平成 21 年度 RICC 利用報告書

課題名（タイトル）：

理研サイネスデータベースを用いた大規模分散処理

利用者氏名：豊田　哲郎
所属　　：横浜研究所　横浜研究所　生命情報基盤研究部門

１．　本課題の研究の背景、目的、関係するプロジェクトとの関係

理研サイネスは，ライフサイエンスにおける各種データベースを格納し，編纂，公開する為のフレームワークである．生命情報基盤研究部門では，この理研サイネスを運用するにあたり，単にデータベースを公開するだけでなく，様々な付加価値を加える努力をしている．その１つにデータベース横断的な検索機能の提供があり，これを支える内部処理としてインデキシングとクローリングがある．現在，定常的なクローリングを必要としているインスタンスが 450 万以上存在する．サービスとして運用していくために，これらインスタンスに対してクローリングを行う必要があり，膨大な計算リソースを必要としている．本プロジェクトは，これらクローリング処理に RICC の計算リソースを用いることで，処理に必要な総所用時間の短縮を目的とするものである．

２．　具体的な利用内容、計算方法

1)概要

理研サイネスのクローリング・ジョブの実行時間は，そのジョブに含まれるインスタンス数に比例関係がある（**図 1**）．今回は，これらインスタンスの集合を単純に分割し，大規模分散化による高速化手法を採用した（**図 2**）．分散化されたジョブは，横浜側に設置されたストレージにデータの読み込み及び結果の書き出しを行うことで計算処理が進行する．

図 2 に示すように，現在利用可能な計算リソースとして，生命情報基盤研究部門が既に持っているリソースと，RICC の多目的 PC クラスタから割り当てられる 100 ノードのリソースがある．それぞれのリソース・ロケーションには，それぞれローカル・スケジューラを持っており，今回これらに対してジョブ割り当てするためのディスパッチャを開発した．これについては後述する．最大同時実行数としては，RICC が 64 ジョブ，和光は 12～16 ジョブまで利用可能である．

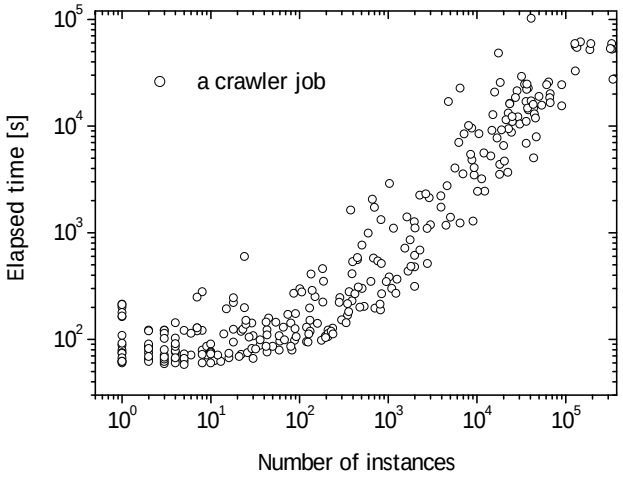

図 1 インスタンス数と経過時間の関係

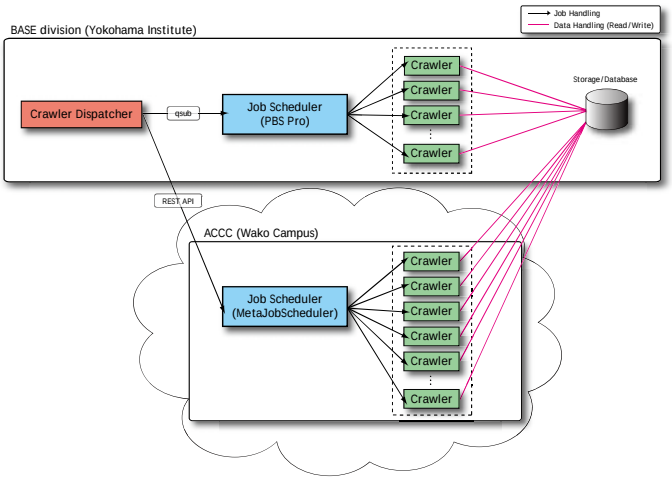

図 2 REST API を用いたジョブ・ステータスのモニタリング及び投入

2)RICC内部VLANの横浜研究所への延伸

計算リソースは和光に，データベースは横浜研究所に配置されていることから，リモート間のデータのハンドリングがクリティカルな問題としてあった．今回は，この問題を解決する為に，1Gbps の広域イーサネットを用いて，RICC の内部 VLAN を横浜研究所まで延伸し，延伸した先にゲートウェイを設置し，そのゲートウェイと理研サイネスを接続することで根本的な解決を図

