首藤一幸, 華井雅俊, 杉野好宏:
"分散システムの大規模シミュレーション" (依頼講演),
BI-5. 仮想化ネットワークのテストベッドとその応用 (依頼シンポジウムセッション),
電子情報通信学会 2013年総合大会 講演論文集,
2013年 3月 19〜22日

分散システムの大規模シミュレーション

Large-Scale Simulation of Distributed Systems

首藤一幸 Kazuyuki Shudo
華井雅俊 Masatoshi Hanai
杉野好宏 Takahiro Sugino

東京工業大学
Tokyo Institute of Technology

1 はじめに

すでに、インターネット上では100万台以上のPCからなる自律分散システムが少なくとも3つ稼働しており [4]、最大のものは1,000万台に達している [8, 5]。インターネットに接続される機器は多様化しつつ増大を続けており、多くの予想が、その、Internet of Thingsの規模が2020年には数百億に達すると見ている。これを想定すると、今後の分散システム研究は、この数百億という規模を対象としなければならない。しかし、現在手に入る実験手法・手段では、10万〜100万ノードという規模が限界である（表1）。

表1 各実験プラットフォームで可能な実験の規模

実験プラットフォーム	ノード数	
本研究のターゲット	**億〜**	インターネット, 脳, 経済, 等々を模擬
Overlay Weaver (著者ら) [12]	約100万	1台上で通信遅延とパケットロスを模擬
PeerfactSim.KOM (TU Darmstadt) [13]	10万以上 [13]	推奨は1万程度
OverSim (U. Karlsruhe) [1]	10万 [1]	1台上で遅延を模擬
p2psim (MIT) [10]	1万 [1]	同上
peeremu (NEC) [7]	1120 [7]	PC 14台で遅延とパケットロスを模擬

つまり、我々は現在、現れつつある規模（億〜）はおろか、すでに稼働している規模（〜1,000万）すら実験できないという状況を迎えている。設計・実装しdeployしようとする分散システムが、インターネットにどの程度の負荷をかけるのか、何かを契機に通信量が爆発的に増えて···といったことがないのか、などを調べる実験手段を、今、我々は持たない。

そこで我々は、億を超える規模のシミュレーションを可能とする手法、および、ソフトウェアとしての実装法を研究している。

2 汎用分散処理システムを用いたシミュレーション

我々は、これまでの研究にて、PC 1台での約100万ノードのシミュレーションを達成した（表1）。さらに上の規模を狙って、複数台でシミュレーションを行う機能を開発・実験してきたが、いくつかの問題にぶつかった：

- 性能
  ノード間のメッセージ通信がPCをまたいで行われるようになり、そこに時間がかかり、シミュレーションの進行速度が大きく低下した。
- 耐故障性・スケーラビリティ
  複数台のうち1台でも停止するとシミュレーションが止まる。1回のシミュレーションには数日かかることも普通なので、これでは数十台以上での動作には耐えない。
- ソフトウェア開発の困難さ
  シミュレータ自体が開発困難な分散システムとなった。そこに耐故障性を持たせることは、さらに困難である（チェックポインティング?）。

そこで今回、次の通り、これまでとは異なるアプローチを採ることとした。

ある種の汎用分散処理システムが、高いスケーラビリティと耐故障性、および、ソフトウェアとしての成熟を達成していることに着目し、それらを用いた大規模シミュレーション手法を研究、確立する（図2）。想定している汎用分散処理システムは、MapReduce [2] 処理系、グラフ処理系 [9, 6] である。これにより、上述した問題のうちいくつかが解決する：

- スケーラビリティ
  例えば、オープンソースのMapReduce実装Hadoopは4,500台のクラスタで稼働している [11]。
- 耐故障性
  例えば、Hadoopはタスク（ジョブの一部）の再実行機能を備える。一部のマシンが停止してもジョブを完遂できる。
- ソフトウェアとしての成熟
  分散システムのシミュレータより利用者がはるかに多く、それだけソフトウェアとして成熟しやすい。例えばHadoopは数多くの企業が使用している:Facebook, J.P.Morgan, ...。

性能、つまりシミュレーションの進行速度には、後述する課題（シミュレーション時間の取り扱い）を解決した上で取り組む。

シミュレーション規模は、マシン1台あたりのノード数に台数を乗じたものとなる。マシン1台あたりのノード数は、ノード情報とノード間メッセージをディスク上に持つのであればディスク容量、メモリ上に持つのであればメモリ容量によって制限される。Hadoopの場合、ディスク上に持つ。また、1台あたりのノード数が多かったり、メッセージが多くやりとりされると、シミュレーション進行速度が下がるので、それを考慮してマシン台数とシミュレート対象ノード数の比を決めることとなる。Overlay Weaverが数GBのメモリで100万ノードを扱

首藤一幸, 華井雅俊, 杉野好宏:
"分散システムの大規模シミュレーション" (依頼講演),
BI-5. 仮想化ネットワークのテストベッドとその応用 (依頼シンポジウムセッション),
電子情報通信学会 2013年総合大会 講演論文集,
2013年 3月 19〜22日

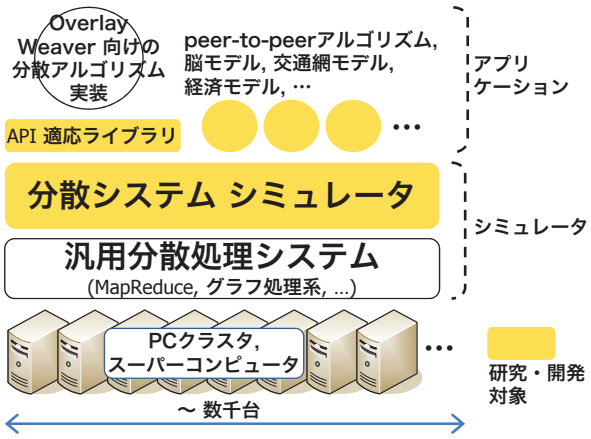

図 1　シミュレータの構造

えるので（表 1）、少なくとも、1 台あたり 100 万ノード程度を想定している。マシン 100 台を用いて 1 億ノード、1,000 台を用いて 10 億ノードのシミュレーションが可能になると見込んでいる。

