Hinemos

ログ管理・ログ監視製品の採用ポイント

機能とライセンス費用における
嵌りやすいポイントと解決方法を一挙ご紹介

課題 1

ITシステムの運用において、発生するログを管理することは非常に重要である。
「ログが重要な障害を示す場合」「ログの件数が異常を表す場合」「ログを蓄積して分析する事で統計的に新たな傾向を把握できる場合」「ログを異常検知や将来予測に用い、その結果を元に運用自動化・効率化に活用する場合」などにおいて、ログの管理は必須となり、その種類とサイズは膨大になる。しかし、ログ管理・ログ監視製品は様々である。ログ管理の専用製品や、監視製品にログ管理が含まれるもの、エンジニアしか扱えない知識が求められる製品や、自動化に伴い製品連携が必須なもの、などがある。

こういった事により、プロジェクトの現場では次の課題が出てくる

課題 1

製品導入の敷居が高い

ログのサイズがどの程度大きくなるか、といったサイジングが困難で、適正なHWや製品のライセンスを選べない

課題 2

監視ツール、ログ管理ツールと似通ったものが導入される

類似の機能を持つ製品が導入され、使い分けが曖昧だったり、両方の製品の使い方を覚える必要があったり等の現場の負担が増える。自動化ツールも別となると製品連携だけで、多大な設計と構築、運用の工数がかかる

課題 3

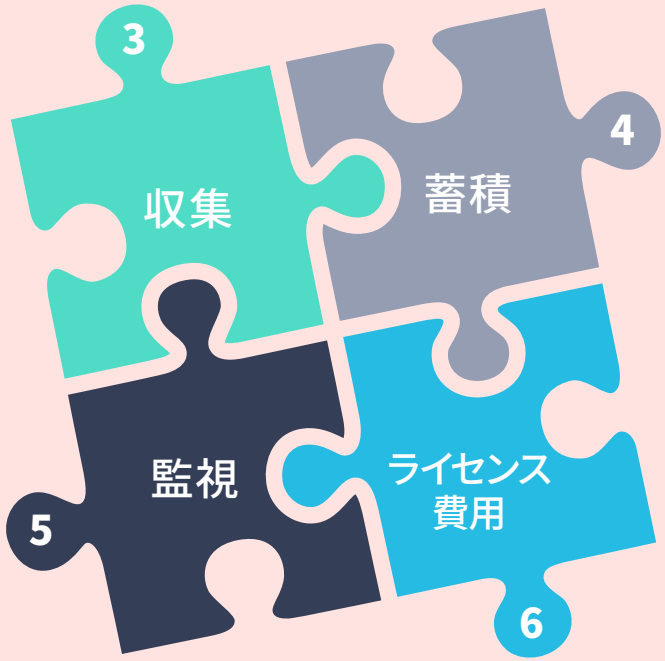
導入しても使われない

海外製品の場合、SEが扱うことを想定したツールの可能性がある。この場合、開発時点では導入は進むが、運用開始後にオペレーターが扱えずに、塩漬けのままになるケースがある

採用ポイント 2

ログは、収集→蓄積→監視の流れで運用自動化・効率化に繋げる。単に収集できる、蓄積できる、監視できるといっても、ログ管理の分野では細かな機能性がユーザビリティにも大きく影響がでるので、もう一歩踏み込んだ実現性の確認を行うべきである。

前述の課題に陥らないためにも、この各機能とライセンス費用について、採用時に確認すべきポイントを示す。そして、Hinemosがこれらのポイントを全てクリアしていることを合わせて紹介する。



収集

3

とにかくログを収集することが先決。
どの様なログに対して、どの様な機能が必要なのか
を紹介する。

POINT 対象ログの種類と作り込み

対象ログの収集を、作り込みなしで簡単に実現できることが重要であるが、カタログスペックだけでは単にログの種類一覧だけが表記されているケースが多い。

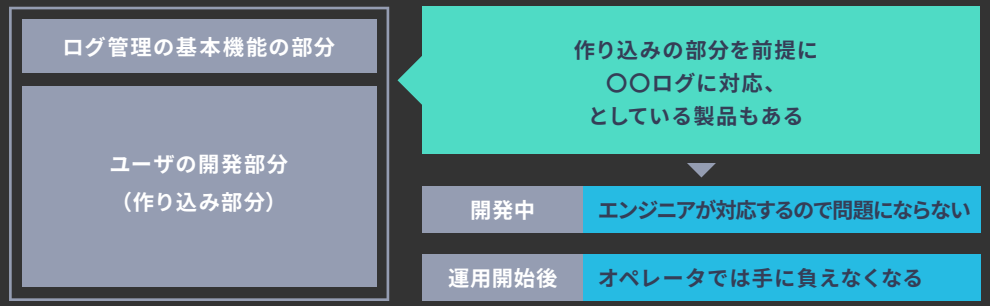
対象ログの種類

対象ログの種類は、一番最初に確認するポイントになる。アプリケーションのログやミドルウェアやサービスのログ、OSのシステムログ（Windowsイベントとsyslog）や、NW機器、ストレージ装置、その他プラットフォーム全般のログは当然対象になる。形式はテキストファイルやバイナリファイル、専用の管理ツールに蓄積されているもの、クラウドのログサービスに蓄積されているものに分類される。広義のログとしては、送信されてくるsnmptrapやsyslogパケットや、パケットキャプチャも対象に含む場合がある。

