

ヘルスチェック項目の標準化と自発積極型転換による システム資料自動診断

千葉 雄太 尾崎 恭一 森山 香織 井上 雅貴

Automatic Diagnosis by Standardization of Health-check Items and Self-starting Method

Yuta Chiba, Kyoichi Ozaki, Kaori Moriyama and Masaki Inoue

システムの障害対応や予防保守サービスを提供する時、事前に関連するシステム情報を収集し、これらの情報を元に各種調査をする。これらを手作業で行うと、情報が膨大であるために大きなワークロードがかかる。そのために従来より自動診断の手法が開発されてきたが、その診断項目数が不十分であったりするなど、その適用が限定されていた。本論文では、診断項目の追加を容易にするためにヘルスチェック項目の記述の標準化と知的資産化を図り、適用拡大のために利用者の要求によって診断処理を開始する待ち受け型から、サービス・センターに送付された資料を自律的に自動診断し、必要な情報を担当者に伝える自発積極型への転換を提案する。これにより適用件数と重要問題の検知の大幅な拡大とコストの削減を実現した。

A vast amount of system information needs to be collected and analyzed for system fault response and preventive maintenance services. Handling this task manually imposes a high staff workload. Therefore, an automatic checkup system was developed but not widely used, because its health-check items and capabilities were limited. This paper discusses the standardization of health-check items and the challenge of switching the automatic analysis method from being passive to being more aggressive in order to increase coverage. As a result, there has been a great increase in the number of cases to which automatic analysis has been applied leading to the detecting of critical problems, allowing cost savings to be achieved.

Key Words & Phrases : オートノミック・コンピューティング, ACPD, 自動診断, 障害未然防止, 安定化
Autonomic computing, ACPD, Automated checkup, Problem prevention, Stabilization

1. はじめに

基幹業務を扱うサーバー・システムに重大な障害が発生すると、そのお客様のビジネスへの影響が非常に大きくなる。特に、金融機関、医療分野、マスコミなどの公共性の高い分野では、お客様のビジネスのみならず社会的なインパクトが甚大となる。システムの保守サービス提供者は、お客様にご満足いただけるようにするだけでなく、大きな社会的責任をも意識し、少しでも早く、正確な保守サービスを提供しなければならない。

そのためにサービス提供者は、日頃から自身の知識の習得やスキルアップを図り、サポートチーム内やバックエンドのサポート組織との各種情報の共有をすることが不可欠だが、課題も少なくない。クラウドに代表されるさまざまな仮想化技術が進む中で、サポート対象の製品とシステ

ム環境は日々複雑さを増している。それに伴い障害が発生した際に調査するべきポイントは増加を続けており、各種ログ情報や稼働状況の指標の種類も多岐にわたる。

単一の製品である AIX を例に挙げても、重大な障害とみなされる HIPER [1]、PE (PTF in error: 不具合を含む修正モジュール) の APAR (Authorized Problem Analysis Reports) が 2009 年に限っても新規に 150 件作成される [2] など、障害情報も日々更新されている。これらの状況から、障害対応を実施するサポート・メンバーの負荷は増大しており、障害の発生したシステムの網羅的な調査は困難になっている。

一方、障害が発生してからの対応ではなく、未然防止のニーズが近年ますます拡大しており、単なる障害対応だけではお客様に十分にご満足を提供できないという課題がある。筆者らの所属するシステム技術支援サービス (STSS) オープンシステムズでは、これらの課題に数年前から取り組み、障害対応に関してはオートノミック・

提出日:2010年9月6日 再提出日:2011年6月14日

コンピューティング [3] を利用した問題判別（以下、ACPD: Autonomic Computing Problem Determination）技術を活用し、システム・ダンプの自動解析を進めてきた。問題発生 of 未然防止に関しては、システム・ヘルスチェック・サービスなどの予防保守関連サービスの提供を充実化させてきた。そのうち、System p のクラスタリング製品である PowerHA に限っては ACPD 技術を応用したチェック自動化により効率的な自動診断を実践してきた。