3　手法

汎用分散処理システムを用いた分散システムシミュレーションには、次の課題がある。

3.1　ノードと通信の表現

汎用分散処理システムの上で、分散システムのノード、およびメッセージとその送受信を表現する必要がある。例えば MapReduce であれば、基本的なデータ表現形式である key-value ペアや、map、shuffle、reduce 処理で表現することとなる。

3.2　時間の取り扱い

シミュレーション上のノード間通信はマシン間通信のタイミングで行われる。MapReduce 処理系やグラフ処理系では、マシン間通信は全マシンが同期して一斉に行う（Bulk Synchronous Parallel モデル）。このため、ノード間通信の処理は、全マシンが一斉に行うこととなる。このままでは、任意のタイミングでのノード間通信をシミュレートできない。

この問題に対して、2 つの方法を試している。1 つは、同期の際に進めるシミュレーション時間上の時間幅を調整することで、シミュレーション速度と正確さのトレードオフを調整する手法である。もう 1 つは Virtual Time [3] である。これは、メッセージ受信処理を投機的に行ってしまい、もしそれより先に受信しておくべきメッセージが後から発覚した場合には、受信を取り消す、という手法である。時間を正確に扱える。

4　シミュレータの実装と実験

MapReduce 処理系 Hadoop を用いた実装 [15] と、Pregel [9] にならった自前のグラフ処理系を用いた実装 [14] を進めている。時間の扱いは Virtual Time で行う。

MapReduce 実装はシミュレート対象をディスク上に持つ。100 万ノード、何種類かのネットワークトポロジで Gnutella のフラッディングをシミュレートできており、ホップ数などの実験結果からトポロジごとの傾向が見えている。

グラフ処理系実装はシミュレート対象をメモリ上に持つ。DHT アルゴリズム Chord をシミュレートできており、1 万ノードで Chord が動作している。

5　まとめ

これまで（〜100 万）を大きく上回る規模（億〜）の実験を可能とするシミュレーション手法の確立、および、シミュレータの開発・提供を目指して、研究を進めている。シミュレータの最初の版が動作しつつあり、今後、スケーラビリティや性能を評価していく。

参考文献

[1] Ingmar Baumgart, Bernhard Heep, and Stephan Krause. OverSim: A flexible overlay network simulation framework. In *Proc. 10th IEEE Global Internet Symposium (GI'07)*, May 2007.

[2] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified data processing on large clusters. In *Proc. OSDI'04*, December 2004.

[3] David R. Jefferson. Virtual time. *ACM Transactions on Programming Languages and Systems*, Vol. 7, No. 3, pp. 404–425, July 1985.

[4] Raul Jimenez, Flutra Osmani, and Björn Knutsson. Sub-second lookups on a large-scale Kademlia-based overlay. In *Proc. IEEE P2P'11*, August 2011.

[5] Konrad Jünemann, Philipp Andelfinger, Jochen Dinger, and Hannes Hartenstein. Bitmon: A tool for automated monitoring of the bittorrent dht. In *Proc. IEEE P2P'10*, August 2010.

[6] U Kang, Charalampos E. Tsourakakis, and Christot Faloutsos. PEGASUS: A peta-scale graph mining system - implementation and observations. In *Proc. ICDM'09*, December 2009.

[7] Daishi Kato and Toshiyuki Kamiya. Evaluating DHT implementations in complex environments by network emulator. In *Proc. IPTPS 2007*, February 2007.

[8] Live Monitoring of the BitTorrent DHT. http://dsn.tm.kit.edu/english/2936.php.

[9] Grzegorz Malewicz, Matthew H. Austern, Aart J. C. Bik, James C. Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski. Pregel: A system for large-scale graph processing. In *Proc. OSDI'10*, October 2010.

[10] p2psim: a simulator for peer-to-peer protocols. http://pdos.csail.mit.edu/p2psim/.

[11] Apache Hadoop Project. Hadoop wiki - poweredby. http://wiki.apache.org/% linebreak[2]hadoop/PoweredBy.

[12] Kazuyuki Shudo, Yoshio Tanaka, and Satoshi Sekiguchi. Overlay Weaver: An overlay construction toolkit. *Computer Communications (Special Issue on Foundations of Peer-to-Peer Computing)*, Vol. 31, No. 2, pp. 402–412, February 2008.

[13] Dominik Stingl, Christian Groß, Julius Rückert, Leonhard Nobach, Aleksandra Kovacevic, and Rralf Steinmetz. PeerfactSim.KOM: a simulation framework for peer-to-peer systems. In *Proc. 2011 Int'l Conf. on High Performance Computing & Simulation (HPCS 2011)*, pp. 577–584, July 2011.

[14] 華井雅俊, 首藤一幸. 分散グラフ処理系を用いた大規模分散システムシミュレーション手法. 電子情報通信学会 技術研究報告 CPSY2012-27, pp. 109–114, August 2012.

[15] 杉野好宏, 華井雅俊, 首藤一幸. Mapreduce による大規模分散システムのシミュレーション. インターネットコンファレンス 2012 (IC2012) 論文集, pp. 21–27, November 2012.