ログの種類	ログの形式	特殊なログ
アプリケーション ミドルウェア サービス OS NW機器 ストレージ装置 プラットフォーム全般	テキストファイル バイナリファイル 専用の管理ツール ログサービス	snmptrap syslogパケット パケットキャプチャ

作り込み

多くの製品では、紹介したログの多くに"対応"している事をうたっているが、ここで重要となるのは"作り込みの有無"になる。海外の製品やシステムエンジニア(SE)向けの製品の場合、"ユーザの作り込み"としてスクリプト等の開発を伴って"対応"をうたうケースがある。この場合、運用フェーズに入るとSEではなくオペレータ主導の作業となるため、ログの追加や問題発生時の対応に再び開発を伴ったり、対応できず使用を止める"塩漬け"にするといった事に陥る。



Hinemos

Hinemosはオペレータ向けの製品である。収集と監視を統一的なインタフェースで提供し、非常に様々な種類のログもGUI操作だけで収集する。GUI操作も簡単で、対象の種類をクリックして、ログファイルパスやWindowsイベントの対象を指定するといった事だけで事足りる。Webサービスの応答(HTTPレスポンス)等の監視結果もそのままログとして蓄積が可能であり、また収集対象が異なっても基本的な画面構成は同じであるため、非常に簡単に製品を習熟できる。

Hinemosが収集できるログ	
PING監視	SQL監視
プロセス監視	JMX監視
リソース監視	ログファイル監視
サービス・ポート監視	システムログ監視
Windowsイベント監視	相関係数監視
Hinemosエージェント監視	収集値統合監視
HTTP監視	バイナリファイル監視
HTTPシナリオ監視	パケットキャプチャ
SNMP監視	カスタム監視
SNMPTRAP監視	カスタムトラップ監視

Hinemosのログファイル監視の設定

POINT 複数行ログ対応

ファイルに出力されるログは、Linuxのsyslogの様に"最大1024byteの1行ログ"といったシンプルなものだけではない。その代表的なものは、Oracleのログやjavaアプリケーションのログになり、それらは複数行からなるログになる。複数行ログは、より詳細な情報を得られる反面、収集や監視を行う立場としては、どこからどこまでが"1つのログの固まり"かを把握する必要がある。一部の製品では、複数行ログも1行単位でしか収集・監視が出来ない仕様の場合もある。

代表的な1行ログ

Linuxシステムログ

```
Jun 29 17:38:11 Tiger shutdown[30243]: shutting down for system halt
```

代表的な複数行ログ

Javaスタックトレース

```
java.lang.NumberFormatException: For input string: "1.1"
    at java.lang.NumberFormatException.forInputString(Unknown Source)
    at java.lang.Integer.parseInt(Unknown Source)
    at java.lang.Integer.<init>(Unknown Source)
    at ExceptionPrintDemo.formatInt(ExceptionPrintDemo.java:7)
    at ExceptionPrintDemo.main(ExceptionPrintDemo.java:14)
```

Oracleアラートログ

```
Sat Feb 07 12:35:53 2015
 create tablespace TEST_SPACE datafile size 5m autoextend on
Completed: create tablespace TEST_SPACE datafile size 5m autoextend on
Sat Feb 07 12:38:40 2015
 alter database datafile
 '/u01/app/oracle/oradata/CDB122/0/datafile/o1_mf_test_spa_bfc20s53_.dbf'
 resize 8m
```

Hinemos

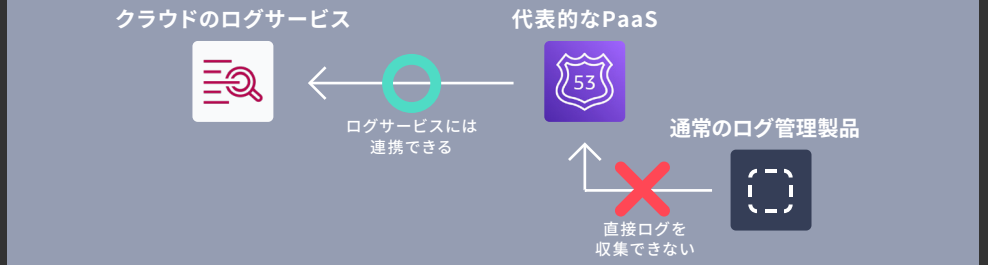
Hinemosでは複数行ログに対応している。ユーザが"先頭パターン"、"終端パターン"、"改行コード"を指定することで様々な製品の複数行からなるログにも簡単に対応できる。
これにより、意味のある単位でログを収集し、監視や分析にそのまま活用する事ができる。

Hinemosの複数行ログ対応

POINT クラウドログサービス対応

クラウドのログサービスとは、Amazon CloudWatch LogsやAzure Monitor Logsを指す。クラウド上のシステムのログ管理には、これらのサービスは必須となる。何故なら、単にEC2の上にアプリケーションを配置してログファイルを収集するのとは異なり、クラウド利用のメリットであるPaaSを活用した場合、PaaSのログは直接的にログ管理製品が収集できないからである。そのため、クラウドのログサービスにログを投入する設定にした後、このログを活用するにはログ管理製品側からクラウドのログサービスに連携する必要がある。