これらの取り組みは一定の効果を上げてはいるが、長期にわたって利用されていくにつれ、いくつかの課題があることが分かってきた。本論文では、2 章で従来の ACPD の課題を明確にし、3 章でそれらの課題についての解決のアプローチを紹介、4 章と 5 章で実装についての工夫などを説明し、6 章で業務への適用とその効果を確認する。

2. ACPDの課題

本章では従来手法である ACPD の課題を提示する。

2.1 待ち受け型の診断トリガー

従来のフレームワークは、図1のように担当者が自動診断のページにアクセスし、診断開始の手順を実施するという待ち受け型であり、診断のトリガーは担当者の診断開始操作であった。そのため担当者が自動診断の内容を知っており、かつ利用する意図がない限りは積極的に利用されなかった。また診断開始のためには、例えば AIX の例では標準の情報収集コマンドである snap など で収集された多数のファイル、ダンプ、トレースなどの資料のアップロードも必要であり、製品によっては資料のサイズが数百 MB ～数 GB に達することも多く、付随する手間が多いため積極的に利用されない要因となった。PowerHA に限った件数であるが、障害調査とヘルスチェック・サービスでの資料受領数はおよそ年間の 650 件に対し、2009 年での自動診断の適用は、ヘルスチェック・サービスで利用された 22 件以外にはほとんど見られ

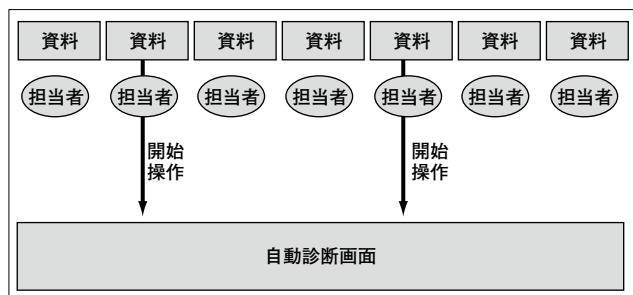


図1. 待ち受け型アプローチ

ず利用率が低かった。

国外も含めさまざまなサポート組織で作成されている分析ツールには、このような待ち受け型となっているツールが多数存在しており、そのツールが有用なものでも活用される機会が制限されてしまっている。

2.2 ヘルスチェック項目追加の際の負担

システム・ヘルスチェック・サービスにとって、チェック項目の蓄積はサービス品質の要である。ACPD は、シンプトン (symptom (症状))*1 [4] と呼ばれる問題を診断するための判別ルールに、実際のデータが合致したことを判定し自動解析を実現している。例えば、ダンプの診断では既知の不特定のスタック・トレースがシンプトンとして登録されており、そのシンプトンに実際のダンプから生成されたスタック・トレースが合致するかを判定している。

この判別ルールの定義手順は汎用化されており、比較的容易に新しいシンプトン（判別ルール）を登録することができる。特にシステム・ダンプの自動解析に関しては、調査対象をダンプのスタック・トレースに絞っていることもあり、他の問題領域への柔軟な対応が難しい代わりに、新たなシンプトンの追加はスタック・トレースを登録することで実現できる。

PowerHA の診断では、クラスター製品という性質上、複数ノードから収集された履歴情報、稼動状況に関する情報、設定情報など多種の情報の調査が必要となる。さらに、多岐にわたるログのフォーマットと複雑な診断条件に柔軟に対応する必要がある。そのため図2に示すように、各種ログをプリプロセスのシェル・スクリプトで Common Base Event（以下 CBE） [5] フォーマットに変換する。そして、その変換したデータとシンプトンとが合致するかを判定するようにしている。

