テーマ名：従来のシステムと比べて柔軟性の高いkey-value storeの開発

申請者名：千々和 大輝

従来のランダムアクセスに向いたkey-value storeに比べ、sortedなkey-value空間を持つkey-value storeではget/putだけでなく、「AからBまでの要素数の取得」や「Aから数えてN個の要素を取得」などの応用ができると思われる。さらにvalueによる範囲検索が可能となることにより、それらの応用がさらに強力になると期待できる。よって本プロジェクトでは、valueによる範囲検索の実現及び自律的な分散エージェントによる拡張機構を実装し、従来のシステムと比べて柔軟性の高いkey-value storeの開発を目標とする。

１．なにを作るか

近年、クラウドコンピューティングにおける基盤技術として、key-value storeが注目されている。このシステムはRDBと比べて大規模なクラスタ内での使用を前提としており、ノード数の増加に対応して自動的にスケールアウトするものがほとんどである。さらに、高速に動作する必要があるのでRDBほど処理が多様でない点も特徴的である。

提案者も今年の六月に、P2P構造化オーバーレイネットワークの一つであるSkip Graphを利用した簡易的なkey-value storeを並列プログラミング言語Erlangで実装している。このシステムの特徴は、大規模クラスタ内で効率的な範囲検索を行うことができるという点である。これはSkip GraphがChordなどのDHTとは異なり、ピアがkeyによってソートされているという性質を利用して実現している。

しかし一般的なkey-value storeは単一keyのgetとput、deleteのような既に実装されている機能しか行うことができず、とても柔軟性があるとは言い難い。そこで本プロジェクトでは、以前作ったシステムをC++で書き直して十分に高速化し、柔軟性向上のため以下の2つの新しい機能を備えたkey-value storeを開発する。

・valueによる範囲検索(*図1*)

通常のkey-value storeでは、あるvalueに対応するkeyを知るという処理はほとんど無意味であることが多い。しかしSkip Graphのような範囲検索が可能な構造化オーバーレイの場合、valueの範囲を指定してkeyを知るという処理はとても有用であり、例えば「1001から2000の範囲内にある値を全て取得する」などが可能になる。

ただこの機能を実装すると、valueが頻繁に変更される環境の場合ピアの入れ替わりが激しくなるので、その際に実行される排他的制御アルゴリズムが遅いと全体の速度が低下してしまう。そこで本プロジェクトでは、1hop先のノードを保持することにより効率的なleave/addの実装も試みる。

・自律的な分散エージェントによる拡張機構(*図2*)

この機能とSkip Graphの連携により、「AからBまでの範囲内にある要素数を知る」や「AからBまでの範囲からN個の要素を取り出す」などの機能をユーザが容易に拡張することが可能となる。各エージェントはLLVM上で動作する予定であり、ネットワーク効率も考慮してそれらのキャッシュや差分処理も実装する。

また、Skip GraphはDHTに劣らない検索効率でありながら範囲検索も行うことができるというとても優秀なアルゴリズムであるが、何も考えずに実装すると物理ノード間のデータ分布が偏ってしまうという欠点が存在する。しかし、これはデータのput時にDHTを利用してノードを選択することにより解消できると提案者は考えた。よって本プロジェクトでは並行してデータ分散基盤としてのDHTの実装も行う(*図1*)。

