

LDAPにおけるメタデータ管理方式

川戸貴之[†] 田胡和哉[‡]東京工科大学大学院 メディア・情報工学科[‡]

1 はじめに

近年、ネットワークサービスの増加に伴い、ネットワーク上に分散する様々な資源をメタデータとして一元管理するLDAPディレクトリサービスの応答性能向上が求められている。

本研究では主記憶装置の低価格化を鑑み、LDAPサーバの仮想メモリにデータベースを全てロードすることで応答性能の向上を行った。

この際LDAPサーバが管理する資源へのアクセスが時間的な偏りを持つことに注目し、アクセスパターンのクラスタリング分析に基づいたページ配置を行うことでページングを効率化する技術を開発し、応答性能の向上を行った。この技術は、全ての情報を仮想メモリにロードできる環境上の様々な技術において転用可能である。提案方式の概要を述べる。

2 関連研究

2.1 アクセス特性に合わせたキャッシングを行う研究

LDAPサーバへの検索要求の種類は3種類あり、各検索要求で検索された情報は検索要求別に再利用性の高いものと低いものが存在する。これに注目し、検索要求別にLRUリストを管理することで再利用性の高い情報をメモリ空間に残すことでキャッシング性能を向上させ性能改善を図った研究がある。

2.2 キャッシュする情報を選別しキャッシングする研究

LDAPサーバはエン트리という単位で情報を管理し、エン트리の中には属性という情報を格納している。従来は1エントリをキャッシュする場合、格納している全ての属性をキャッシュしていたが、必要な属性のみをキャッシュすることによりメモリ空間を節約し性能改善を図った研究がある。

3 提案機構

エントリのアクセスには時間的な特性がある。例えば午後5時にしか動作しないシステムのエントリが存在する場合である。ここに着目し、特性が似たエントリを仮想メモリ空間上に纏めて配置する(図1)アルゴリズムをLDAPサーバ上に実装する。

従来のキャッシング方法では1エントリが2ページに渡っていることにより、2ページ目が必要のない情報を含んでしまいLRU方式により必要のない情報がメインメモリに残り続けることが頻発しキャッシングの効率を損なっていた。

本提案手法は、グループ単位でロードすることで、必要な情報は残り続け不必要になればグループ単位で不必要になるためLRUの影響を受けずページアウトが可能となる。

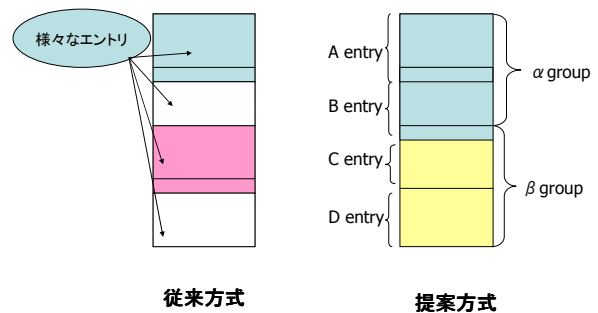


図1 ページの状態

Fig.1 page state

3.1 想定したシステム

本研究で想定したシステムはLDAPサーバが管理するエントリデータの総容量が主記憶の総容量に近く、ページングが発生するシステムとした。

3.2 実装環境

記述言語 : C 言語

LDAP のバージョン : OpenLDAP 2.4.23

3.3 実装

本研究では LDAP サーバがキャッシュ動作を行う部分を変更することで実装した。図2は LDAP の動作の図である。通常は検索リクエストを受けるとキャッシュ済みかを確認し、キャッシュ済みでなければ DB を検索し、ヒットしたものをキャッシュし、Index tree に登録をする。[A]は従来ではキャッシュ要求がくるたびに、仮想メモリ空間を確保しキャッシュを行っていた。しかし提案手法では[A]の部分で、グループごとの仮想メモリ領域を確保している。この動作はサーバ起動時に行われ以降のキャッシングは確保した仮想メモリへの格納のみとなる。

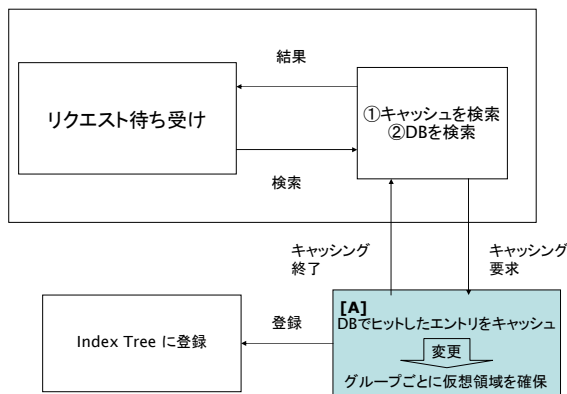


図2 実装

Fig.2 Implementation

4 性能評価

4.1 実験環境

CPU : Intel Core2Duo 2.3GHz

Memory : 512Mbyte

OS : Debian etch

Kernel version : 2.6.26

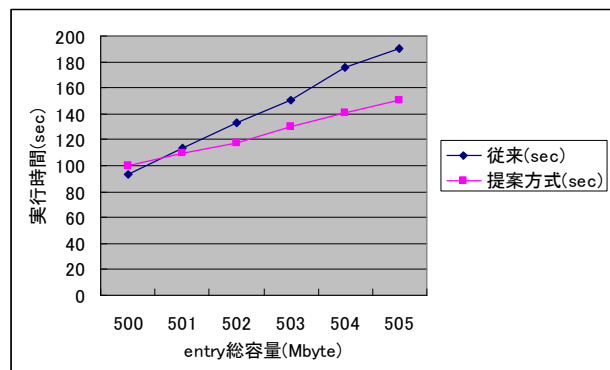
4.2 評価方法

ランダムなエントリの検索リクエストを2つのグループに対し、7 : 3の割合で送信し、その応答時間を計測する。本研究での総リクエスト数はマシン性能の都合上10回のリクエストとした。

4.3 計測結果

表1 計測結果

Table 1 Measurement result



計測結果から従来の LDAP サーバより 20%ほど応答性能が向上しているのが確認できる。しかしメインメモリを越える情報が少なくなるほど効果が出ていないことも見て取れる。

5 考察・まとめ

本論文では、LDAP サーバの応答性能改善を目標に、特性の似たデータを連続的にメモリへ配置する手法を提案した。提案手法はシステムのメインメモリをキャッシュする情報が少しでも超える場合に、LDAP サーバの応答速度が従来手法より 20%程度改善され本手法が応答速度改善に有効であることが分かった。

6 今後の課題

6.1 クラスタリング手法

エントリをグループ化するためのクラスタリング手法が必要である。

6.2 様々な条件下での測定

本論文では1エントリサイズや、1ページの容量、評価のグループに対するアクセス割合などが固定されているため、条件を変えて有効性を測る必要がある。

7 参考文献

- 1) S.Kikuchi et.al, High Performance Caching Techniques for LDAP Directory Servers, Information Processing Society of Japan, pp. 49-54 (1998)
- 2) Wang et.al, Measurement and Analysis of LDAP Performance, IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 16, NO. 1, FEBRUARY 2008
- 3) Apurva Kumar, The OpenLDAP Proxy cache
- 4) Wahl et.al, Lightweight Directory Access Protocol (v3), RFC 2251 [December](#) 1997
- 5) Wahl et.al, Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions, RFC 2252 [December](#) 1997