュレーション用仮想グリッド
 - ミドルウェアの評価などのため
 - 仮想グリッドの構成が現実を反映する必要性
- GridG
 - 計算グリッド向け仮想グリッド構成を作成ツール
 - トポロジ
 - ホスト、ルータ、ネットワークリンク、から構成
 - インターネットの”power law”に従った階層構成
 - リソースのプロパティ
 - ヒューリスティクス(例:ホストが多数のCPUからなるSMPならばメモリも大量に搭載)により生成
 - <http://www.cs.northwestern.edu/~donglu/GridG.html>
からソースコードをダウンロード可能




A Self-Organizing Flock of Condors

- Condorプールの”flock”を自動構成
 - P2P技術によりリモートプールの発見、flockへの動的なjoin/leaveを実現
- Condorを拡張し、self-flockingを実現
 - FreePastryを実装に使用
 - P2Pアプリケーション作成支援ライブラリ
 - <http://research.microsoft.com/~antr/Pastry/>
- 大西洋をまたいだ4つのプールをflockingした実験により、ジョブスループットの向上を確認
- セキュリティに関しては以下の対処法を提唱
 - 設定ファイルで静的にflockingを許可するリモートプールを指定
 - 通常のsandboxing機構を使用
 - 認証を行うレイヤを追加




Tools and Services for Grids

- Peter Dinda (Northwestern University, Computer Science), et al., “*Nondeterministic Queries in a Relational Grid Information Service*”
- Tahsin Kurc (Ohio State University), et al., “*Optimizing Reduction Computations In a Distributed Environment*”
- Hongzhang Shan (Lawrence Berkeley National Laboratory), et al., “*Job Superscheduler Architecture and Performance in Computational Grid Environments*”



Nondeterministic Queries in a Relational Grid Information Service

- Relational GISサーバにおけるSQL処理
 - 例:「計32GBのメモリを持つ、同一LAN内に存在する16台のノードからなるセットを見つけよ」
 - × 完全な結果は処理コスト高
 - 実際は、一部だけで十分な場合が多い
 - ある時間内に見つけれられた結果のみ、など
 - ただし、常に「同じ」一部であっては困る
- RGIS
 - 著者らによるRelational GISサーバの実装
 - RDBMSとしてOracle RDBMSを使用
 - クエリは標準SQLで記述
 - nondeterministic なクエリを実装
 - ノード集合からランダムに選択した部分集合に対して検索を開始
 - 部分集合の割合を調整することで、クエリ処理時間と結果の量のトレードオフを調整可能
 - 処理時間にデッドラインを設定すること可能
- 前述のGridG使って構成した仮想グリッドの情報を持つRGISについて実験。トレードオフの調整が可能なることを確認。



Job Superscheduler Architecture and Performance in Computational Grid Environments

- **スーパースケジューラ (GS) のアーキテクチャを提唱**
 - distributed
 - ローカルスケジューラ (LS) と GSの協調動作を定義
 - sender-initiated, receiver-initiated, symmetrically-initiated の3種類を提唱
 - 例: sender-initiated
 1. ジョブをあるサイトのGSに投入
 2. GSがそのサイトのLSにジョブに実行がまわってくるまでの時間をクエリ
 3. 時間がある一定値以下 そのLSにジョブを投入
 4. 一定値以上 リモートのGSにジョブのターンアラウンドタイムをクエリ
 5. ターンアラウンドタイム最小のリモートGSにジョブをマイグレーション
 - **あくまでブループリント**
 - ジョブ実行時間のみでマイグレーションを判断
 - ネットワーク越しのマイグレーションコストを考慮しない
- **実ワークロードを用いたシミュレータ上での評価**
 - USの複数のスパコンセンタで6ヶ月間にわたるデータ収集
 - それぞれ個別にスケジューリング(GSなし)で運用
 - GSを導入した場合のジョブのレスポンスタイムを実データと比較し、性能向上を確認



Data Managements in Grids

- Gurmeet Singh (Information Sciences Institute, University of Southern California), et al., “*A Metadata Catalog Service for Data Intensive Applications*”
- Ewa Deelman (ISI), et al., “*Grid-Based Galaxy Morphology Analysis for the National Virtual Observatory*”
- Matthew S. Allen (University of California, Santa Barbara), et al., “*The Livny and Plank-Beck Problems: Studies in Data Movement on the Computational Grid*”