しかしながら、実際に PowerHA ではイベント・スクリ

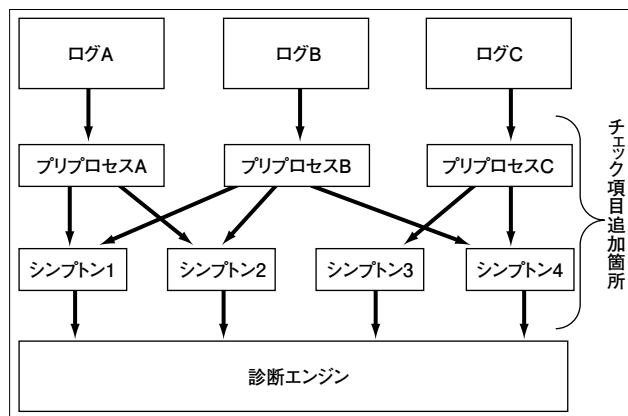


図2. PowerHAにおけるACPDの処理概要

*1 問題の判別ルールを記述したナレッジ

プトの多量のデバッグ情報が出力されるようなログなどを CBE フォーマットに変換する意味がほとんどないことも多かった。また条件判別では、単なる文字列や数値比較だけでは対応できず、例えば複数ノードの物理コンポーネントのリソース情報を比較するために while 文などの繰り返し制御構文で記述したり、TCP/IP サブネットの計算のためにビット演算をする必要があったりする。

このような比較的複雑な情報処理を必要とするチェック項目では、プリプロセスのスクリプト処理の段階で問題検知までを済ませ、CBE ではその結果のみを記載するなど、汎用化の利用が形骸化されてしまい利点が生かしくいという面があった。またプリプロセスとシンプトンの関係を意識してチェック・スクリプトを記述する必要があるため、その作成やテストが煩雑になってしまう面もあった。これらのスクリプト作成はそれぞれ独立した作業となるため、プリプロセスとシンプトンに分けた従来の手法が逆にチェック項目の個別の開発の足かせにもなっていた。

チェック・スクリプトを作成するのは通常は障害対応の業務に携わる技術者であり、問題解析の分野では深い経験があり製品知識も豊富ではあるものの、予防の観点から製品を見る機会は少なく、予防保守のためのチェック項目を挙げるには異なる着眼点も必要であり、必ずしも適切なチェック項目が容易に挙がるとは限らなかった。また、チェック・スクリプトの品質の確保のためにはテストも必要であり、実装手間は決して少なくないため、結果としてメリットが少なく積極的に項目の追加がされないという課題があった。

2.3 限定された ACPD の適用範囲

従来の ACPD 技術は、基本的には製品タイプには依存しないデザインであったが、シンプトンの蓄積方法や Web 画面などのユーザー・インターフェースは固有の製品を対象に設計されているために他製品への展開は必ずしも容易ではなかった。システム・ダンプ解析に関してはシステムクラッシュの問題に焦点を当てていたが、件数としては AIX 関連の問題のうち 60 件に 1 件程度の割合であり、対象範囲は狭かった。また、システム・ダンプ解析はクラッシュを引き起こしたスレッドのスタックのパターンに注目して判別を行う仕組みであり、別の問題への展開は困難だった。

2.4 予防保守のためのチェック関連情報の不足

従来の ACPD 技術では、システム・ダンプや各種ログファイルから変換された CBE を、登録されたおのおののシンプトンに照らし合わせ、条件が合致し検知された問題を列挙するという診断手法であった。

障害対応の場合にこれで十分な場合が多いが、目的が予防保守の場合には、提供された資料に対してどのようなチェックが行われ、問題点だけではなく、どのような点は問題ではなかったのかを知りたいというニーズがあった。また、資料の不足などで検査ができなかった場合にも、やはり同様に不足情報の記録が必要であったが、ACPD 技術では、ファイルが存在しないということはすなわち合致するべき対象のイベントがないということと同義であり、診断は実施されなかった。このように、ACPD の枠組みだけでは検査が不可能な場合の警告を出力することは困難であった。

3. 課題解決のための4つのアプローチ

前章の課題を解決するための4つのアプローチを以下に示す。

3.1 待ち受け型から自発積極型への転換

従来の待ち受け型のフレームワークに対し、図 3 に示すように、障害調査などの目的で送付されるすべての資料（ログ情報、構成情報、システム稼働状況に関する情報など）に対し診断を実行し、結果を案件担当者に自動的に通知するという自発積極型アプローチを検討し実装した。システム名は IBM 幕張事業所の所在地の千葉市美浜区にちなみ美浜式 ACPD とした。