った（**図 3**）．この試験環境の実測ネットワークスループットは 851[Mbps]，RTT は 1.6[ms]と，ローカルネットワーク並の性能が得られている．また，ミドルウェアには SSHFS と HPN-SSH を利用し，暗号化と非暗号化を使い分けることで，ネットワークスループットを稼いでいる．

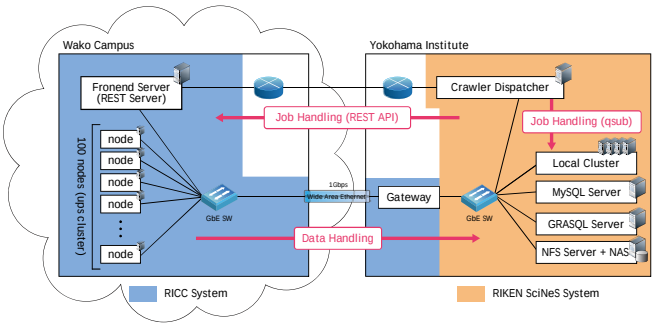

図 3 広域イーサネットを用いた RICC 内部 VLAN の延伸

3)RESTを用いたリモート・ジョブ割り当て

RICC と理研サイネスはそれぞれ自律的に運用されているシステムである為，これらを接続するためのインタフェイスが必要となる．今回は RICC が提供している REST とばれる Web サービスを用いることで，リモートからのジョブのハンドリングを可能としている．本年度は，理研サイネスからこのインタフェイスを用いるための Java のクライアント・ライブラリの開発及びシェル・スクリプトによるシステム構築を行った．

4)ジョブ割り当ての最適化

現在のシステムは，2 つのリソース・ロケーションがある．最大同時実行数としては，RICC が 64 ジョブ，和光は 12～16 ジョブまで利用可能である．これらロケーションに対して，最適に割り当てるためのアルゴリズムの実証研究を行った．所謂組み合わせ最適化と呼ばれる領域に属し，極めて古典的な問題ではあるが，NP 完全でもある．現在のクローリングは，１回のクローリングにあたり，650～1100 程度のジョブが生じ，各ロケーションに対して振り分けている．最適化には動的最適化と静的最適化があり，これらをヒューリスティックに組み合わせることで，クローリング処理の総所要時間の短縮を試みた．

３．結果

1) 横浜側ストレージに対するRead/Writeベンチマークテスト

図 4 に，横浜研究所に設置されているターゲット・ストレージへアクセスした場合の，Read/Write 性能を示す．測定には，10GB のファイルを dd コマンドで作成に必要な経過時間を元にスループットを算出した．なお，ターゲット・ストレージは NFS によってマウントされており，同一セグメントに設置された計算サーバ(yiccom)からアクセスした．また，キャッシュの影響を抑えるために，測定環境のサーバではすべての測定の度に sysctl を用いたクリアを行った．

測定の結果，最大 Read 性能が 63[MB/s]，最大 Write 性能が 76[MB/s]程度得られており，同時実行数（多重度）が増加するとスループットも低下した．一般的なネットワーク・ストレージに見られる傾向である．

