

参照局所性に着目した分散ハイパーリンクストアの研究

鳥居 孝行[†] 古瀬 一隆[‡] 陳 漢雄[‡]筑波大学情報学群情報メディア創成学類[†] 筑波大学システム情報系[‡]

1. はじめに

近年、Web データは増加を続けているため検索エンジンの必要性が高まっている。検索エンジンでは PageRank[1]に代表される Web ページ間のハイパーリンク構造 (Web グラフ) を解析するランキングアルゴリズムが使用されるが、2007 年に Najork により、SALSA アルゴリズム [2]の検索結果が他より優れるという報告がされている[3]。しかし、SALSA アルゴリズムは検索時に検索語句に関連する部分グラフ (隣接グラフ) を抽出し順位計算を行うため、グラフ抽出にかかる時間が問題となる。

そこで本研究では、複数の計算機に Web グラフを分散して保存する「分散ハイパーリンクストア」を構築し、SALSA アルゴリズムをはじめとするランキングアルゴリズムの計算高速化を図る。

2. 関連研究

2009 年に Najork により Scalable Hyperlink Store が発表された[4]。このシステムでは、Web グラフを圧縮しネットワークで接続された複数のサーバーのメモリ上に分散配置することにより、高速なアクセスを可能にしている。各ノード (ページ) のデータは、URL のホスト部分が同一のものが同じサーバーへ配置されるように分散され、データを冗長にディスクに保存することにより障害にも対応する。また、データの増加に応じてサーバーの追加が可能であり、スケールアウト性能も備えている。

しかし、関連研究にはいくつかの問題がある。まず、URL のホスト名による分散はランキングアルゴリズムの計算にとって最適とは言えない。また、サーバーを追加する度にノードからサーバーへの対応関係が増え、ノードの探索時間が

増加することや、障害発生時にディスクからメモリへデータをロードする間、システムが停止するという問題もある。

本研究ではこれらを踏まえ、ランキングアルゴリズムの特性に最適化したノード分散を実現し、また実装に汎用分散データストアを利用することで、より柔軟なスケールアウト、対障害性能を備えたシステムを提案する。

3. 提案手法

3.1. 参照局所性を考慮したデータ分散

SALSA アルゴリズムでは、テキスト解析型検索エンジンから検索語句に関連するノード集合 (ルートセット) を取得する。そして、ルートセット中のノードとリンクでつながるノードを探索して隣接グラフを抽出し、順位計算を行う。一般にルートセットに選ばれるノードは特定のトピックと強い関連を持っていることから互いにリンクされていることが多く、密な部分グラフを形成する傾向が強い。

そこで、Web グラフに対して予めクラスタリングを行い、密な部分グラフのデータ分散を一致させることにより、隣接グラフ抽出にかかる時間の短縮や通信量の低減が期待できる。

3.2. 汎用分散データストアでの実装

汎用分散データストアとは、データを複数の計算機に分散させてスケールアップや対障害性能の向上を図るシステムであり、分散方法やデータ構造により様々な種類が存在する[5]。実装にこれらの汎用分散データストアを利用することで分散機能をシステム本体から独立させ、より柔軟なシステムの構築が可能となる。

本研究では、扱えるデータ構造の柔軟さとカスタマイズ性の高さから Apache Cassandra[6]を採用した。Cassandra は複雑なキー・バリュー型のデータ構造を扱うことができ、キーに基づいてデータ分散を行う。そのため、クラスタリングにより求めた各クラスタの ID をキーとして、それに属するノードをまとめて保存することで、クラスタごとのデータ分散を一致させることが

Study on the distributed hyperlink store focused on locality of reference

Takayuki TORII[†], Kazutaka FURUSE[‡], and Hanxiong CHEN[‡]

[†]College of Media Arts, Science and Technology, University of Tsukuba

[‡]Faculty of Engineering, Information and Systems, University of Tsukuba

可能となる。

3.3. データ構造

ノードは URL によって一意に決定されるが、URL の状態ではアルゴリズムの計算には向かないためノード ID に変換する。そのため、ノード ID と URL を相互変換する機能がシステムに求められる。リンクについては、あるノードのインリンクとアウトリンクを高速に参照できる必要がある。

