WIDE クラウド WG 2015 年度活動報告

浅井 大史 * 小林 諭 * 島 慶一 † 関谷 勇司 *

2015 年 12 月 14 日

1 はじめに

WIDE クラウドワーキンググループは、今後のクラウド技術の研究開発を推進するために 2010 年 1 月に設立された。複数の WIDE 組織間に渡って運用される広域連邦型クラウドシステムである WIDE クラウドシステムの運用と、それを用いた研究開発を行っている。

2015 年度の主な活動は以下の通りである。

- 仮想計算機監視 MIB 策定 (第 2 章)
- システムログ解析技術 (第 3 章)

以降、それぞれの活動の詳細を報告する。

2 仮想計算機監視 MIB 策定

2.1 仮想計算機監視 MIB の目的

大規模な基盤システムを運用するには、整備された監視の仕組みが必要となる。仮想計算機の数が増加していく環境において、多数の仮想計算機の状態を外部の管理ドメインから把握することは重要であるが、現状において、統一した監視手法が確立されているとは言い難い。

WIDE クラウドワーキンググープでは、自身で仮想計算機基盤を運用した経験からこの問題に注目し、仮想計算機の監視に必要な要素の洗い出しと、それらの情報をやりとりするための標準化された手段が必要であると考えた。本活動は 2012 年の IETF での提案を発端とし、同時期に提案されていた他のグループからの類似提案を統合しつつ、WIDE プロジェクト主導により 2015 年 10 月に RFC7666 として発行に漕ぎ着けたことにより完了した。

*東京大学
†IIJ イノベーションインスティテュート

2.2 仮想計算機監視の問題点

仮想計算機の資源使用量を外部から監視することは一般的に難しい。その理由は大きく二つ考えられる。

一つ目は、管理範囲の問題である。仮想計算機は、特定の仮想計算機基盤事業者が運用するホスト計算機の上で動作しているものの、その内部については完全に仮想計算機の利用者のものであり、基盤事業者がアクセスしてよいものではない。これは、仮想ではない通常のホスティングサービスの運用を考えてみれば容易に想像できる。

ある計算機の資源使用量を外部から知るために利用される手法としてエージェントの導入が挙げられる。監視したい対象に特別なプログラムを導入し、ネットワーク通信によって状態を取得する。代表的なエージェントとして SNMP がある。この方式の問題点は、対象計算機に監視のための追加プログラムを導入しなければならない点にある。ホスティング事業者の運用支援のために、自分が運用している計算機に（自分からみれば不要な）プログラムを導入する障壁は高い。データ漏えいなど、セキュリティに対する不安もある。

実計算機であれば、これ以上の監視は困難であるが、仮想計算機基盤になるともうひとつ別の手法を利用できる。仮想計算機は、ホスト計算機上で動作しているエミュレーターの中で動作しており、仮想計算機に割り当てられる資源、もともとはホスト計算機の資源の一部を切り出したものであるが、を管理しているのはホスト計算機に他ならない。エミュレーターの内部情報には、仮想計算機がどれくらいの資源を利用しているのかの情報が蓄積されており、それを収集することで仮想計算機の監視が可能になる。

もちろん、エージェントを導入したときと全く同様の情報を入手することはできない。仮想計算機が内部で管理している資源については外から監視できないためである。代表的なものとして、仮想計算機のファ

表 1: 仮想計算機監視 MIB が提供する情報カテゴリ

カテゴリ	内容（例）
仮想計算機設定情報	名称、割り当て vCPU 数、割り当てメモリ量など
仮想計算機運用情報	起動中、停止中など
CPU 資源情報	起動時間、各 vCPU の利用時間など
ネットワーク資源情報	ネットワーク I/F 情報、通信量など
ストレージ情報	ストレージデバイス情報、IOPs、I/O 量など

イルシステムの利用率が挙げられる。ホスト計算機は、仮想計算機に与えた（最大）ディスク容量は把握できるが、割り当てられたディスクのうち、どの程度を実際に利用しているかは不明である。