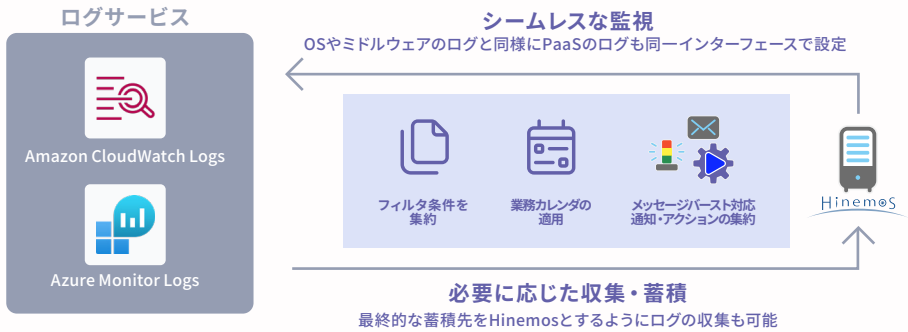
クラウドログサービス



Hinemos

Hinemosでは、Amazon CloudWatch LogsやAzure Monitor Logsのログを収集・監視することが可能である。ログファイルやWindowsイベントを指定するかのように、これらのログを簡易に扱えるため、クラウドにおけるログもHinemosで統括的に扱うことができる。

Hinemosのクラウドログサービス対応



POINT

アプリケーションのログ

多くのプロジェクトでは、アプリケーション開発者以外が運用設計を行っている。具体的には、アプリケーション開発中は開発のみに注力して、運用については別工程や別チームが考える事として分割する。この場合では、既に出来上がってしまったアプリケーションに対して運用を考えるため、本質的な監視を導入しようともしても情報が足りず、アプリケーションの改修を伴わないと実現できないケースが多い。

アプリケーション監視の課題 ❶

アプリ開発後に運用設計
アプリケーション開発中は開発のみに注力して、運用については別工程や別チームが考える事として分割されている

アプリ開発者以外が運用設計を
既に出来上がってしまったアプリに対して運用を考えるため、本質的な監視を導入しようともしても改修を伴い実現できないケースが多い

また、クラウド化・コンテナ化・アジャイル開発といった要素により開発からリリースまでのサイクルが非常に短く、イテレーティブ（反復的）な開発の流れになってきている。つまり、アプリケーション開発から運用開始までの期間が短くなり、基本的な監視を導入しだけで、本質的な監視が行えていないケースが発生する。

アプリケーション監視の課題 ❷

短いシステム開発のサイクル
クラウド化・コンテナ化・アジャイル開発といった要素により開発からリリースまでのサイクルが非常に短くイテレーティブ（反復的）な開発の流れに

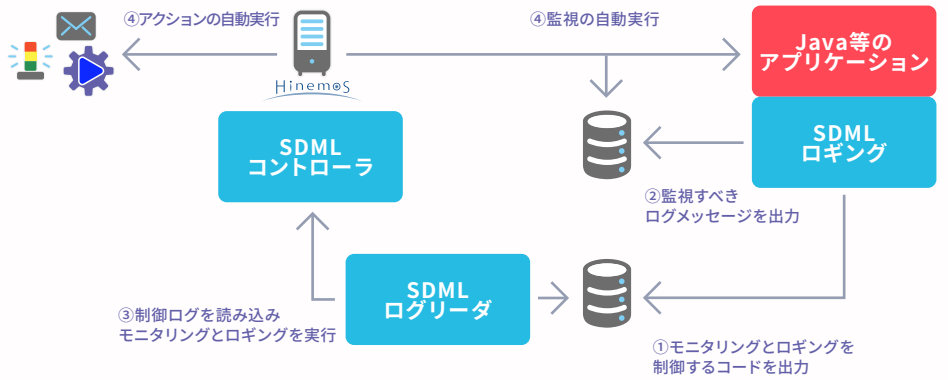
アプリ監視設計の期間がない
アプリ開発や試験から運用開始までに監視設計の期間が短くなり、基本的な監視を導入ししだけで本質的な監視が行えていないケースが多い

こういったケースでは、重要な異常を見落とす可能性がある。それは、運用設計の役割分担や短期開発の優先が、運用の安定稼働を阻害する事を指す。

Hinemos

これに対して、Hinemos は新たなアプリケーション運用の手法として、Software Defined Monitoring and Logging（SDML）機能を提供している。SDML ロギングモジュールをアプリケーションに組み込むことで、制御ログを介しアプリケーションのログと監視を制御する。

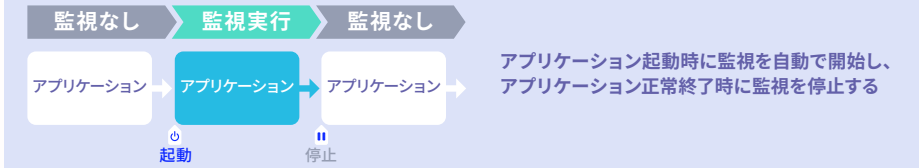
SDML機能



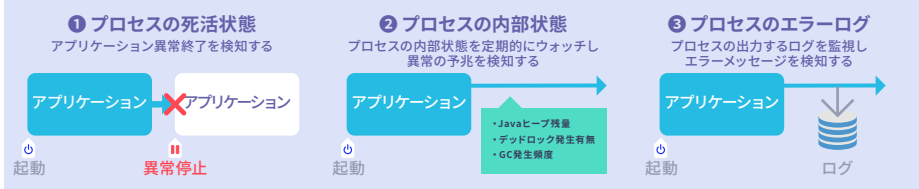
これによって、アプリケーションの起動に伴い必要なログを自動で収集したり、監視を自動実行する事が可能になる。
監視対象も、プロセスのエラーログだけではなく、プロセスの死活状態や内部状態も含まれる。これにより、アプリケーション監視の設計が不要になり、またアプリケーションの改修を伴わずに運用者が把握すべき情報を自動的に監視が可能になる。