美浜式 ACPD の処理フローの概略は以下のようになる。

1. 定期的に資料アップロード先サーバーをチェックし、新着資料の有無を確認する。
2. 新着の資料が見つかった場合には登録されているチェック・スクリプトを順次実行し、問題の有無を判別する。
3. ファイル名などから問題管理レコード（以下、PMR: Problem management record）番号を判別する。
4. PMR 管理のためのデータベースから PMR のオーナー情報を取得し、診断結果をメールで通知する。

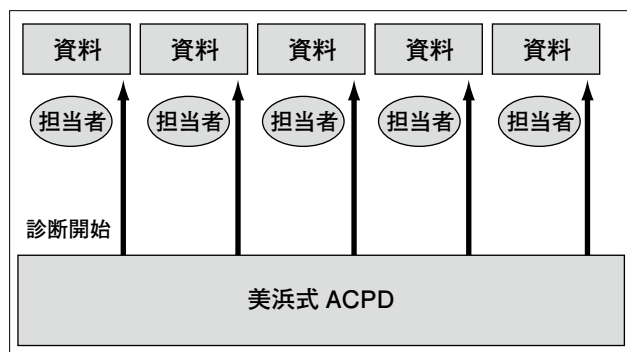


図3. 自発積極型アプローチ

上記の実現により、お客様から資料が送付されてからは手作業なしで診断結果がメールで担当者に通知され、この診断結果を参考に担当者は障害を調査することができる。また、対応中の障害とは直接関係がなくても、潜在する重要な問題が検知されている場合にはお客様にとって有用な情報となりえるので、お客様に提供することも可能である。重大な問題が検知された場合は、予防保守の必要性の訴求に有効な根拠となる。

3.2 チェック・スクリプトのアセット化

チェック項目と自動化チェック・スクリプトの提供者へのメリットが少ないという課題に対して、提供者のメリットにつながりやすくなるようチェック項目のアセット化をサポートする工夫をした。具体的には、チェック・スクリプト記述の仕様を公開し、標準化仕様として IBM の社内アセット管理システムに登録した。このチェック・スクリプトの標準化仕様についての詳細は 4 章で述べる。この仕様に個々のチェック・スクリプトからのアセット管理システム上の関連付けの情報をもたせることにより、即座に美浜式 ACPD の自動チェックとして登録し業務の効率化ができることが分かるため、有用性のアピールとなり、アセット登録の際のバックアップとなる。チェック・スクリプトのアセット登録を多くした技術者にインセンティブを与えることができればさらに登録数は拡大できると考えられる。

また、多数の技術者が独立して実装できるようにするために、図4に示すようにチェック・スクリプト実装者の立場からの平易さを重視し単一のチェック・スクリプトでチェック処理が完結可能なデザインとした。

このチェック・スクリプトの仕様は、汎用的な使用を視野に入れており、同梱の診断開始スクリプトを使用すればローカル環境で診断が行える。従来のような Web ベースの自動診断をお客様に対して提供するなどの新しい仕組みを検討する場合にも、一度記述したチェック・スクリプトの再利用が可能である。