二つ目はエミュレーション環境間での整合性の問題である。現在、様々な仮想計算機エミュレータが利用されている。オープンソーソでは QEMU[1]/KVM[2] や VirtualBox[3] などが代表的である。商用あるいは商用サポート付きのものでは、VMware[4] や Citrix の Xen[5] が市場を牽引している。それぞれ、内部的には類似の処理を実行し、管理している資源の種類や粒度も同等であるが、それらのデータにアクセスする手法が統一されておらず、監視システムを構築し辛い状況にある。外部に公開するデータの種類を共通のものとして定義し、またそのデータへのアクセス手段を標準化することでこの問題は解決できる。

2.3 仮想計算機監視 MIB の概要

前述の問題点を踏まえ、我々は仮想計算機監視 MIB で取得可能な情報として表 1 を選定した。より詳しい項目に関しては、RFC7666[6] を参照してほしい。

なお、仮想計算機の情報取得のための共通通信手段として SNMP を用いることとしている。

2.4 仮想計算機監視 MIB の実装

net-snmp のプラグイン（AgentX 形式）として virt-snmp が作成、公開 [1] されている。virtsnmp プラグインは libvirt[7] を用いた仮想計算機管理環境において、前節の情報を SNMP 経由で公開する。現在のところ、RFC7666 で策定されているすべての仕様に対応しているわけではない。特に、運用上重宝すると考えられる通知のメカニズムの実装が欠けており、今後の開発を検討している。

2.5 今後の進め方

仮想計算機運用基盤での資源監視の共通要素を洗い出し、エミュレーターの実装に依存しない情報取得方式を提案、標準化するという当初の目的は達成された。今後は、今回含めなかった仮想計算機関連情報の精査を進め、必要に応じて改定を検討することになる。また、仕様を完全に網羅した実装が現状存在しないことは相互接続性の実証という点で不完全であるため、我々が開発しているオープンソースでの実装を充実させるとともに、商用ベンダーでの採用を呼びかけていく必要がある。

3 システムログ解析技術

3.1 システムログ解析による異常検知

WIDE クラウドワーキンググループでは小林の手法 [8] を元にしたシステムログ解析技術の研究を実施している。本研究で言及するシステムログとは、UNIX 系システムで利用されている `syslog(3)` ライブラリを用いて出力されるテキスト文字列を指す。ただし、解析手法はログ出力ライブラリの実装方法とは独立しているため、記録された文字列が `syslog(3)` ライブラリから出力されるものと類似していれば、任意のシステムで出力されたログを汎用的に取り扱うことが可能である。図 1 にシステムログの例を示す。

システムログの文字列はログを出力するプログラムの設計者が自由に指定できるため、機械的に意味を汲み出すことはできない。またその種類にも制限がなく、新しいプログラムが開発されればそれに応じた新しい

[1] `https://github.com/scyphus/virtsnmp`

```
Dec  2 06:48:47 syslog-collect.wide.ad.jp
  nfcapd[1107]: Process_v9: Found options
  flowset: template 256
Dec  2 06:49:05 cloud-net2.wide.ad.jp
  named[29923]: error (unexpected RCODE
  SERVFAIL) resolving
  '21.212.27.200.in-addr.arpa/PTR/IN':
  200.27.2.7#53
Dec  2 06:49:18 vm2.komatsu.wide.ad.jp
  console-kit-daemon[2862]: GLib-CRITICAL:
  Source ID 1925981 was not found when
  attempting to remove it
```

図 1: システムログの例

文字列が生成される。逆に考えれば、粒度の細かい内部イベントの情報が記述されているとも言えるため、異常が発生した場合に出力される一連のログから状態を識別可能になると推測される。また、ログ文字列には設計者の意図が込められているため、異常発生時に関連して出力されたログを特定できれば、何が起きていたのかが類推できる可能性がある。

3.2 WIDE クラウドセグメントの解析

前述の通り、システムログは設計者による自由記述文字列となっているため、そのままでは解析には不向きである。文献 [8] では、まずシステムログを類似した文字列を持つもので分類し、分類された単位での解析を試みている。我々の研究活動の発端となっている小林の研究 [8] では Vaarandi による手法 [9] と Lafferty による手法 [10] が検討され、Lafferty の手法を採用していた。今年度はさらに水谷による手法 [11] を検討し、こちらの手法への移行を開始した。

分類されたログ（文献 [8] ではログテンプレートと呼ぶ）を時系列に並べたのちに単位時間での出現分布を算出する。ある一定期間の出現分布表を取り出し、ログテンプレート間の関連を尤度比検定関数を用いて判定することにより、その期間内でのログテンプレート間の因果グラフを導き出すことができる。さらに、Kalisch 等によって提案された手法 [12] を用いることで、より正確な因果関係のグラフを構築できる。