自動監視

自動的な監視実行



様々な異常検知



蓄積

ログを収集すると、次に必要なのは蓄積すること。どのような形で、どこに蓄積する必要があるかを紹介する。

POINT 蓄積するログ情報と検索容易性

ログを蓄積する際には、生ログ（加工されていないログ）をそのまま蓄積する事が重要である。しかし、生ログだけの蓄積ではログの検索が非常に難しくなる。蓄積したログを簡単に検索したいケースは多く存在するが、一部の製品では生ログだけを蓄積し、1つの検索窓があるだけのものがある。シンプルな中間一致の場合では問題にならないが、ログの特定の部分を指定しての検索の場合に、非常に高度な検索条件をユーザが検討して指定する必要がある。製品によっては、製品依存の言語でこの検索条件を指定するといった事になる。

難しいログ検索言語

Apacheアクセスログ

127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326

検索条件

生ログのみ

Apacheアクセスログから、HTTPステータスコード=200のログだけを検索するには？

- 前後にスペースを入れればヒットするだろうか？
  “_200_”
- 予期しない行のヒットを除外するため別の条件も入れるか？
  “.*GET.*_200_“

Hinemos

Hinemosではログ収集時に指定箇所にタグ付けをしてメタ情報として合わせて蓄積するログフォーマット機能を備えている。例えば、Apacheのアクセスログの場合、ステータスコードの部分にstatus_codeとタグ付けすることで、検索時に"status_code=200"として簡単に検索する事ができる。各製品のログのログフォーマットは各製品にて仕様が明確に決まっているため、エンジニアが設計しておくことで、オペレータが非常に簡易に検索活用が出来るようになる。

タグ付けと検索利用

検索条件

status_code=200

Apacheアクセスログ

127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326

先頭は送信元IP

[]内は日付情報

規則性からタグ抽出

キー	値
src_ipaddress	127.0.0.1
date	10/Oct/2000:13:55:36 -0700
method	GET
status_code	200
send_byte	2326

生ログ

メタ情報

POINT サイジングとクラウドストレージ活用

ログ管理の一番の最初の難関はサイジングにある。収集したいログの一覧を作成し、ログの出力量を見積り、ログの流量と蓄積に必要なディスク容量を決定する。しかし、収集したいログの出力量は設計時では適切に見積もることができず、また収集したいログは運用フェーズに入った後に増えていくものである。よって、過大なサイジングにより非常に大きなストレージ装置を用意する事になり、ハードウェアコストがかかるが実際の利用率は低かったり、逆にストレージサイズを押さえてしまった事で、集めるべきログを取捨選択して削っていくという無駄な運用コストが発生する可能性がある。ログの圧縮は改善の1要素ではあるが、根本解決にはならない。

ログのサイジングの難しさ

困難なログサイジング → 見切り発車でのHW調達 → 無理のある運用

設計が進んでいない段階で
- ログの種類の確定
- ログ出力量の確定

→ 安全を見て余裕のあるディスク容量のHW購入 → 余裕のある運用だが、HW維持費も高くつく

→ ギリギリ収まるディスク容量のHW購入 → 余裕が無いため蓄積対象を都度選定・削除

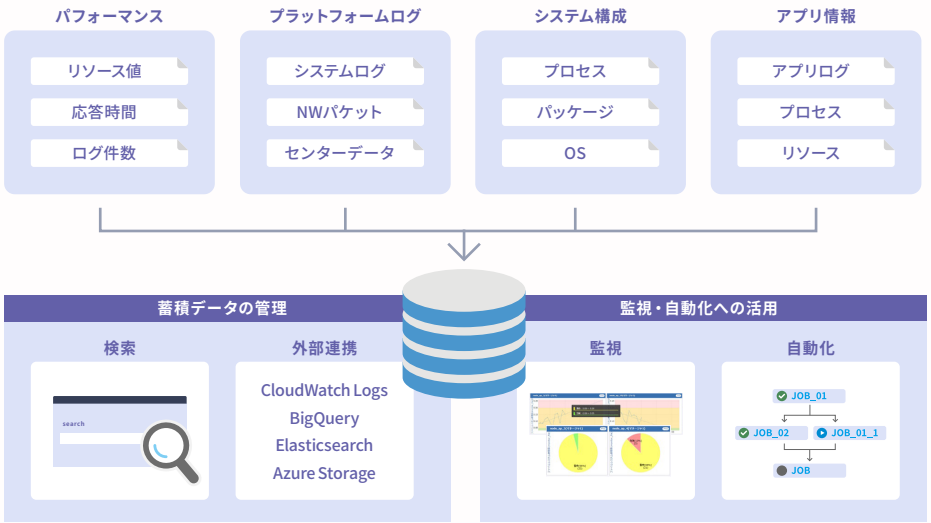
この問題は結局のところ、ログ管理製品だけで解決できない。一般的な解法はクラウドを活用したログの退避にある。クラウドのログサービスやストレージは、物理的なストレージを購入するより安価で利用でき、インバウンドのNW通信には課金されないというメリットがあり、古いログの退避先にすることはクラウド黎明期からの代表的なユースケースである。これにより、スモールスタートでログ収集と蓄積を開始でき、対象の増減にも柔軟に対応できる。これには他にもメリットがあり、クラウド上には機械学習などのサービスもあるため、大量に蓄積したログをそのまま活用する場合にも都合が良い。

