B-007

ソフトウェア抽象化レイヤーの性能評価方法
Performance Evaluation of The Software Abstraction Layer

河井 裕* 岩井 剛 佐々木 節 渡瀬 芳行
Yutaka Kawai Go Iwai Takashi Sasaki Yoshiyuki Watase

1 はじめに

ヘテロな計算環境において、抽象化レイヤーを用いてソフトウェアのインターフェースを統一化する手法は必要とされる技術である。SAGA (A Simple API for Grid Applications)[5] は、異なるデータグリッドを利用するための統一的なインターフェースを提供するが、オーバーヘッドによるコストの影響を考慮する必要がある。

ソフトウェア抽象化レイヤーの性能評価を行うために留意しなければならない点は、比較対象になる既存のソフトウェアが性能評価に適したものであるか否かを判断することである。本論文では、SAGA ベースのアプリケーションによって生じるオーバーヘッドを評価した際に直面した評価方法における問題点を示し、その解決方法を述べる。

2 SAGA アプリケーション例

高画質の写真や動画などの大きな容量を必要とするファイルにとって有効な手段の一つして、ファイルを分割して異なるストレージに分散させて保存する方法が考えられる。今回の例では、分割したファイルを異なるデータグリッド上に保存している。具体的には、ファイルを 3 つに分割し、それぞれを iRODS (The integrated Rule-Oriented Data System)[2, 3]、Gfarm (Grid Data farm)[1]、そしてローカルファイルシステムに保存するようにした。図 1 は bubble chamber の写真を分割保存した例を示している。

作成した SAGA アプリケーション “img_cat_saga” コマンドは、分散したデータグリッド上のファイルを連結して標準出力にデータの中身を表示することができる。

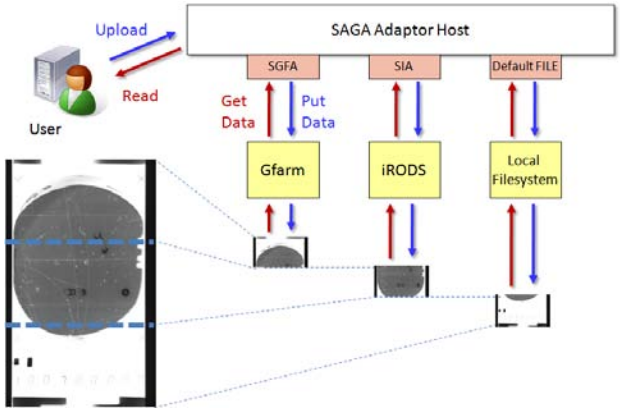

図 1. File Access to separate pieces of a photograph via SAGA.

このコマンドに、分割されたファイルのデータグリッド上の位置情報が記述されたテキストファイルを読み込ませることによって、分散しているファイルを集めることが可能になる。各ファイルのそれぞれの位置情報は、RFC1630[4] に準拠した SAGA URL で記されている。

3 性能評価方法の問題点

3.1 テストケース

扱うファイルの大きさを 100MB から 1GB まで 100MB 単位で変化させるとする。テストの準備として、各ファイルを 3 つに分割し、それぞれを iRODS, Gfarm, ローカルファイルシステムの 3 つの異なるファイルシステムに保存することとする。テストの方法は、分割されたファイルの断片を別々のファイルシステムから読み込んでから標準出力に表示するまでに要する所要時間を測定する

*Computing Research Center(CRC), High Energy Accelerator Research Organization (KEK)

Copyright © 2011 by Information Processing Society of Japan and
The Instiute of Electronics, Information and Communication Engineers
All rights reserved.