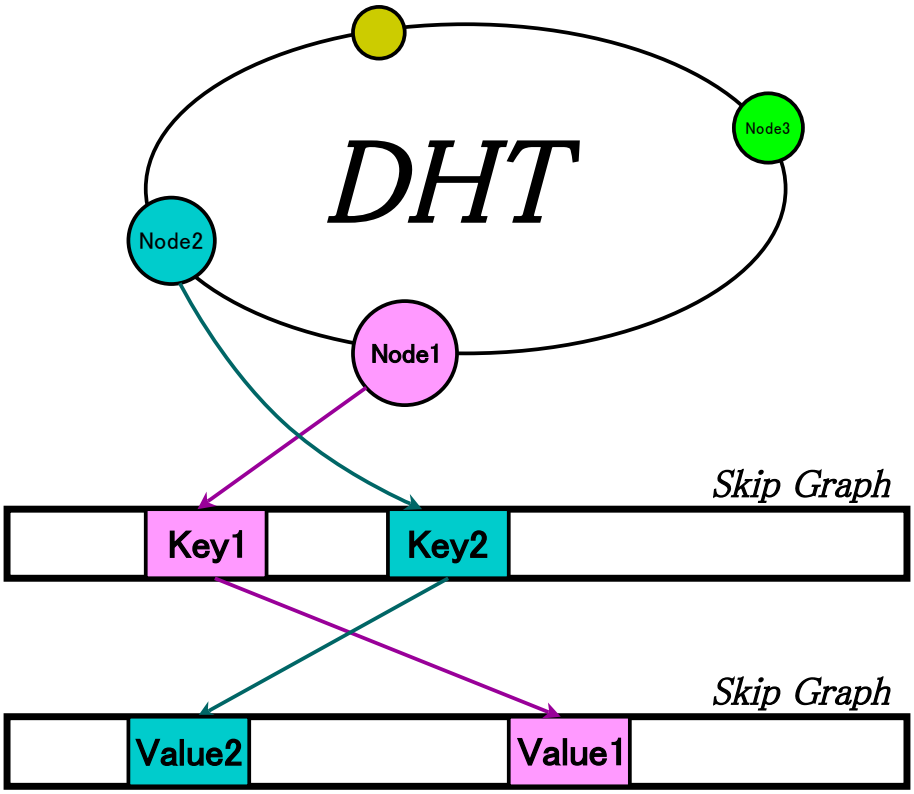

図1：valueによる範囲検索，データ分散基盤としてのDHT

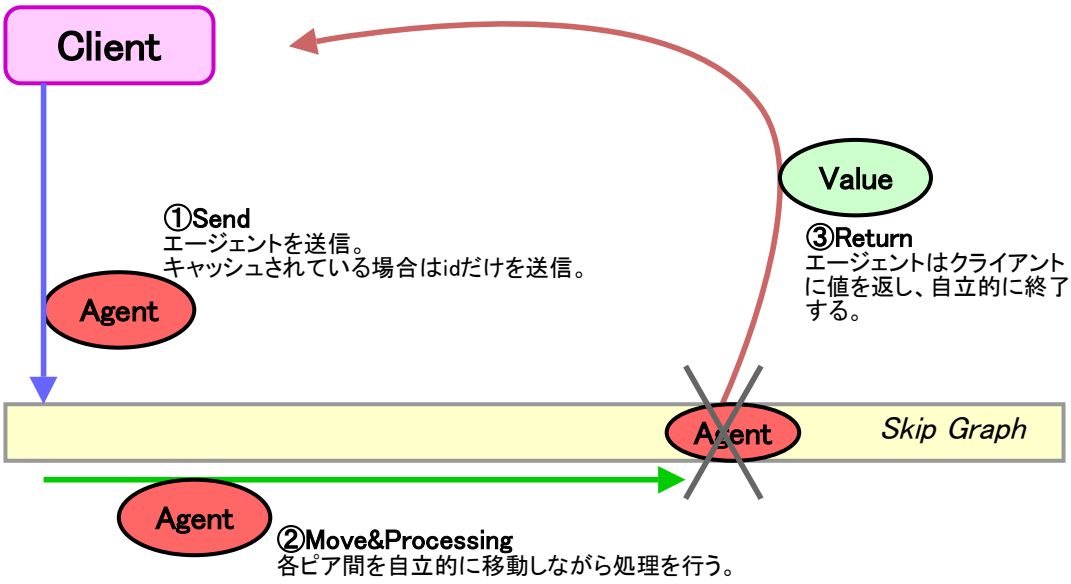

図2：自立的な分散エージェントによる拡張機構

２．どのような出し方を考えているか

オープンソースソフトウェアとしての公開を予定している。ライセンスは特に考えていないのでとりあえずBSD Licenseとするつもりであるが、今後さらに緩いライセンスに変える可能性もある。

３．斬新さの主張、期待される効果など

・valueによる範囲検索

範囲検索のできる key-value store は少なからず存在すると思われるが、valueによる範囲検索を実現している例は提案者の知る限りでは存在しない。

この機能を提供することで様々な応用、例えば「要素Aの値からBの値の範囲内で値を取得する(図3)」などが可能になり、ユーザが柔軟な範囲検索を行えるようになる。

一つのkeyに複数のvalueを対応づけることも可能で、具体的なシチュエーション例としては「コンテンツ検索においてサイズの範囲を指定」や「タグを指定して検索」などが挙げられる。

