
製薬 ASP サービス信頼性向上への取り組み

富士通エフ・アイ・ピー 株式会社

■ 執筆者 Profile ■



2002年 富士通エフ・アイ・ピー（株）入社

八 巻 辰 二

■ 論文要旨 ■

製薬ASPサービス『PostMaNet（医薬品市販後調査データ収集システム）』は、医薬品が市販された後に、製薬企業が医療機関からその効果や副作用情報、有事事象などをインターネットを使って収集するシステムである。

副作用や人命にも関る情報を扱っているため、製薬企業はシステム障害に対して非常にシビアであり、加えて、ほぼ24時間稼働のシステムであるため、信頼性のある安定したシステムが必須である。

本論文では、このような厳しい要求に応えるために『PostMaNet』で実施した以下の3点の改善点についての取り組みとその成果について報告する。

1. サーバ監視の増強
2. 障害時連絡体制の強化
3. システム修正時のバリデーション（正当性の検証）実施

■ 論文目次 ■

| | |
|---------------------------------|-------|
| 1. 背景 | 《 3》 |
| 2. ASPサービス開始直後の問題点 | 《 4》 |
| 3. 問題点発生の原因と課題 | 《 4》 |
| 3. 1 監視項目 | |
| 3. 2 連絡・通知手段 | |
| 3. 3 保守作業 | |
| 4. 課題への取り組み | 《 5》 |
| 4. 1 監視項目の増強 | |
| 4. 2 連絡・通知手段の改善 | |
| 4. 3 システム修正時のバリデーション作業 | |
| 5. 実施後の効果 | 《 8》 |
| 5. 1 監視項目増強の効果 | |
| 5. 2 連絡・通知手段の改善の効果 | |
| 5. 3 システム修正時のバリデーション作業効果 | |
| 6. 新たな問題 | 《 10》 |
| 7. 現状の問題点と今後の課題 | 《 10》 |

■ 図表一覧 ■

| | |
|------------------------------------|-------|
| 図 1 PostMaNetの仕組み | 《 3》 |
| 図 2 製薬ASPマシン構成(概略図) | 《 5》 |
| 図 3 障害通知の流れ | 《 7》 |
| 図 4 モジュール確認作業 | 《 8》 |
| 表 1 Tomcatプロセスダウン回数 | 《 8》 |
| 表 2 仮想WEB監視でのタイムアウト回数 | 《 9》 |
| 表 3 閾値の調整 | 《 10》 |

1. 背景

製薬企業では新医薬品の市販後に適正使用情報の収集及び検討を行い、厚生労働大臣への副作用報告、添付文書の改訂、医療機関への情報提供などを行わなければならない。この適正使用情報の収集は医薬情報担当者（Medical Representative）が実施しており、医師に、医薬品が投与された患者の情報を調査票に記入してもらっている。この調査票は収集後、製薬企業が電子化し、評価・分析を行っている。回収数は1つの薬品当たり約 3000 例にもなる。また、一度承認を受けた新薬はその後永久に製造・販売できる訳ではなく、承認を受けてから 6 年後には厚生労働省に対して再審査申請を行い、薬の安全性や有効性などを報告する必要がある。さらに 5 年毎に再評価申請も行う必要があり、薬の安全性や有効性などを報告しなくてはならず、製薬企業にとって大きな負担になっている。

このような人的・経済的負担を削減し、市販後調査業務を効率的に行うために 2001 年 1 月、日本ロシュ株式会社様と共同開発したシステムが『PostMaNet』である。