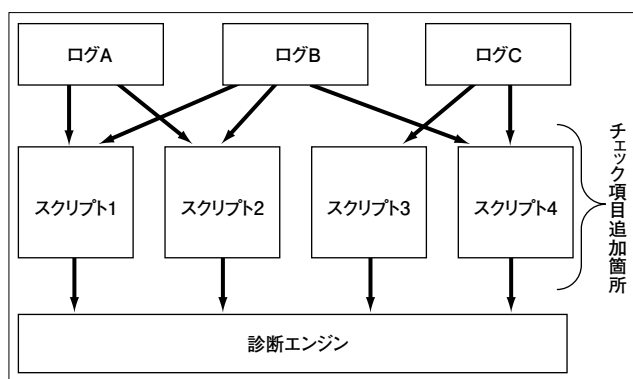


図4. チェック概略図

3.3 統一したインターフェースの策定

美浜式 ACPD を複数の製品に対応できるように以下の工夫をした。

- 1) 製品ごとに異なるユーザー・インターフェース画面が提供されていたが、単一の形式ですべての製品の診断結果が記載されたメールを送信するようにした。
- 2) 特定の製品情報だけがが必要な場合や、予防保守には不要な情報がある場合など使われる条件に対応できるように、製品、診断目的、重要度でフィルタリングした診断結果を送付できるようにした
- 3) スクリプトの仕様として日本語、英語両方のメッセージが必須であり、未実装ではあるが送信対象によって英語のメッセージを送信することができるように海外展開のことを配慮する。

3.4 予防保守のためのヘルスチェック関連情報の記録

予防保守サービスの場合、問題の検知以外に、どのチェックは問題なかったかなどの情報が出力されなかった課題に対して、チェック項目を軸とした診断で、「問題を検出」、「問題なし」、「情報不足でチェック不可能」の分類を明確にした。

診断結果は、診断結果 CSV ファイルに一旦保存し、ヘルスチェックや障害調査などの用途によって診断結果 CSV ファイルからヘルスチェック結果レポートや障害調査担当者へのメールを生成できるようにした。

障害調査担当者へのメールでは重要な情報を効果的にアピールするため、検知された問題の概要のみを、図5のような診断結果として送付することにした。そのほかの情報は保管し、リンクをメールに追加した。

4. チェック・スクリプトの標準仕様

さまざまな背景を持つ多数技術者がチェック・スクリプト作成に参加できるように、その仕様は平易に、決まりごとを極力排するデザインとした。仕様書は印刷しても気軽に参照できるよう、サンプルも含めて 4 ページ以内に収まるよう工夫した。

仕様のポイントとなる部分は下記である。

- ・ 展開された調査対象ファイルの場所が第一引数に指定されること。
- ・ 出力として、下記の情報をホスト名と調査日付を名前に持つ図6のようなコロン区切りの診断結果 CSV ファイルに記述すること。なお、ksh の共通関数としてファイル書き出し用の関数が用意されており、詳細な書式をチェック・スクリプトで指定する必要は無い。

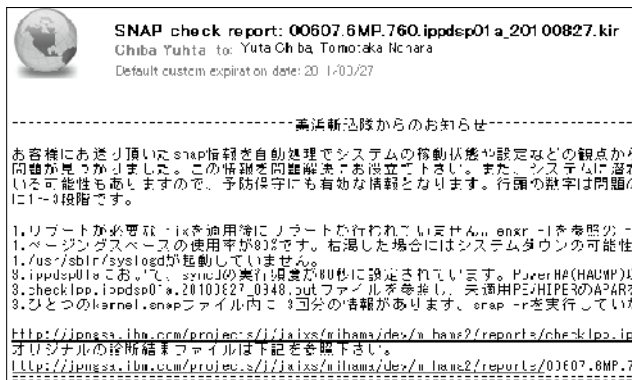


図5. 問題通知メール例

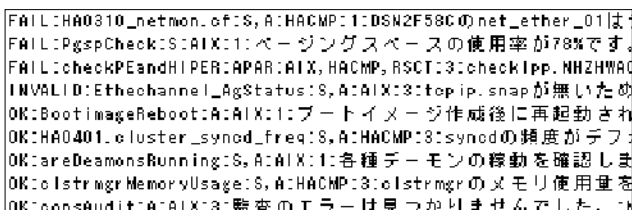


図6. 診断結果CSVファイル例

- ▶ 診断結果キーワード（問題検知、問題非検知、資料不足などにより調査不可能の区別）
- ▶ チェック項目識別名
- ▶ 調査目的（予防保守、障害調査）
- ▶ 製品種別キーワード（AIX,WAS,DB2 など、APAR 適用状況調査の区別）
- ▶ 重要度（1～3まで3段階、1がサービス停止を伴いえる最重要項目）
- ▶ 概要メッセージ（日本語）
- ▶ 概要メッセージ（英語）
- ▶ 追加詳細情報のファイル名（必要な場合のみ）
- ・ ksh スクリプトで作成すること。別言語での実装が必要な場合には一度 ksh スクリプトを介す必要がある。
- ・ ヘッダー部分にコメントで特定のキーワードとチェック項目情報、作成者情報、対象資料種別を記述すること。