表 2: システムログ収集期間と収集元

期間	2015 年 9 月 26 日から 2015 年 11 月 16 日
収集元	WIDE クラウド基盤のハイパーバイザー (29 台)、WIDE クラウド基盤の NAT64 機材 (3 台)
ログ数	21,181,352 行

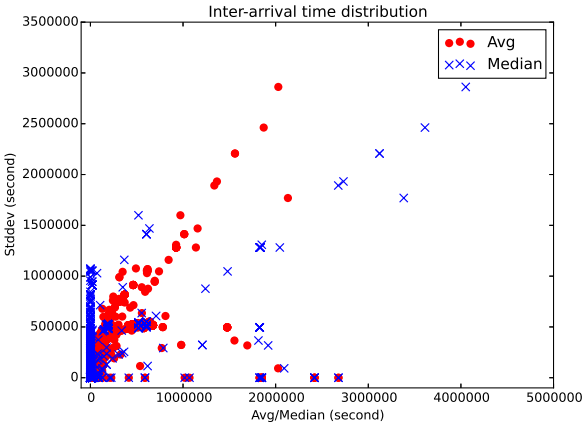

図 2: 各ログテンプレートの出現間隔の分散

これ等の手法を用いて WIDE クラウドを構成しているハイパーバイザーのシステムログの解析を試みた。解析対象とした期間、ログの収集元、ログの行数を表 2 に示す。

3.2.1 ログの分類

手法 [11] を用いてログを処理した結果、1147 種類のログに分類された。さらに、n-gram (n=3) を用いて 1147 種類のログを類似グループごとにまとめると 531 グループに分類できた。表 3 に分類結果の例を示す。ログテンプレート中の**は変動値であることを示している。また表 4 にグループの例を示す。この例では、named プログラムから出力された 8 つのログが類似ログとして一つのグループにまとめられたことを示している。

表 3: 分類されたシステムログの例

named[**]: client **: view world: query (cache) '**' denied
named[**]: error (unexpected RCODE **) resolving '**': **
CRON[**]: pam_unix(cron:session): session opened for user ** by (uid=0)

表 4: グループ化されたシステムログの例

named[**]: client **: view world: query (cache) '**' denied
named[**]: client 2001:200:0:1::**: view world: query (cache) '**' denied
named[32557]: client **: query (cache) '**' denied
named[32557]: client **: query '**' denied
named[2607]: client 116.213.132.50#58774: view world: notify question section contains no SOA
named[2607]: client fe80::216:3eff:feba:**: view world: query (cache) '**' denied
named[29923]: client fe80::216:3eff:feba:**: view world: query (cache) '**' denied
named[2607]: client 10.10.20.190#14732: view widecloud-normalvm: bad zone transfer request: 'jinmei.org/IN': non-authoritative zone (NOTAUTH)

3.2.2 ログの出現頻度の解析

分類されたログテンプレートごとに、平均出現間隔、中央値、標準偏差を算出した。図 2 に散布図を示す。図からわかる通り、ログの出現間隔は広範囲に渡っている。間隔が短いものは 1 分単位で継続するものもあれば、およそ 2ヶ月に渡る計測中に数度しか出現しなかったものもある。また、間隔の平均値と中央値に大きなずれがあることも観測されており、通常は短い間隔で現れているにもかかわらず、例外的に大きな出現間隔を持つものがあることを示している。

各ログテンプレートの出現間隔を詳細に見ていくと、いくつか興味深いものがある。図 3 は以下のログメッセージの出現間隔を描画したものである。

```
sshd[**]: warning: /etc/hosts.allow, line **:
  can't verify hostname: getaddrinfo(**,
  AF_INET) failed
```

このログメッセージは、1 日単位で周期的な出現をしているが、2015 年 11 月 6 日から 7 日にかけて、通常とは異なり、バースト的にログが出現していることがわかる。