『PostMaNet』は電子化された調査票を医療機関と製薬企業がインターネットを通じてやりとりするASPサービスである。

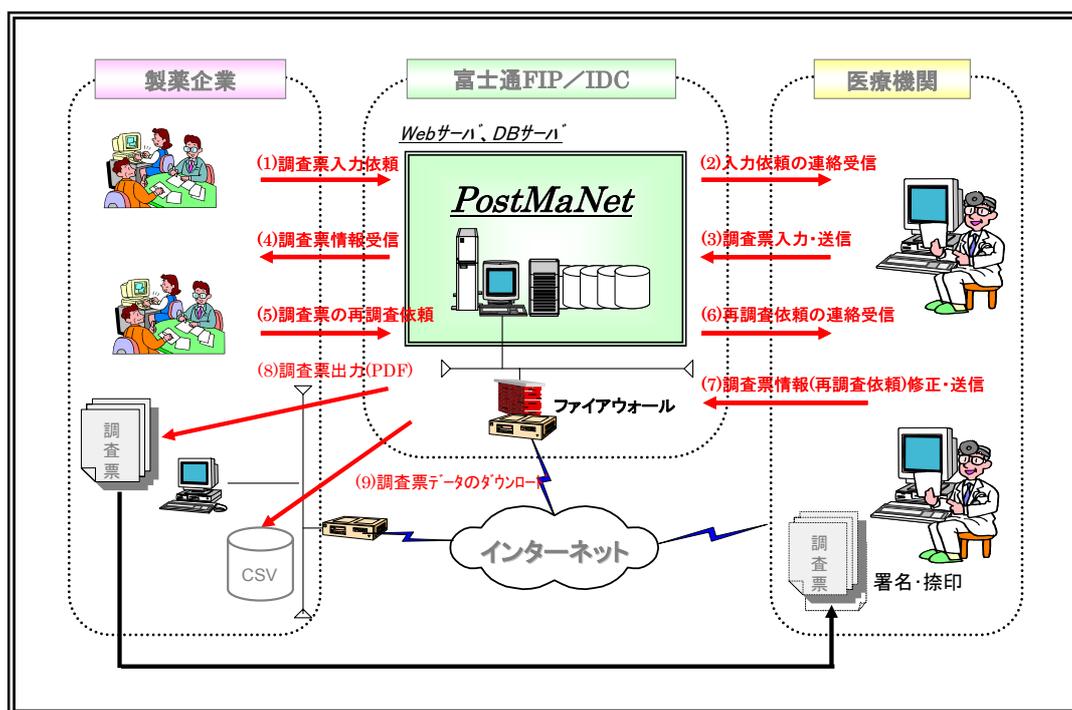


図1 PostMaNetの仕組み

同年 2 月から試験的な運用を開始し、2002 年 8 月より武田薬品工業株式会社様の大規模調査が始まった。

PostMaNet サーバは神奈川県川崎市武蔵中原にある富士通エフ・アイ・ピー・Internet Data Center (以下 IDC) に設置され、そのサービス提供時間は、平日は午前 6 : 00 ~ 翌日の午前 3 : 00、日曜日は午前 6 : 00 ~ 午後 1 時までとなっている。2003 年 9 月時点での登録ユーザ数は製薬企業、医療機関合わせて約 2000 人となっている。一日当たりの平均ログイン回数は約 150 回、ピーク時は約 1200 回にも達した。

2. ASP サービス開始直後の問題点

武田薬品工業様の大規模調査開始とともに潜在的に抱えていた様々な問題点や障害が噴出し始めた。

1. アプリケーションサーバの原因不明のダウン
2. 長時間に渡り障害未検知
3. プログラムモジュール置換え・確認ミス

具体的には以下のような障害が発生した。

WWW サーバ「Apache」とアプリサーバ「Tomcat」のプロセス監視を実施していたため、プロセスダウンは検知することが出来たが、それ以外のインベントリについては監視が未実施であったため、原因の特定が困難であった。

また、2002年9月3日の早朝5:30にTomcatがダウンしてから、担当者が出社する8:20までの約3時間もの間、利用できない状態が続いた。

モジュール置換えミスにより、2台のサーバ間、さらには本番系とテスト系でモジュールのバージョンが一致しないことが度々あった。

3. 問題点発生の原因と課題

3.1 監視項目

- 監視しているのは上述の二つのプロセスのみで、業務に必要なその他のプロセスやDBサーバ「ORACLE」の監視をしていなかった
- プロセス以外のインベントリの監視を実施していなかった

3.2 連絡・通知手段

- 監視ツールによって障害を検知しても、製薬ASPメンバー宛にメールを送るだけであったため、不在時は障害を検知することができなかった
- 障害時の連絡手順が明確化されていないため、第一報を連絡するまでに時間がかかったり、連絡そのものを忘れてしまう可能性があった

3.3 保守作業

モジュール修正後のバリデーション作業が未実施であった

4. 課題への取り組み

4. 1 監視項目の増強

(1) システム構成

製薬 ASP サーバの主な構成を以下に示す.