また、クラスタリングの際にノードの重複を許すかどうかは重要な選択となる。重複を許さない場合、ノードからクラスタへの対応関係が 1 通りとなるためノード ID 中にクラスタ ID を埋め込むことが可能となり、所属クラスタの解決を通信なしで実現できる。一方、重複を許す場合はノードがどのクラスタに属するかを問い合わせる必要があるが、より柔軟なクラスタリングを行うことができる。

本研究では、以下のデータ構造 (Cassandra におけるカラムファミリ) を用いて、上述した機能とクラスタリングによるデータ分散を実現する。

- **UrlToNodeId**

URL をキーとしてノード ID を保存する。重複を許すクラスタリングの場合はクラスタ ID も保存する。

- **Node**

クラスタ ID をキーとしてそのクラスタに属するノードの URL, インリンク, アウトリンクを保存する。

- **NodeIdToClusterId**

重複を許すクラスタリングの場合のみ使用し、ノード ID をキーとしてクラスタ ID を保存する。

4. 実験

スイッチで接続された 5 台の計算機それぞれに Cassandra プロセスを 1 つずつ起動した環境で評価実験を行った。実験には、ノード数 10 万、リンク数 100 万の合成データを使用した。なお、ノードのインリンク数とアウトリンク数はそれぞれ Zipf 分布、一様分布に基づいている。

クラスタリングには、クラスタ内のノード数が一定数になるまで、ノードのリンクを探索しクラスタに追加していく単純な方法を用いた。

図 1 は、隣接グラフの取得速度を表している。simple, single はそれぞれ、ノード ID をキーにした単純な実装、重複を許さないクラスタリングでの実装を示す。また、ルートセットはリンク

を共有するノードを 3 つ選択した。これより、隣接グラフのサイズが大きくなるとクラスタリングにより取得時間が短縮できることがわかる。

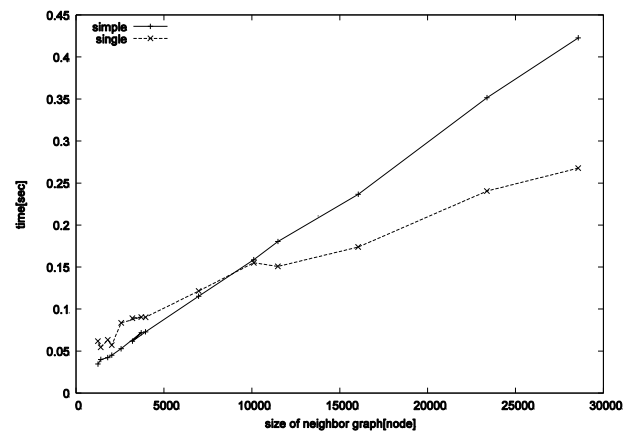


図 1: 実験結果

5. まとめ

本研究では、ランキングアルゴリズムにおける Web グラフへのアクセス高速化を目的とする、参照局所性に着目した分散ハイパーリンクストアを提案した。クラスタリングを用いて、データ分散の最適化を行い、実装に汎用分散データストアを利用することで柔軟なシステムの構築を行った。

実験より、クラスタリングによって SALSA アルゴリズムにおける隣接グラフ取得速度が向上することを確認した。

今後は、より高度なクラスタリング方法を検討し、クラスタリングがデータサイズやスケールアップ性能にどのような影響を及ぼすかを調査したい。

参考文献

- [1] Page, L., Brin, S., Motwani, R. and Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web, Stanford Digital Libraries Working Paper (1998).
- [2] Lempel, R. and Moran, S.: SALSA: The Stochastic Approach for Link-Structure Analysis, ACM Trans. TOIS, Vol.19, Issue 2, pp.131-160 (2001).
- [3] Najork, M.: Comparing the effectiveness of HITS and SALSA, Proc. CIKM '07, pp.157-164, ACM (2007).
- [4] Najork, M.: The Scalable Hyperlink Store, Proc. HT '09, pp.89-98, ACM (2009).
- [5] Cattell, R.: Scalable SQL and NoSQL data stores, ACM SIGMOD Record, Vol.39, Issue 4, pp.12-27 (2010).
- [6] Apache Cassandra, <http://cassandra.apache.org/>