クラウドストレージ活用

Hinemos

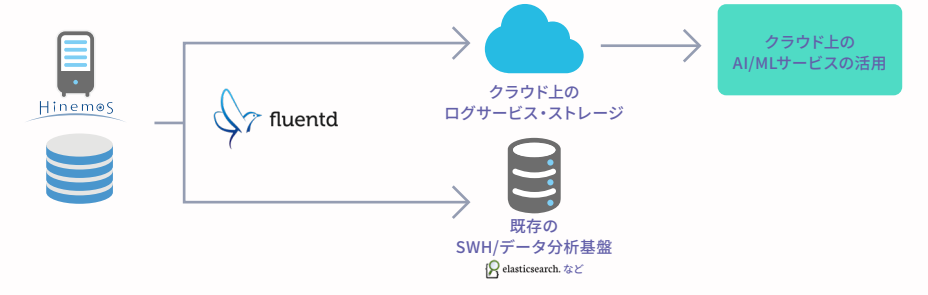
Hinemos では、収集・蓄積したログを fluentd 経由で AWS、Azure、Google Cloud などクラウドのサービスにログを連携する事が可能である。
Hinemos は、これまでの説明の通り、ログ以外にも様々な情報を蓄積し、その蓄積データの管理と監視・自動化への活用を行う。

Hinemosの収集・蓄積



Hinemos の動作するサーバのローカルディスクにユーザの指定する保存期間分を蓄積して、性能グラフ表示やログの検索・分析などが簡易に行えると共に、蓄積した情報をログ転送の機能を使用することでクラウドサービスに簡単に連携する。ログを連携するインタフェースとして一般的な fluentd を採用しているため、クラウドサービス以外にも様々な製品と連携が可能である。

ログ転送と外部サービス活用



この機能を使ったクラウドストレージ活用により、スモールスタートかつ適正・安価なログ運用が可能になる。

監視

5

収集と蓄積をする目的は、運用の自動化・効率化であり、監視はその入口になる。
ログの監視から自動化への連携について、求められる機能を紹介する。

POINT　ログ件数とバースト抑制

ログの監視は、ログメッセージに対するパターンマッチが代表的な手法であるが、それだけでは異常を検知することができない場合もある。例えば、同じ内容のログが連続で出力されるケースや、単位時間当たりのログ件数が急激に増えるケースである。

ディスク異常などでは、I/O　ERRORのログがwriteの単位で発生するため非常に多くの同じログが出力される。また、Apacheのアクセスログにおいて特定IPアドレスからのリクエストが急増して、これが異常を示すような場合もある。

こういった際に、ログ件数による挙動の変化を捉えたいと共に、ログバースト時にユーザへの通知(アラート)を抑制できないと、同一事象を示す通知がユーザに大量に到着し、他の重要なイベントを見逃してしまう二次被害が発生する。

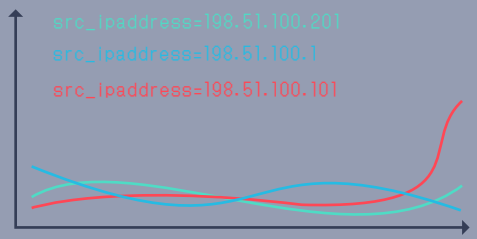
異常時に発生するログ

例）ディスクI/O ERROR

```
INFO service start
ERROR diskio /hoge
ERROR diskio /hoge
ERROR diskio /hoge
ERROR diskio /hoge
ERROR diskio /hoge
```

2件目以降はほぼ同じ内容

例）送信元IPによるログ件数

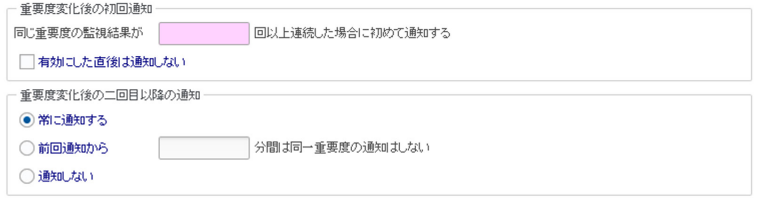


Hinemos

Hinemosでは、ログの件数による監視と、バースト時の通知抑制の機能を持つ。ログ件数監視により、指定の条件を満たすログが単位時間に一定件数出力されるとアラートを上げることができる。指定条件には、ログフォーマットによるタグを利用できるため、例えば送信元IPアドレス(src_ipaddress)を限定した件数のカウントが可能である。これにより、普段と異なるログ出力状態を直ちに検知ことができる。また、バースト発生時に、初回通知や二回目以降のアラートの発砲を制限する事ができる。これにより、全て同一事象を示す通知により運用監視端末やメールが埋め尽くされことを防ぐことが出来る。