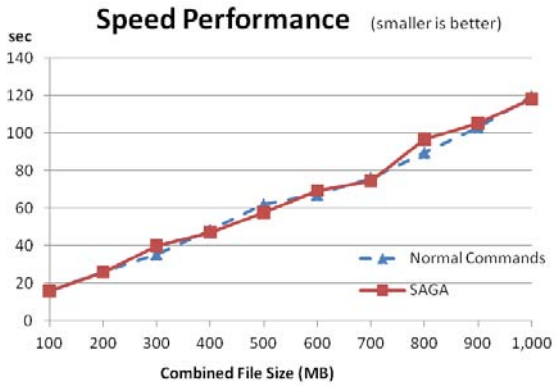

図 **2. Speed Performance Test Results**

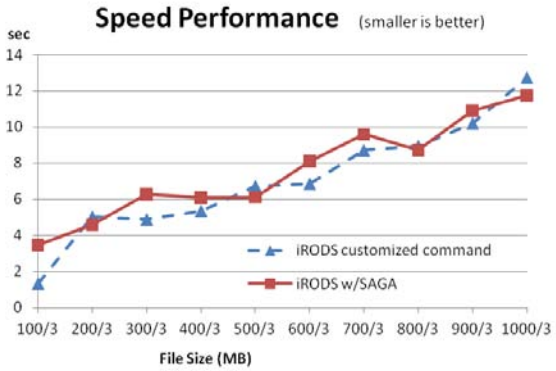

図 **3. customized iget VS SAGA iRODS**

こととする。テストケースは、従来の個別のクライアントコマンドを使用するケースと、SAGA アプリケーションを用いるケースとする。

3.2 テスト結果

テストの結果を図 2 に示す。それぞれのケース共に、ほぼ右肩上がりの傾向になっている。しかし、SAGA アプリケーションのケースと、従来のコマンドを使用するケースとは、ほぼ変わらない結果になっているのがわかる。本来であれば、ソフトウェア抽象化レイヤーである SAGA にオーバーヘッドのコストがかかると考えられる。しかし、今回のテストの結果では、全くコストがかかっていないことが示されている。以上の結果から、今回のテストの評価の方法に問題があったと考えられる。

4 比較対象の是正

データグリッドのクライアントコマンドにおいても Linux の標準コマンドにおいても、運用性と機能性の点で完成度が高いが、そのためにコマンド自体がリッチになっている可能性がある。一方、SAGA アプリケーションは、各クライアントの API を直接使用しており、余計な機能などが入っていない。

そこで、SAGA アプリケーションで使用している API のみで構成した iRODS のコマンド (iget) を新たに作成したところ、SAGA を使用するよりも比較的速い結果が得られた (図 3)。ローカルファイルシステムにおける cat コマンドに関しても同様の結果が得られた。したがって、SAGA のオーバーヘッドのコストを測るためには、新たに作成したコマンドを比較対象にした方がより適切であると言える。

5 結論

近年において、抽象化レイヤーを用いてソフトウェアのインターフェースを統一化する手法は、ますます必要とされる技術になってきている。本論文では、ソフトウェア抽象化レイヤーの性能評価を行うために必要な手法を示した。ソフトウェア抽象化レイヤーの例として、異種グリッドミドルウェアを抽象化する SAGA を用いて性能評価の結果が比較対象の設定によって変化することを示し、従来の既存コマンドを単純にそのまま比較対象にする場合の不十分さに言及した。オーバーヘッドのコストを正確に把握するためにも、適切な比較対象を設定することが必要と考えられる。

参考文献

[1] Gfarm – Grid Data Farm. Online. `http://datafarm.apgrid.org/index.en.html`.
[2] iRODS – the Integrated Rule-Oriented Data System. Online. `http://www.irods.org`.
[3] A. Rajasekar, M. Wan, R. Moore, and W. Schroeder. A Prototype Rule-based Distributed Data Management System. In *Proc. HPDC workshop on "Next Generation Distributed Data Management"*, Paris, France, May 2006.
[4] RFC1630 – Universal Resource Identifiers in WWW. Online. `http://www.ietf.org/rfc/rfc1630.txt`.
[5] SAGA: A Simple API for Grid Applications. Online. `http://saga.cct.lsu.edu/`.

Copyright © 2011 by Information Processing Society of Japan and
The Instiute of Electronics, Information and Communication Engineers
All rights reserved.