5. 美浜式 ACPD の運用上の工夫

美浜式 ACPD を運用する上で工夫した点について述べる。

5.1 資料サーバー上での自動診断

お客様から資料をお送りいただくサーバーは海外にあり、通常では資料をローカルの環境に転送しなおして調査をする。しかしながら回線の帯域は狭く資料の転送には時間がかかり、自動診断ですべての資料を転送した場合には解析が間に合わないことが考えられた。この課

題の克服のため、各チェック・スクリプトは、通常はローカルの開発環境に配置し、自動診断の実行の際には資料サーバーに各チェック・スクリプトを転送の上、資料サーバー上で解析を実行し、診断結果のみをローカルに転送する方法を取った。3ヶ月間の案件の実績から一日当たりの資料の合計サイズは数 GB に達するが、必要なファイルのみの転送であれば転送量は数 MB に抑えることができ、効率的な診断が可能であることが分かった。

5.2 重要 APAR 適用状況の判別

IBM 製品の不具合はそれぞれ APAR（Authorized Problem Analysis Reports）という文書が作成され、ユニークな番号が採番される。その中でもデータ・コラプションを引き起こすような重要な不具合は HIGH IMPACT/PERVASIVE APAR（以下、HIPER）と呼ばれる。また、修正モジュールそのものに新たな不具合が含まれる場合は PTF IN ERROR（以下、PE）と呼ばれ、この HIPER と PE のいずれかに該当するものは重要 APAR とみなされ、ヘルスチェック実施の際の重要な確認ポイントとなる。特定の資料に対してこれらの APAR の適用状態を確認するためには PowerHA 自動診断を利用するしかなかったが、PowerHA 自動診断は APAR 番号の適用状態を判別する instfix コマンドの結果を元に判別を行っているため、厳密性の面では不完全であった。例えば、あるファイルのバージョンが 5.3.9.20 から 5.3.9.30 の間のみが存在する問題があるとすると、当該ファイルのバージョンが 5.3.9.10 のシステムでは実際にはその問題は発生しないが、instfix コマンド結果としてはその問題の APAR 番号は出力されないため、診断結果として修正は未適用と判断されてしまう。この場合、APAR の修正が未適用なのは事実であるが、本来問題が発生しないファイル・レベルであるにもかかわらず、修正が未適用の問題として検知されるのは望ましくない。

これに対して、美浜式 ACPD では判別の基準をファイル・レベルの比較で行っており、より厳密な判別が行えるようにした。重要 APAR のリストは社内のデータベースから自動で取得が可能だが、ファイル・レベルの情報は書式化されていないため、手作業で転記が必要である。AIX 関連では月 15 件程度ではあるが、この作業の自動化は今後の課題である。

6. 実践と効果

6.1 診断件数

美浜式 ACPD の中心となる処理の実装を完了し、トラ

イアルとして従来の PowerHA のチェック項目のうち重要なものと、AIX の新規チェック項目合計 36 項目、そのほかに AIX 関連製品の重要 APAR 合計 562 個のチェックをチェックリスト仕様にのっとり実装した。トライアルであるため、メールの送信対象も AIX 関連製品のサポート・メンバーに限定した。8 月の1カ月間美浜式 ACPD の自動チェックを稼働させたところ、表1の結果となった。

表1. 診断件数内訳

	AIX	PowerHA	合計
診断資料数	345	54	399
問題検知資料件数	322	36	338
重要度1検知資料数	14	10	24

一月の診断資料数が 399 件であるため、年間では全体で約 4,800 件、PowerHA だけでは約 650 件の資料を診断することとなる。2009 年の PowerHA 自動診断の実績数は 22 件であり、これと比較すると PowerHA だけで 30 倍近く、AIX も含めると 200 倍以上の環境に対して診断を手作業なしで実施できる計算であり、高い価値を提供できることが分かる。399 件中 338 件の実に 85% のケースで問題が検知されており、非常に検知数が多いように思えるが、これは未適用の重要 APAR がほとんどのシステムで検知されるためである。重要 APAR 以外の項目のみに注目すると検知率は 50% 程度であり、かつ重要度がそれほど高くない問題が大部分である。