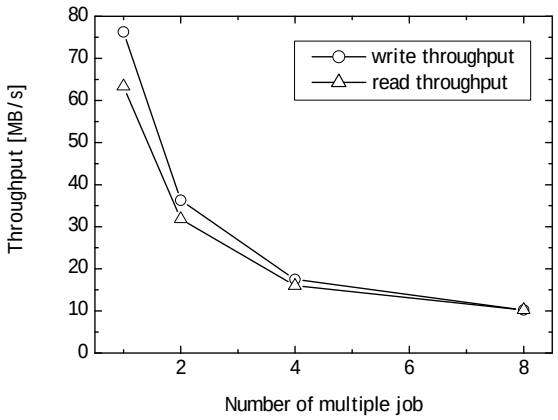

図 4 横浜側ストレージの Read/Write 性能
（ローカルネットワーク内からのアクセス）

図 5 に，広域イーサネットを用いた場合の同ターゲット・ストレージの Read/Write 性能を示す．最大 Read 性能が 29[MB/s]，最大 Write 性能が 22[MB/s]が得られた．これについても同時実行数の増加により，スループットの低下が見られた．図 4 と図 5 の比較から，同時実行数が大きくなった場合，ローカルネットワークでの測定結果に律速していることから，現時点ではネットワークよりもストレージに高速化の鍵があると考える．一方，大規模分散によって同時実行数が大きくなる今回のケースでは，ローカルネットワークからのアクセスとほぼ同等な性能になることが示された．

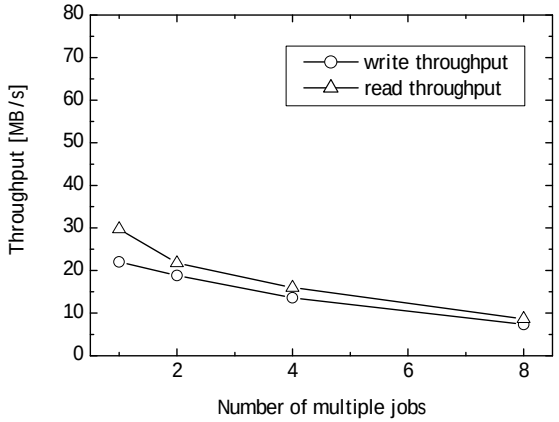

図 5 横浜側ストレージの Read/Write 性能

(広域イーサネット経由でのアクセス)

2)ジョブ・シミュレーション

図 6 に，ローカル・クラスタのみを用いた同時実行数 8 の実測結果を用いたジョブ・シミュレーションの結果を示す．クローリングのジョブは，そのジョブ間に順序や依存関係がないことから，典型的なジョブ割り当て問題である．なお，クローリング全体の総所要時間は，ジョブ集合に含まれる 1 つのジョブの最長経過時間に収束する．

図 6 を基にすることで，例えば，同時実行数を 8 ジョブから 16 ジョブに増加させることで，スループットは 18[instances/s]から 35[instances/s]程度までは理想的には増加することが予想される．同時実行数を 16 ジョブから 32 ジョブに増加した場合は，スループットは 30[instances/s]から 42[instances/s]程度までは理想的に増加することが予想されるが，前よりは鈍っている．最終的に同時実行数を 32 ジョブから 64 ジョブに増加した場合は，スループットは 42[instances/s]から 44[instances/s]程度しか増加しない．これらは図 5 に示したような同時実行数を増加させた場合のストレージ自体のスループットの低下等を考慮していない．あくまで理想的な条件下でのクローリング・ジョブの最高性能を予測するものであるが，ジョブの定性的な傾向を計る指標としては有用であると考える．