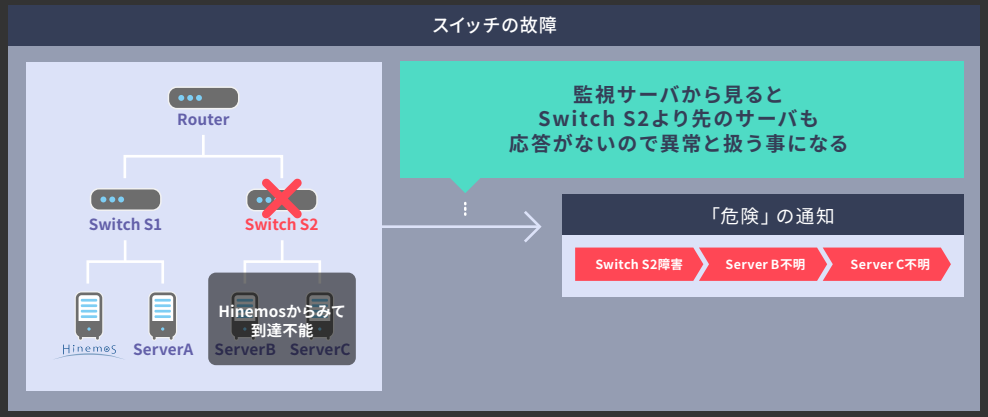
ログ件数監視

バースト抑制

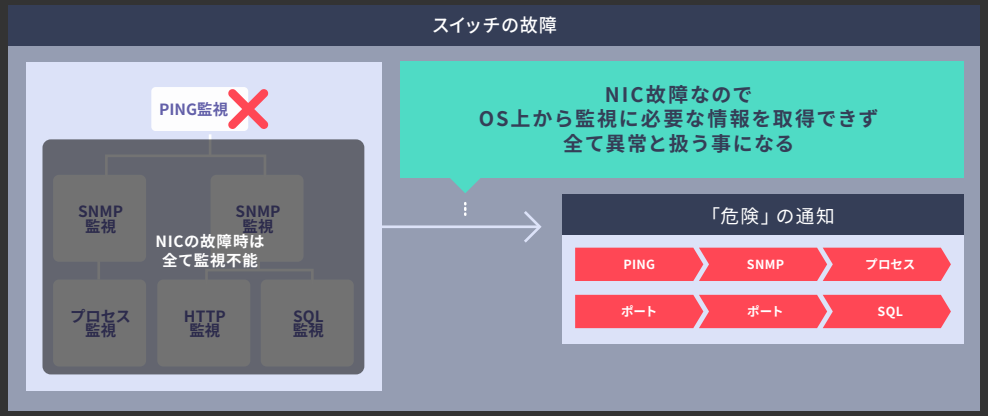


POINT メッセージフィルタ

実際にはログだけではなく、様々な監視結果と合わせて異常を判断すべきである。例えば、NWセグメントを中継するスイッチに障害が発生すると、このスイッチ経由でアクセスするアプリケーションやサービスが全て通信エラーでログを大量に出力する。そのため、スイッチの障害に影響を受ける様々なアプリケーションやサービスの異常だけで運用端末の情報が埋め尽くされてしまう。



例えば、あるサーバに対してPING監視からプロセス監視までの多種多様な監視を行っていた際にNICに障害が発生すると、PINGの応答だけでなく、サーバのOSから何も情報を取得できないため、行っていた全ての監視でエラーが発生し、同様に運用端末の情報が埋め尽くされてしまう。



この様な形で、真の障害に付随するエラーで運用端末の情報が埋め尽くされると、同時に発生していた重要なログメッセージを見逃す可能性もある。そのためには本質的なイベントに注力して監視できる事が重要になる。

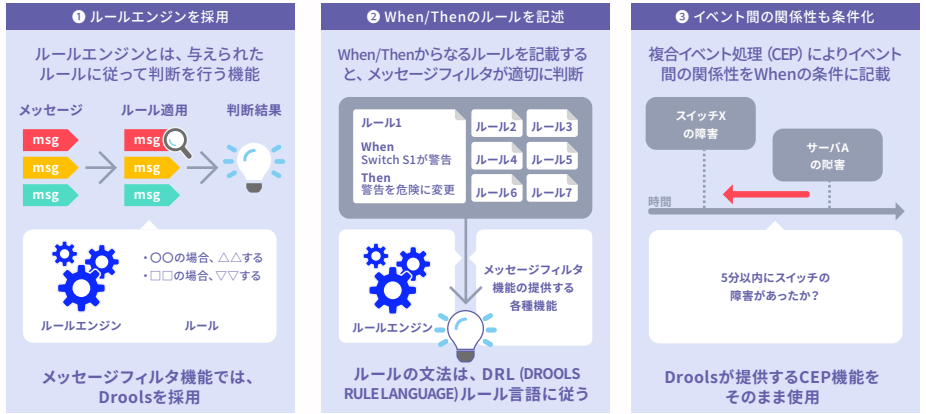
Hinemos

本質的なイベントの特定には複数のログや監視結果との関係性から、判断する必要が出てくる。この関係性は、対象システムやアプリケーションの構成により様々であり、そのために柔軟に関係性を定義し、イベントを特定（フィルタリング）する事が求められる。
Hinemosでは、メッセージフィルタ機能により、本質的なイベントを抽出が可能になる。