また以下のログの出現間隔を図 3 に示す。

```
sshd[**]: refused connect from ** (**)
```

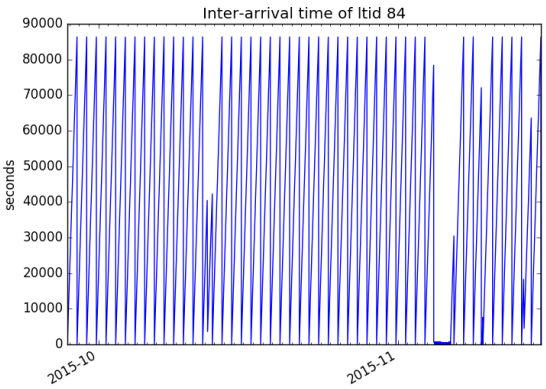

図 3: ログテンプレート番号 84 の出現間隔

こちらのログメッセージには周期的な動きは見られないが、2015 年 11 月 9 日から 15 日にかけて集中的に記録されている。どちらも sshd のアクセス認証に関するログであり、この期間になんらかの集中的な ssh アクセスが発生していたことがわかる。

3.2.3 因果関係の解析

グループ単位にまとめられたログテンプレートを基準として、その出現パターンから [12] で提案された手

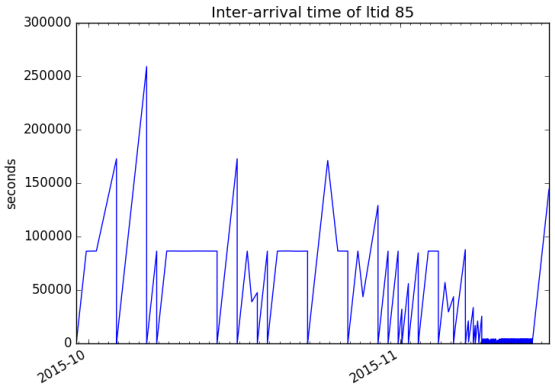

図 4: ログテンプレート番号 85 の出現間隔

法を用いてグループ間の因果関係を解析した。因果関係は図 5 に示すような、非循環部分有向グラフとして出力される。

まず、1 日単位で因果関係を推測した情報をもとに、因果関係を表すエッジの数に注目して解析を行った。図 6 にエッジ数の推移を示す。

図 7 にエッジの出現日数をもとにした CDF を示す。この図より、およそ 80%のエッジは 5 日以内しか出現していないことがわかる。出現頻度の少ないエッジがなんらかの異常状態を示していると仮定し、CDF の 80%を占める 5 日以内のエッジのみを描画したものを図 8 に示す。頻出するエッジを排除したことにより、いくつかの点でピークのように見える部分が出現していることがわかる。例えば、もっともエッジ数の多い 9 月 27 日は、東京大学根津 NOC に設置してある一部のハイパーバイザーが停電の影響で予定外のシャットダウンをしていた時期であることがわかっている。

因果関係の計算単位を 1 日から 2 時間に短縮し、同様に 80%をカバーする希少エッジのみを描画したものを図 9 に示す。こちらでも、9 月 27 日にピークが発生していることがわかる。また、10 月 17 日の夕方に、図 8 では明確には読み取れないピークが現れており、この時間帯には奈良先端大 NOC の停電で、ハイパーバイザーが計画的にがシャットダウンされていたことがわかっている。

その他の部分にもピークが見られるが、現在のところこれらのピークがどんな事象に関連していたかを正確に判断する材料がない。今後は、システムログに加えて、トラッフィックの推移や、トラブルチケットなどの手法を導入して、事実の検証ができる体制を拡充していく必要があると思われる。

3.3 今後の進め方

今年度はシステムログを用いた異常検知技術の研究開発に着手した。システムログは自由形式で記述されたテキストデータであり、その処理にはある種の言語解析的な手法が必要となる。その代わりに、もともとログに含められていた意味、ログを出力するタイミングが、あらかじめプログラマによって意図的に調整されているといった点から、異常事態の意味を取り出すことが可能ではないかと考えている。今後は、より正確なログメッセージの分類手法を確立していく。また、ログメッセージの分類には大きな計算負荷がかかることも分かっており、リアルタイムでの運用を目指した分散処理システムなどへの適用も検討していく。

異常検知には様々な手法が考えられ、今回はログメッセージの出現間隔やログメッセージ間の因果関係を利用した。より正確な異常判定のため、今後も様々な手法を検討していく。

また、検出された異常状態が正当なものなのか、誤検知なのかは、単体のデータでは判断できないため、トラフィックログなど、他のデータ及び異常検知手法との比較などを含めた正確性の向上に努めていく。