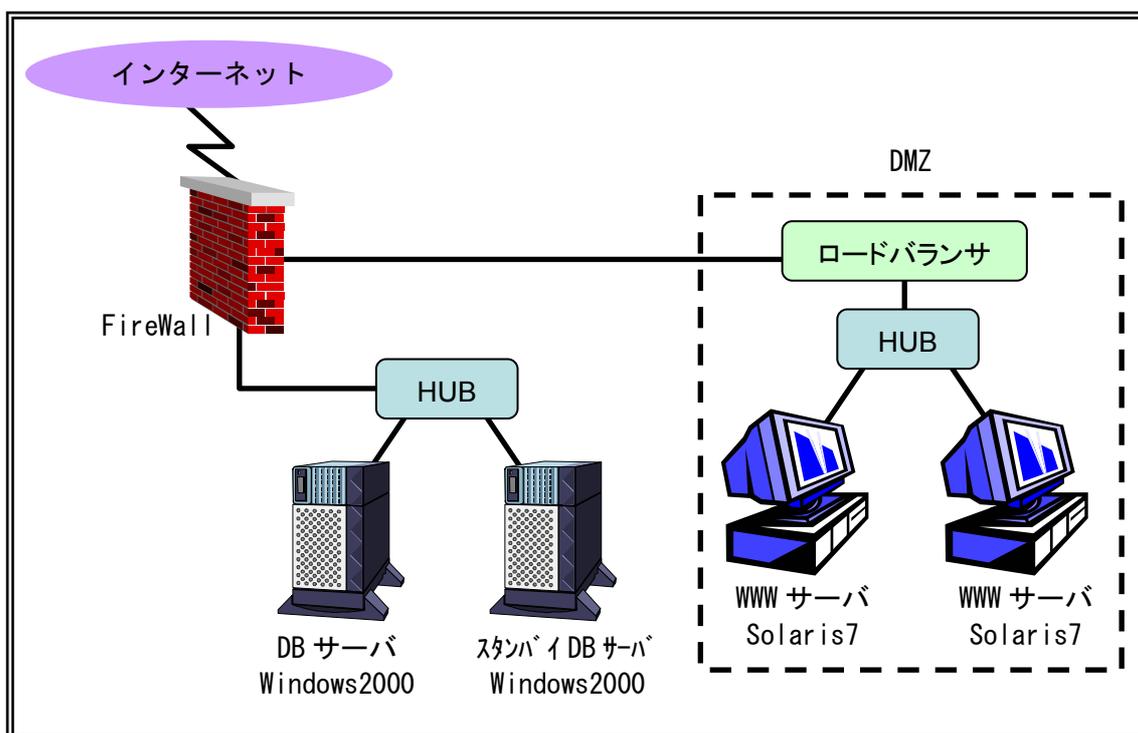


図2 製薬 ASP マシン構成(概略図)

(2) 取り組み指針

以下の指針に基づき、監視の増強を行った。

1. 早急な対応が求められていたため、導入が比較的簡単なツールを選択する
2. プロセス監視以外のインベントリについての監視を充実させる
3. 障害検知時に即座に情報を伝達できる仕組みを実装する

(3) 利用したツール

製薬 ASP のマシン構成は、WWW サーバに UNIX(Solaris)、DB サーバに Windows2000 を使用している。UNIX 機の方はシェルスクリプトを作り込むことで必要な監視項目を全て網羅できると判断した。Windows 機の方は「SystemWalker CentricMGR」の導入を検討したが、①巨大であること、② unnecessaryな機能が多いこと、③設定が複雑であり難しいこともあり、導入を見送った。代わりに、「BOM」(セイテクノロジーズ 株)を導入した。「BOM」は MMC(Microsoft Management Console)準拠のインターフェースを備え、起動サービスが最大2つと、機能は少ないが軽くて設定も簡単な監視ソフトであったことが導入の決め手となった。

(4) 監視項目

上述のプロセス以外に必要なと判断した項目は、以下の4項目である。

1. 資源監視
 - ・ CPU 使用率監視
 - ・ HDD 空き容量監視
 - ・ 空きメモリ監視
2. 性能監視
 - ・ 仮想 WEB 監視
 - ・ httpd プロセス数監視
3. ORACLE 監視
 - ・ インスタンス生存監視
 - ・ オープンカーソル数監視
 - ・ テーブルスペース監視
 - ・ セッション数監視
4. ログ監視
 - ・ シスログ (UNIX)
 - ・ イベントログ (Windows)