メッセージフィルタによる本質的なイベントの特定

このメッセージフィルタの特徴は大きく３つある。１つ目の特徴は、ルールエンジンを採用、している事である。ルールエンジンとは、与えられたルールに従って判断を行う機能であり、ルールの定義により様々なフィルタリングが可能になる。２つ目の特徴は、When/Thenのルールを記述、する事である。ルールを記述するとは難しそうに聞こえるが、When（～の時）/Then（～する）というシンプルな形になる。ルールの数は複数あっても構わなく、ルールを記載するとメッセージフィルタが適切に判断する。３つ目の特徴は、イベント間の関係性も条件化、できる事である。これは複合イベント処理（CEP）と呼ばれるもので、あるサーバからの応答がないというイベントが発生した際に、スイッチも障害のイベントが発生していたら、というイベント間の関係性をルールの条件（When）に記載できる事を指す。
このメッセージフィルタ機能を使用する事で、本質的ではない不要なメッセージを排除し、運用者が効率的に目的の運用業務を果たしたり、ユーザの判断を介在せず運用自動化に繋げる事が可能になる。

メッセージフィルタの特徴

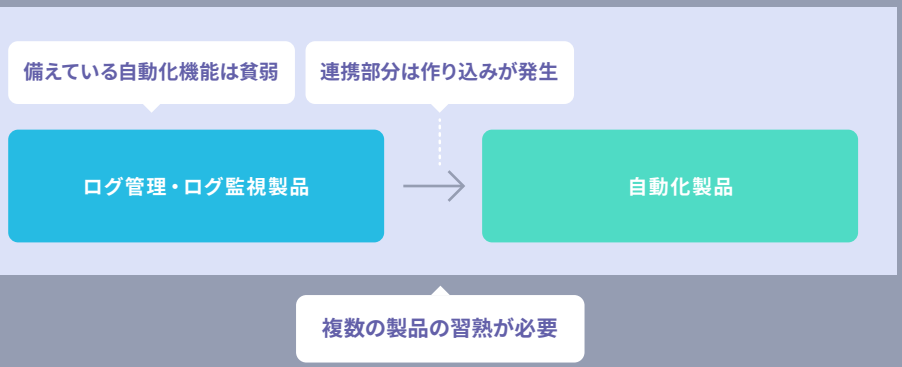


POINT

自動化連携

運用自動化・効率化には、ジョブ管理やワークフロー管理の機能が必要になる。ログ管理・ログ監視の製品に、異常を検知した際に簡易なコマンドを実行する機能は備えているものは多いが、これでは不十分である。例えば障害復旧を行うような処理を自動化した際には、その処理が正常に稼働したかの判定や複雑な条件分岐、サーバを跨った処理の制御が必要になる。また、専用のジョブ管理・ワークフロー管理の製品を導入した場合、ログ管理・ログ監視の製品との連携を作り込み、そして２種類の製品を使いこなす運用コストがかかる。

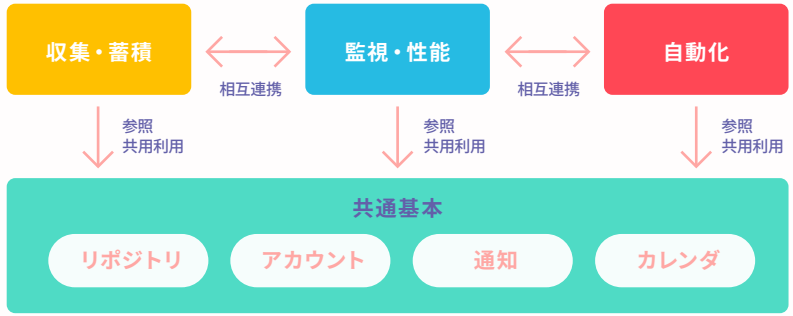
自動化連携の課題



Hinemos

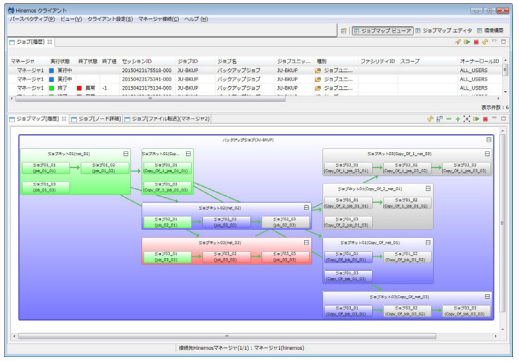
Hinemosでは、大きな機能分類として「収集・蓄積」「監視・性能」「自動化」の３大機能と、それらを支える「共通基本」機能をワンパッケージで提供している。もちろん、これらの機能間は製品として相互に連携している。

Hinemosがワンパッケージで提供する機能



そのため、Hinemosはログの収集や監視から直ちに自動化機能、つまり、ジョブ管理・ワークフロー管理の機能に連携する事が出来る。連携方法も簡単であり、監視結果が「危険」の場合に指定の障害復旧ジョブを起動する事を、GUIから選択するだけで実現できる。もちろん、このジョブは複数のサーバに跨る複雑なものであってもよく、途中にユーザが対話的に判断を入れる事も可能、そして、起動したジョブの状態遷移もビジュアライズされている。