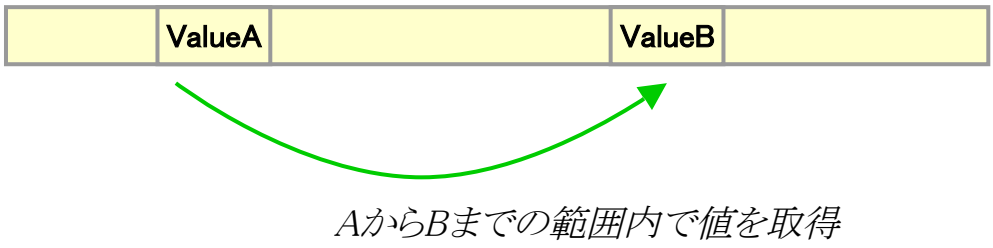

図3：要素Aの値からBの値の範囲内で値を取得する

・自律的な分散エージェントによる拡張機構

この機能も同様、提案者の知る限りでは分散エージェントによる拡張機構を備えたkey-value storeは存在せず、斬新であると言える。

これを提供することで従来のkey-value storeにおけるget/putに収まらず、とても柔軟な処理を効率的に行うことができるようになる。ネットワークトラフィックがネックになると予想されるが、エージェントのキャッシュ機能や差分処理の実装を行うことにより改善できると思われる。

さらにvalueによる範囲検索と組み合わせることで、「要素Aの値よりも小さい要素をN個取得する(図5)」や「要素Aの値が上から何番目かを知る」などを容易に実現できる強力な柔軟性をユーザに提供することができ、開発効率の向上に繋がると期待できる。

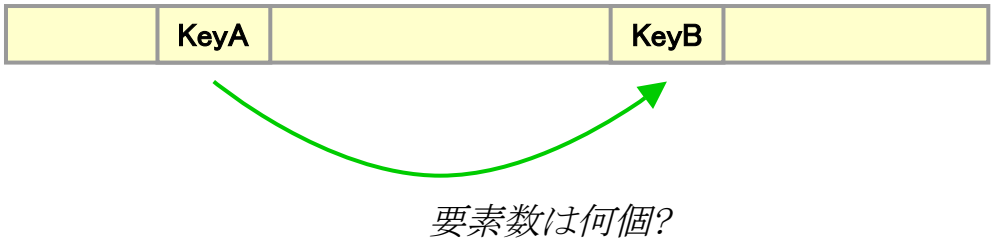

図4：エージェントを使って要素数を知る

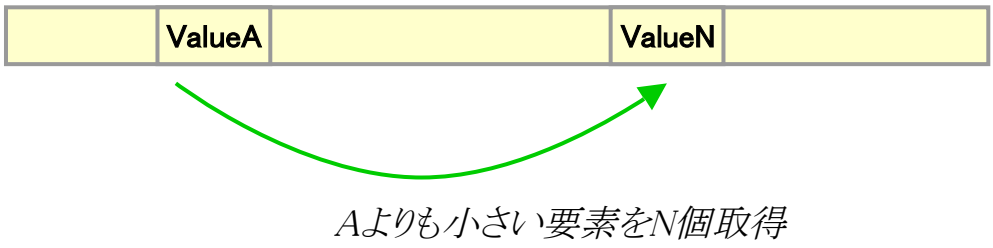

図5：Valueによる範囲検索と組み合わせる

４．具体的な進め方と予算

(1) 主に開発を行う場所

主に家で開発を行う。

(2) 使用する計算機環境(ハード、OS)