※仮想 WEB 監視とは、「check_http」というフリーツールを用いて html を仮想的に表示させ、Apache のステータスコードと応答時間をチェックするものである。我々は html の代わりに JSP (DB に接続し、検索結果を出力する) を使用することで Apache⇔Tomcat⇔DB のスルー監視を実現した。この結果、プロセスは生きているが P o s t M a N e t へのアクセスが出来ないような状況を検知することが可能となる。

(5) 監視方法

資源監視のような数値を取得するような項目に関しては、ワーニング(注意)レベルとクリティカル(危険)レベルの 2 段階の閾値を設定し、サーバの状態を細かくモニタリングするようにした。

プロセスダウン検知時は自動的に再起動を行うように設定し、瞬時に復旧するようにした。

4. 2 連絡・通知手段の改善

(1) 障害検知手段

障害検知時は製薬 ASP メンバー全員にメールを配信するように設定した。メールタイトルを以下のようなフォーマットで統一し、瞬時に状況を判断できるような工夫を行った。

[*イテ-レベル*_ID:XX] サーバ名: 監視項目 / ステ-ス

- ・ エラーレベル: ワーニングレベルは”INFO”, クリティカルレベルは”ERROR”
- ・ ID: 各監視項目毎に番号を振った
- ・ ステータス: 例えば空きメモリの具体的な値, あるいは「Apache down」などの

ように監視している項目の状態を出力する

各監視ツールによってワーニングレベルの障害が検知された場合、これまでどおり製薬 ASP メンバー全員の会社の PC にメールを投げるだけだが、クリティカルレベルの障害が検知された場合は、それに加えて製薬 ASP メンバー全員の携帯電話に SE コールの電話および E メール配信をするように改善した。その流れを図 3 に示す。

なお、IDC 統合監視サーバへのメッセージの通知には IDC 配付の共用監視ツール「CheckTool」を使用した。

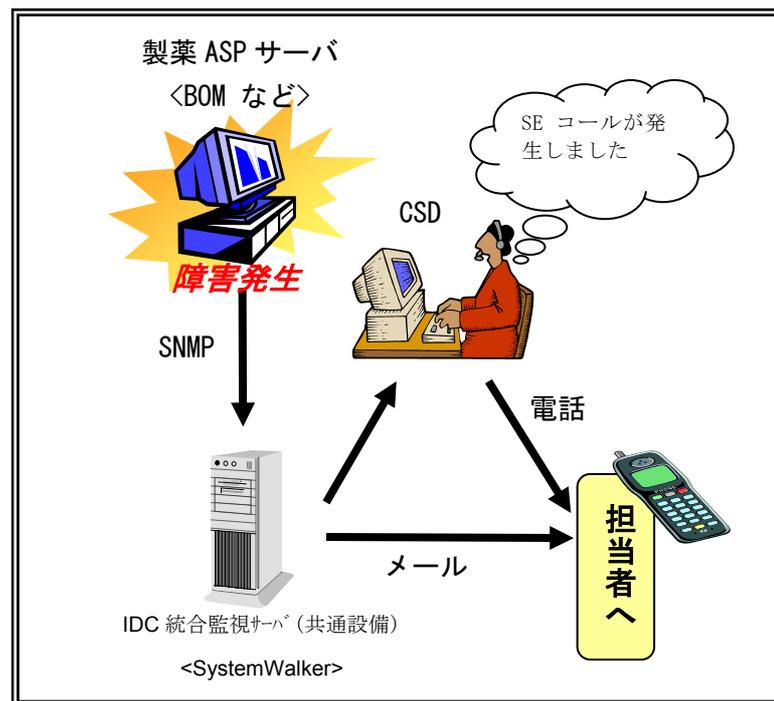


図 3 障害通知の流れ

(2) 顧客への連絡手段

手順書を作成し、社内連絡体制および顧客への連絡体制を明確化した。また、顧客へのメールタイトルを以下のようなフォーマットで統一した。

プロトコール略名 障害連絡：yyyymmdd_障害内容(第 N 報)