6.2 重大な問題の検知

注目に値するのは重要度 1 が 24 件検知されている点である。この内訳は表 2 の通りである。なお、1 つの資料で複数の問題が検知される可能性があるため、合計しても 24 件にはならない。

1 点目の netmon.cf の問題は社内向けの重要技術情報であるテクニカル・フラッシュでも報告されている重要な設定であるが、実際の運用環境では正しい記述がなされていないケースが少なくなく、障害の原因になることがある。この診断結果は設定ミスがいかにも多くの環境でなされているのかを客観的に示す重要な情報である。

また、一部のカーネル関連の設定変更や累積修正適用時のブート・イメージ更新後に必要なリポートがなされているかの確認が 2 点目であるが、これだけ多くの環境で正しくリポートがなされていないというのは意外な結果であり、新しい発見であった。この結果から元の資料を詳しく調査し、どのようなオペレーションの後のリポートがなされにくいのか突き止められれば、さらなる安定稼働のため

の有効な情報となりえる。

このように、多数の資料の診断結果を多角的に検討することにより各種問題の発生傾向について客観的かつ重要な情報が得られ、予防保守の品質をいっそう高めることが可能となる。

表2. 重要問題件数内訳

項目名	検知数
PowerHAのnetmon.cfの設定が正しくない。	10
ブート・イメージ作成の後リポートされていない。	10
リポートが必要なifixの適用後リポートされていない。	2
起動していない重要デーモンがある。	2
ページング・スペース枯渇のリスクがある。	1

6.3 診断所要時間

お客様がシステムの資料を送付する FTP サーバーはサポート・センターのほほすべての製品で同一であり、製品ごとにディレクトリー分けなどされていない。自動診断を実施する際、対象の資料が今回のトライアル対象の製品である AIX、PowerHA の資料がどうかの判別のために、製品を問わず国内のすべての新着の資料の zip や tar などの圧縮形式から展開して、AIX、PowerHA の資料に特有のファイルが含まれているかを確認する必要があった。そのため、多数の資料を展開するためにかかる時間が懸念点であった。結果として、全資料展開と一日あたり 20 件程度の診断、結果の送付を含めて 1 時間程度で完了し、実用上問題ないことが分かった。

処理時間のうち、圧縮形式からの資料の展開に要する時間が大部分を占め、現在すでに製品を問わずにすべての資料の展開は行っているため、仮に他製品に自動診断を展開しても診断の所要時間がリスクとなる可能性はごくわずかである。

6.4 コスト面での効果

例として、重要 APAR の有無の判別を例に試算してみる。重要 APAR の洗い出しとシステムの該当有無確認は手作業では一件あたり約 1 時間かかる。1 時間あたりのサポート・メンバーのコストを 10,000 円と仮定すると、AIX だけでは年間で 4,150 件の資料を調査するため、年間で 4,150 万円の削減ということになる。実際にはすべての案件で重要 APAR の有無の判別が行われるわけではないが、本来は実施されるべきであり、自動で行われる価値は大きい。この重要 APAR のチェックは美浜式 ACPD のチェック項目のごく一部であり、全体としてはより

大きなコスト削減効果となる。

7. おわりに

チェック・スクリプト仕様を標準化したことで多数の技術者からの協力を仰ぐための基礎は整ったが、美浜 ACPD の認知度は AIX 関連サポート・メンバー以外には不十分である。今後はこの美浜 ACPD の有効性やスクリプト標準化仕様を広く理解してもらい、それぞれの分野の技術支援のエキスパートの協力を得て、自動診断の適用範囲を拡大していきたい。