ハードはThinkPad、OSはUbuntu Linux上で行う。

(3) 使用する言語・ツール

DHT実装はErlang、それ以外はC++を使用する予定。

(4) 共同開発者がいる場合は、作業の分担

なし。

(5) (もしあれば) ソフトウェア開発に使う手法

特に決めてないが、テストファーストで開発を進める予定。

(6) 開発線表―― いつまでになにをやるか、手順と時期(月単位) を明確に書いたもの

採択ー１月：　以前Erlangで実装したSkip GraphをC++で書き直し、高速化をはかる。

２月ー３月：　valueによる範囲検索(C++)、DHT(Erlang)を実装する。

４月ー６月：　自律的な分散エージェントによる拡張機構(C++)を実装する。

7月：　各種言語用にライブラリを作成する。

(9) 購入する備品とその使途

なし。

(10) 予算内訳をまとめた表

人件費	¥2300000
旅費	¥200000
プロジェクト管理組織費用	¥500000
計	¥3000000

５．提案者の腕前を証明できるもの

・Skip Graph in Erlang

P2P構造化オーバーレイネットワークの一つであるSkip GraphのErlangによる実装で、複数マシンを利用してスケールアウト可能である。keyにatomを指定することもできるので柔軟な範囲検索を行うことができ、1マシン6万ピアでの高速な範囲検索を確認している。ソースコードはgithub.comにて公開中。

http://gist.github.com/131918

http://b.hatena.ne.jp/entry/http://d.hatena.ne.jp/daiki41ti/20090619/p1

・Cradle

Consistent HashingをErlangで実装したもの。レプリケーションなども実装しており、突然あるノードが落ちても一瞬でネットワーク再構成/データ復旧することが可能な程度の耐障害性が特徴。これもソースコードはgithub.comにて公開している。

http://github.com/daiki41ti/Cradle/tree/master

・JIT-Brainfuck

JITコンパイル実装の習作としてC言語で作ったプログラム。+-<>の機能は実装しているが、IO処理(.,)の実装はアセンブリを深く勉強していなかったので諦めてしまった 。github.comでソースコードを回覧することができる。

http://gist.github.com/118559

・Ruby 1.9 の GC における効率的なメモリ管理の実装

セキュリティ&プログラミングキャンプ2009にて鹿児島高専3年の稲付智昭さんと一緒に行ったプロジェクトで、madvise(2)を使用してRuby1.9のGCのメモリ効率を改善する。

http://www.slideshare.net/daiki41ti/ruby-19-gc

・Pure P2P 匿名チャットシステム

中学生(14歳)のときに作ったPythonプログラム。PureP2Pなので中央サーバが必要無く、多段プロクシ機能を実装しているので少人数(500人以下)であればリアルタイムなチャットを匿名で行うことができる。ソースコードは紛失してしまったが、ブログで公開していたのでWebキャッシュサービスなどで探せば見つけることができるかもしれない。

６．プロジェクト遂行にあたっての特記事項

なし。

７．ソフトウェア作成以外の勉強，特技，生活，趣味など

高校に在学しているので、普段は宿題や勉強などに取り組んでいます。ペン回しが趣味で、インターネット上の仲間とともに自分達で撮った動画を公開し合いながら技術を向上させています。

８．将来のソフトウェア技術について思うこと・期すること

現在、Web上では様々な面白いサービスが生まれており、世界中に大きな影響を与えつづけている。中でもAjaxやFlashなどの技術を利用したサービスはデスクトップアプリケーションとほとんど大差無く感じる。さらに近年クラウドコンピューティングという概念が爆発的に流行し、今最も注目されている物の一つであろう。

ただ、やはりWebには致命的な欠点もある。それは「データを向こう側のサーバが全て管理している」ということだ。これは既知の通りセキュリティ上の観点からも同様に言えることだが、将来的な観点から見ても非常に危険である。例えば、最近Amazon EC2やGoogle App Engineなどの便利なプラットフォームを利用したサービスが増えているが、(まず確率的に低いと思われるが)急にGoogleやAmazonがそのサービスを停止することを決定した場合、大多数のサービス運営者やそのユーザが混乱してしまうことは容易に想像することができる。

しかし、一重に欠点であると主張することは乱暴であるとも考えている。なぜなら、クラウドコンピューティングという概念が示している通り、データをクライアント側に置く必要が無いということはモバイル機器からも容易に利用できることに繋がるからだ。

このトレードオフを念頭に置き、今後どのような形態でサービスを提供していくべきかを考えることが今後のIT技術の課題になってくるだろうと予想する。

一つの手段としてP2P技術を応用することも提案されておりとても面白い分野だと思うので、今後さらに知識を深め、技術を向上させていきたい。