Hinemosのジョブ管理



よって、Hinemosではワンパッケージで提供されているため、自動化連携を行う際にセットアップが不要であり、統一的なインタフェースで操作ができ、そして何が製品の問題が発生した際も製品間の切り分けが不要なワンストップ保守というメリットがある。

ライセンス費用

どんなに高機能な製品であってもプロジェクトとして
採算が合わないと意味がない
そもそも費用は見積れるものなのだろうか

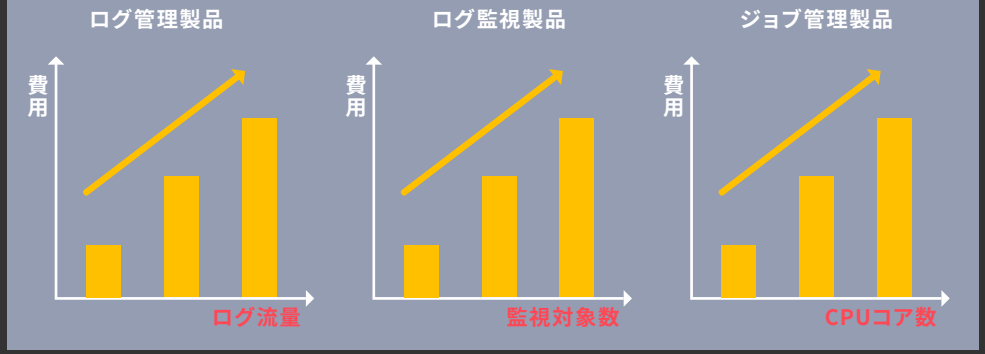
POINT 費用のスケールファクタ

ログ管理製品は、基本的にログ流量でライセンス費用がスケールする。サイジングが難しいにも関わらず、製品選定と購入時のフェーズで既にログ流量の見積が必要となってしまう。また、開発や運用フェーズで扱いたいログが増えた場合に、追加の費用を支払うかログ流用を上限に調整する必要が出てくる。

ログ監視製品は、基本的に監視対象台数でライセンス費用がスケールする。仮想化やクラウド利用が当たり前の時代に、サーバーのスケールアウトがログ監視製品のライセンス費用により制限される。

自動化を担うジョブ管理製品は、基本的にはサーバ数やCPUコア数でライセンス費用がスケールする。サーバ数が少なくても高スペックのサーバを使用すると高価になり、またリソースのスケールアップがライセンス費用に制限される。

費用のスケールファクタ



Hinemos

Hinemosは基本的に、ログ流量や監視対象台数、コア数によりライセンス費用はスケールせず、あくまでマネージャ機能のインストール単位で費用が決まる。

Hinemosのサブスクリプション費用

これにより、ポイントで説明した各観点において、非常にメリットがある。

費用のスケールファクタが少ないメリット

	Hinemosのメリット
製品選定時	拠点単位にHinemosマネージャを配置する台数、等の簡易な見積もりで済む
設計・構築時	費用変動がないので運用対象の増減、環境変更が簡易に行える
ログ管理機能の利用時	ログ流量の厳密な見積もりは不要
ログ監視機能の利用時	管理対象に導入するエージェント数を削減する必要が無い
自動化機能の利用時	自動化製品の追加不要、ハイパフォーマンスなサーバを導入しても安価に済む

クラウドの様な柔軟に環境変化するシステムにおいてでも、ログ管理・ログ監視機能を適切な費用で提供する製品、それがHinemosである。

まとめ

7

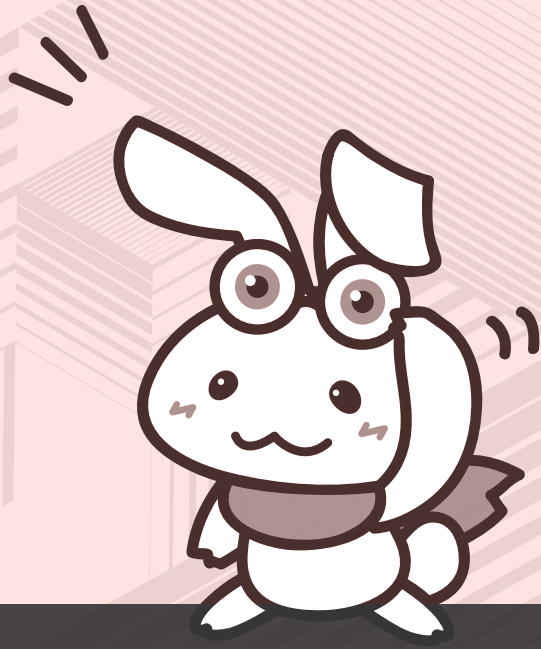
ログ管理・ログ監視の選定ポイントを、収集→蓄積→監視の機能流れで紹介した。
その中で、採用時に確認すべきポイントを示し、Hinemos がそれら全てクリアし、
安価で効率的なログ管理・ログ監視を実現することを紹介した。

Linux、Windows、各種 UNIX 環境から、仮想化・クラウド環境に幅広く対応する Hinemos を
ログ管理・ログ監視における重要なソリューションとして是非活用頂きたい。

お問い合わせ

→ **Hinemosポータルサイト**
https://www.hinemos.info

→ **問い合わせフォーム**
https://www.hinemos.info/estimate



NTTデータ 先端技術株式会社

2021.12 版