特に主要ミドルウェア製品への展開は急務であり、STSS オープンシステムズで提供している各製品の安定化ガイドラインの各項目の診断の自動化は重要である。

また、AIX の snap などの資料収集機能では収集されないログファイルや稼働情報は多数存在し、製品の資料収集機能のみに頼った調査には限界がある。一括では収集されない情報を、別途収集スクリプトなどを使って取得し診断することで、より幅広い診断をすることが可能となる。この観点から、製品の機能に頼らない独自の資料収集と調査の検討も必要である。

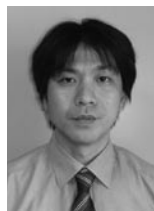
本論で取り組んだ自発的な診断と、上記で述べたようなさらに広範囲の自動診断の実現により、緊急度の高い障害発生時にも迅速に対応することができ、また予防保守の観点でもより品質の高いサービスを提供することが可能である。結果として、お客様の業務の安定稼働に大きく貢献することが期待できる。

謝辞

本論執筆にあたり、多くの有効な助言をいただいた STSS オープンシステムズ湯浅裕典さん、岸本清さん、末広正明さんをはじめ、System p サポートチーム・メンバーと第一システム技術支援メンバーに、ここに深く感謝の意を示します。

参考文献

- [1] IBM ソフトウェア・サポート・ハンドブック, <http://www.ibm.com/software/jp/supportguide/>
- [2] <http://www.ibm.com/support/jp/search/index.html>のAPAR検索結果
- [3] An architectural blueprint for autonomic computing, <http://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf>
- [4] シンプトン(症状)の深層を探る、第1回: オートノミック・コンピューティングのシンプトンフォーマット, <http://www.ibm.com/developerworks/jp/autonomic/library/ac-symptom1/index.html>
- [5] Autonomic Computing Toolkit Developer's Guide, <http://www.ibm.com/developerworks/autonomic/books/fpy0mst.htm#HDRAPPA>



日本アイ・ビー・エム株式会社
グローバル・テクノロジー・サービス事業
STSS オープン. 第一システム技術支援
アドバイザー・IT スペシャリスト

千葉 雄太 Yuta Chiba

[プロフィール]

2002年、日本 IBM に IT スペシャリストとして入社。AIX, PowerHA 製品の重要障害の解決支援、予防保守サービスのデリバリーに従事。金融、保険業界のお客様へのシステム技術支援サービスを提供する傍ら、デリバリーの自動化検討にも携わる。

.....



日本アイ・ビー・エム株式会社
グローバル・テクノロジー・サービス事業
STSS オープン. システム技術推進
アドバイザー・IT スペシャリスト

尾崎 恭一 Kyoichi Ozaki

[プロフィール]

2001年、日本 IBM 入社。Lotus Notes/Domino におけるクリティカル案件の問題解決に従事。2005年以降、システム技術支援サービス(STSS)のオプションサービス(システム・ヘルスチェック・サービスなど)にて、お客様システムの安定稼働に携わっている。

.....



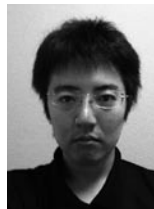
日本アイ・ビー・エム株式会社
グローバル・テクノロジー・サービス事業
STSS オープン. システム p テクニカルサービス
主任・IT スペシャリスト

森山 香織 Kaori Moriyama

[プロフィール]

2001年、日本 IBM 入社。入社以来 AIX における技術支援に従事し、重要障害の問題解決などに携わっている。現在はシステム技術支援サービス(STSS)を通してお客様システムでの早期問題解決と安定稼働に向けた活動を行っている。

.....



日本アイ・ビー・エム株式会社
グローバル・テクノロジー・サービス事業
STSS オープン. システム p テクニカルサービス
主任・IT スペシャリスト

井上 雅貴 Masaki Inoue

[プロフィール]

2002年、日本 IBM 入社。入社以来 AIX における技術支援に従事し、重要障害の問題解決などに携わっている。現在はシステム技術支援サービス(STSS)を通してお客様システムでの早期問題解決と安定稼働に向けた活動を行っている。