4 まとめ

今年度は仮想計算機監視 MIB の標準化と、システムログを用いた異常検知手法の研究開発を実施した。MIB に関しては、今後は実装の拡充及び将来に向けた改定を検討していく。システムログ解析については、基礎技術の確立を継続するとともに、データセットの拡充、高速・リアルタイム処理のための技術開発などを進め、他の異常検知手法との比較などによる提案技術の有用性の検証を続けていく。

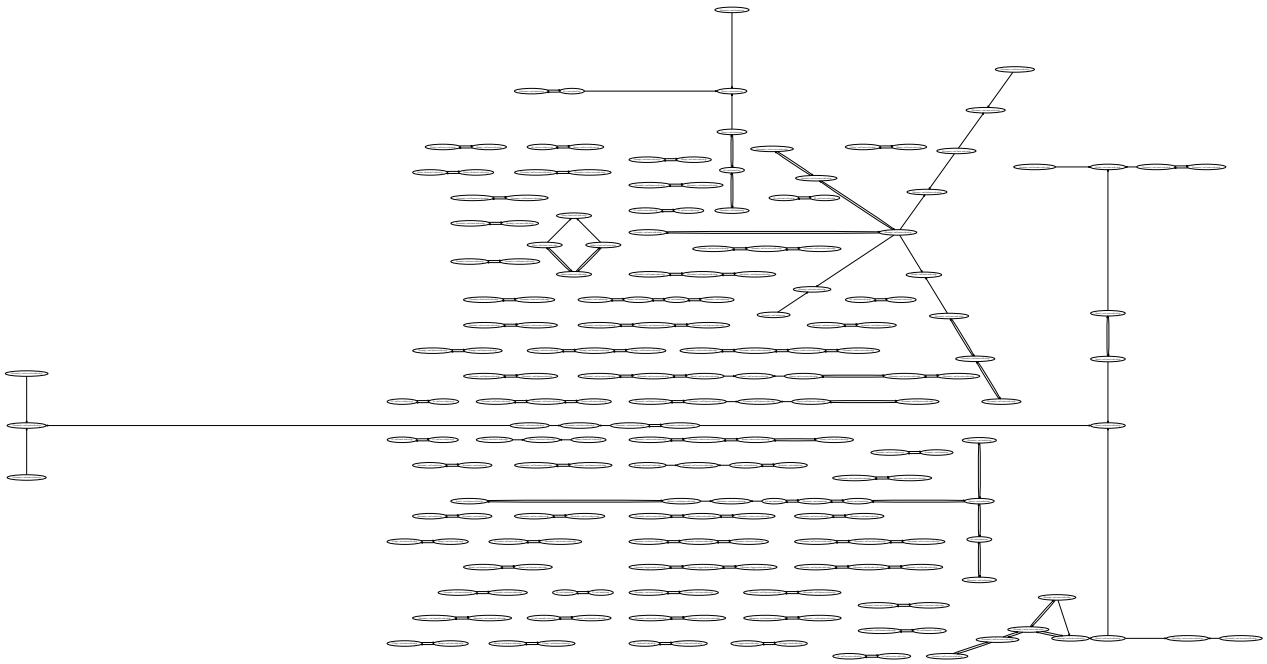

図 5: 2015 年 10 月 1 日の因果グラフ

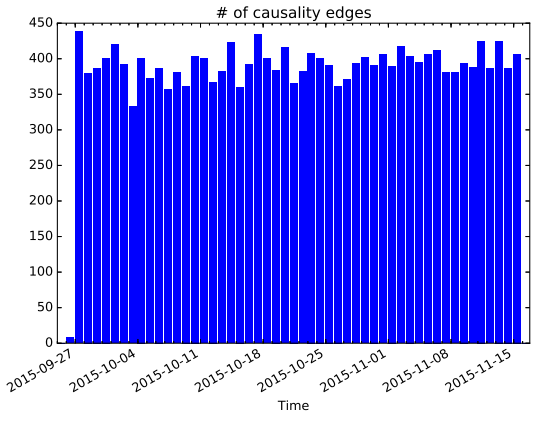

図 6: エッジの出現数推移 (1 日単位)