A Metadata Catalog Service for Data Intensive Applications

- Metadata Catalog Service (MCS)
 - オブジェクトのメタデータをDBに保持
 - 例えば、オブジェクトが実験データならば、メタデータとして実験を行った日付などを持つ
 - クライアントはメタデータでオブジェクトを検索可能
 - 「何月何日に生成したデータは？」
- 実装
 - ファイルのメタデータを対象としたMCSを実装
 - メタデータはユーザによる追加可能
 - バックエンドDBMSとしてMySQLを使用
 - Webサービスとして実装
 - 単一サーバーで集中管理
 - 対象アプリケーションで強い一貫性維持が必要なため
- 実装したMCSを人工的に作成したDBで評価
 - 単一メタデータによる検索に比較して、複数のメタデータの合成による検索の性能が大幅にダウン MySQLによる
 - SOAPのオーバーヘッド大
 - SOAPなしに比べて数パーセントの性能に低下




The Livny and Plank-Beck Problems: Studies in Data Movement on the Computational Grid

- データグリッドにおける重要な課題を提唱
- Livny problem
 - 仮定: k 個の同一サイズのファイルが k 台のホスト分散
 - 目標: ある一点への短いダウンロード時間と多数のダウンロードファイル数の達成
 - 想定: 分散アプリケーションのチェックポイントの管理ノードへの収集
- Plank-Beck problem
 - 仮定: 巨大なファイルが k 個のセグメントに分割され、各セグメントのレプリカが分散配置。かつ各セグメントは r 個のレプリカを持つ。
 - 目標: ファイル全体をある一点へダウンロードする時間
 - ダウンロードはセグメントの順序を保つ必要有り
 - 想定: P2Pファイル共有システムにおけるファイルのダウンロード
- 両問題に対して、いくつかのアルゴリズムを提案
 - 静的、NWSによる動的予測機能を使用、など
 - 各アルゴリズムを実装し、実環境で比較
- 問題に名前をつけて、明確にしたことが貢献(?)



その他

- 性能解析関係の発表多数
- Gordon Bell Award
 - 地球シミュレータで地震シミュレーションプログラムを高速化



おわりに

- このスライド
 - <http://matsu-www.is.titech.ac.jp/~naoya/reports/sc2003.pdf>
- 小柳義夫@東大、「SC2003報告」
 - SC全体に関する詳細な報告書(ただし、テクニカルペーパーについてはあまり記述されていない)
 - <http://olab.is.s.u-tokyo.ac.jp/~oyanagi/reports/SC2003.txt>
- 丸山直也、「APART5報告」
 - 併設ワークショップ APART (Automatic Performance Analysis: Resources and Tools, <http://www.fz-juelich.de/apart/sc03/>) に関する報告書
 - <http://matsu-www.is.titech.ac.jp/~naoya/reports/apart5.pdf>



おわり

- 以降、予備スライド



Gordon Bell Performance Evaluation

- Dimitri Komatitsch (California Institute of Technology), et al., “*A 14.6 billion degrees of freedom, 5 teraflops, 2.5 terabyte earthquake simulation on the Earth Simulator*”



Performance Analysis and Modeling

- Fabrizio Petrini (Los Alamos National Laboratory), et al., “*The Case of the Missing Supercomputer Performance: Achieving Optimal Performance on the 8,192 Processors of ASCI Q*”



Tool Infrastructure

- Philip C. Roth (University of Wisconsin-Madison), et al., “*MRNet: A Software-Based Multicast/Reduction Network for Scalable Tools*”
- Barton Miller (University of Wisconsin), et al., “*The Tool Daemon Protocol (TDP)*”
- Lingyun Yang (Department of Computer Science, University of Chicago), et al., “*Conservative Scheduling: Using Predicted Variance to Improve Scheduling Decisions in Dynamic Environments*”



Compilation Techniques

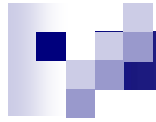
- Yonghua Ding (Purdue University), et al.,
*“A Compiler Analysis of Interprocedural
Data Communication”*
- pending



Software Systems



Performance and Reliability



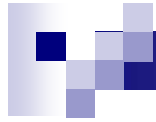
Supercomputing Applications



Networking



Performance Programming



Gordon Bell Computational Methods



Runtime Systems



Algorithms and Programming



Advanced Architectures



Scheduling and Communication



High Performance Input/Output



Performance Measurement and Analysis