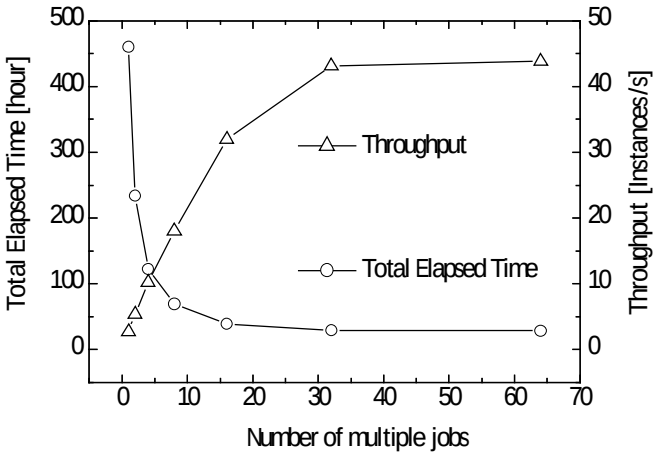

図 6 クローリング性能のジョブ・シミュレーション

3)パフォーマンス・テスト

図 7 に，2 つのロケーションを用いて構成した 4 条件についての実測結果を示す．ローカル・クラスタ単独で同時実行数 16 ジョブの場合，クローリングのスループットとして 19[instances/s]が得られた．RICC 単独で同時実行数 32 ジョブまで増加させた場合は，22[instances/s]が得られた．さらに，ローカル・クラスタで同時実行数 16 ジョブ及び RICC で同時実行数 16 ジョブを実行した場合，24[instances/s]程度まで上昇した．図 6 のジョブ・シミュレーション結果に比べると相当乖離があるが，シミュレーションで考慮していない要素（同時実行数の増加に伴う性能低下）が大きく働いた為と考える．以上，リモート環境である RICC 単独の結果と，ローカル環境単独の結果に大きな差がないことから，今回開発システムそのものには問題がないと考える．

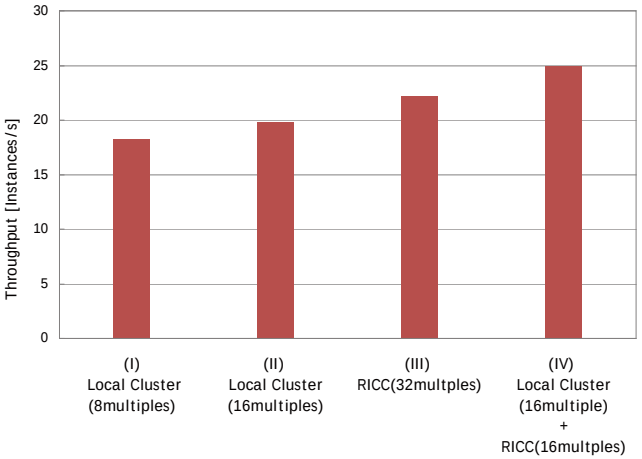

図 7 クローリング性能の比較

(RICC 及びローカル・クラスタ)

４．　　まとめ

今回，RICC と理研サイネスを広域イーサネットで接続し，RICC の多目的 PC クラスタを用いた大規模分散化によるクローリング処理の高速化手法を採用し，検証を行った．結果，同時実行数 32 ジョブまでの実行は確認でき，ローカルネットワークと同等な性能が得られた．以上について，中間報告を含む内容のポスター発表を行った．

５．　今後の計画・展望

今回構築したシステムを用いて，クローリング処理を対象とした同時実行数 32 ジョブまでのパフォーマンス・テストは完了した．今後の運用計画を立てる上で必要となる各種実データを蓄積することができたので，来年度は一般利用で申請する予定である．来年度は，クローリングの効率化とともに，もう１つの内部処理であるインデキシングについて検討を重ねる予定である．

６．　RICC の継続利用を希望の場合は、これまで利用した状況（どの程度研究が進んだか、研究においてどこまで計算出来て、何が出来ていないか）や、継続して利用する際に行う具体的な内容

理研サイネスの内部処理であるインデキシングとクローリングを対象に，システムの構築を実証的に検討をしている．現在までの利用で，クローリングについては一定の見通しがつき，大規模分散化への方法論を得た．今後は，クローリングの効率化とともに，インデキシングについて検討を重ねる．クローリングの高速化は，ジョブに含まれるインスタンス数を現在調整すべく実データを蓄積しているが，これを基に実環境に沿った形での最適化を行う．また，同時実行数を増やした場合にストレージの本来性能が低下しているが，これに本質的な対応を行うことで，より大きな同時実行数での実行を検討する．今回のシステムは，アプリケーションに大きく依存している．インデキシングに関しては，性能がどの程度見込まれるか基礎調査を行う予定である．

平成 21 年度　RICC 利用研究成果リスト

【その他】

Masaaki Terai, Yuko Yoshida, Norio Kobayashi, Yoshiki Mochizuki, Akihiro Matsushima, Motoyoshi Kurokawa, Takayuki Shigetani, Tsuyoshi Horiki, Ryutaro Himeno, Tetsuro Toyoda, *RIKEN SciNeS using RICC Supercomputer as Cloud Computing Infrastructure*, ASI-Yokohama Institute 連携フォーラム (2010).

以下からダウンロード可能
https://database.riken.jp/sw/view#cria42s2ria42s351i