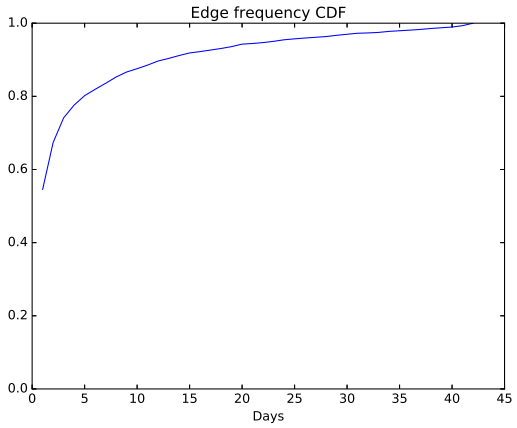

図 7: エッジの出現数の CDF (1 日単位)

参考文献

[1] Fabrice Bellard. QEMU: Open Source Processor Emulator. `http://www.qemu.org/`. (accessed Nov 2015).

[2] R.A. Harper, A.N. Aliguori, and M.D. Day. KVM: Kernel-based Virtual Machine. `http://www.linux-kvm.org/`. (accessed Nov 2015).

[3] Oracle. VirtualBox. `http://virtualbox.org/`. (accessed Nov 2015).

[4] VMware, Inc. VMWare. `http://www.vmware.com/`. (accessed Nov 2015).

[5] Citrix Systems, Inc. `http://www.xen.org/`. (accessed Nov 2015).

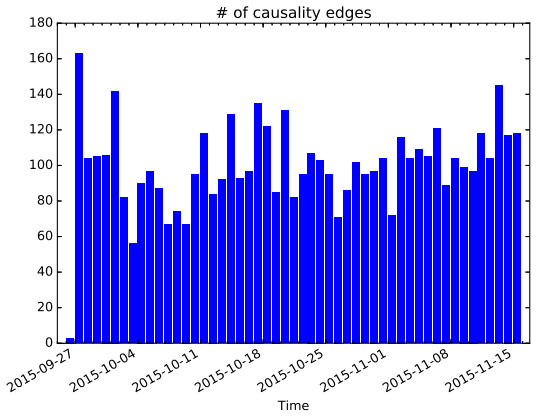

図 8: エッジの出現数推移 (1 日単位、出現数が 5 日以内のもののみ)

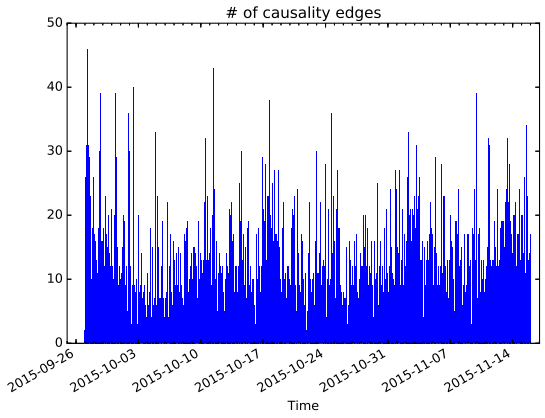

図 9: エッジの出現数推移 (2 時間単位、希少エッジのみ)

[6] Hirochika Asai, Michael MacFaden, Juergen Schoenwaelder, Keiichi Shima, and Tina Tsou. *Management Information Base for Virtual Machines Controlled by a Hypervisor*. IETF, October 2015. RFC7666.

[7] The libvirt developers. The libvirt virtualization API. `http://www.libvirt.org/`. (accessed Nov 2015).

[8] 小林論. システムログ解析に基づく異常検出・原因究明技術に関する研究. Master's thesis, 東京大学, 2015.

[9] Risto Vaarandi. A data clustering algorithm for mining patterns from event logs. In *Proceedings of the 2003 IEEE Workshop on IP Operations and Management (IPOM 2003)*, pp. 119–126, 2003.

[10] John Lafferty, Andrew McCallum, and Fernando C.N. Pereia. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001)*, pp. 282–289, 2001.

[11] Masayoshi Mizutani. Incremental Mining of System Log Format. In *Proceedings of the IEEE International Conference on Services Computing (SCC 2013)*, pp. 595–602, 2013.

[12] Markus Kalisch and Peter Bühlmann. Estimating High-Dimensional Directed Acyclic Graphs with the PC-Algorithm. *The Journal of Machine Learning Research*, Vol. 8, No. 5/1/2007, pp. 613–636, 2007.