- ・ プロトコール略名： 調査テーマ(プロトコール)の略名 (3~4 文字程度)

また、プログラムに不具合が生じた場合や DB サーバがダウンしたときなどは、トップページに障害アナウンスを掲載し、ID/パスワード入力欄を非表示にして、影響を受ける利用者の拡大を防ぐ工夫を行った。

4. 3 システム修正時のバリデーション作業

モジュール修正後に以下の作業を行い、バリデーションを実施した。

- ・ diff コマンドを用いて本番系とテスト系モジュールを比較する

- モジュール修正時は、左右分割型ウィンドウを持つファイル転送ツールを使用して置き換える。置換え後は左右の画面を比較して確認を行い、各フォルダ単位で画面のハードコピーを取る。

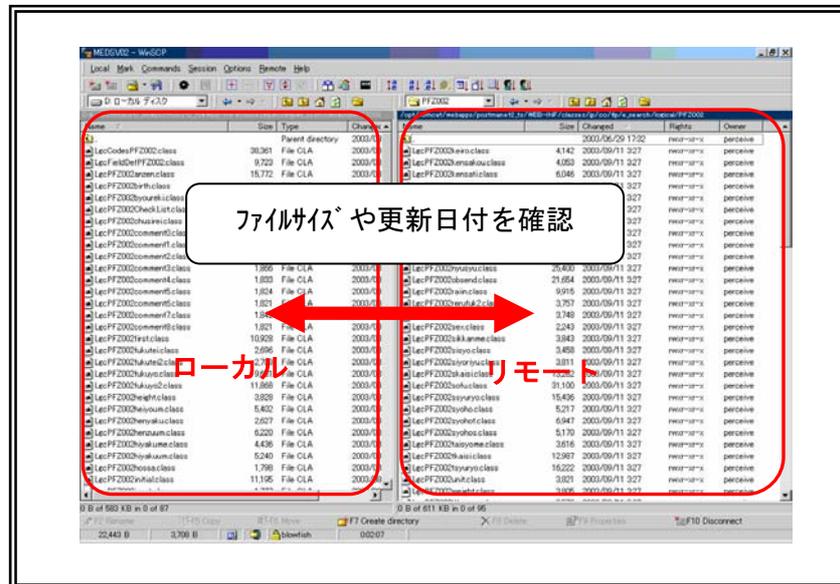


図4 モジュール確認作業

5. 実施後の効果

5.1 監視項目の増強の効果

約20項目にも渡る監視項目の追加により、多角的な監視が可能となり、これまで発生していた様々な障害の原因特定を行うことが可能となった。以下にその具体例を示す。

ケース1：Tomcatのプロセスダウン

ダウン時に共通する状況として、空きメモリが少ないことが判明した。このため以下の作業を実施した。

- WWWサーバは日曜日の保守時間に必ず再起動する
- Apache+Tomcatを毎日深夜の保守時間に必ず再起動する
- この結果Tomcatのプロセスダウンが起こる確率が1/3以下になった。

表1 Tomcat プロセスダウン回数

| | 2002年9月～ 2003年3月まで | 2003年4月～ 2003年9月29日まで |
|----|-----------------------|--------------------------|
| 回数 | 9 | 2 |

ケース2：アクセス不可

仮想WEB監視によるタイムアウトエラーが度々発生し、PostMaNetへのアクセスが不可能になる障害が発生した。この障害に共通する状況として、

- Apache の httpd プロセス数が増加し続ける
- 利用者はそれほど多くない
- Tomcat のログファイルに同じメッセージが延々と吐かれ続ける

ことが確認できた。このことから、Tomcat 側で何らかの問題が発生したために、Apache からのリクエストを処理できないと判定した。Tomcat のログや java のスタックトレースを調査した結果、例外情報出力の中に特定の文字が混ざっている場合に、例外処理が無限ループのような状況に陥っていることが判明した。

ループを回避するように修正したライブラリファイルにより上記障害が発生しなくなった。

ケース 3： 仮想 WEB 監視タイムアウト

仮想 WEB 監視によるタイムアウトエラーが断続的に発生する障害が発生した。

- 利用者はそれほど多くない
- Apache および Tomcat のログに異常なメッセージなし
- CPU 利用率および HDD 使用率は正常

上記状況から、製薬 ASP サーバ側ではなく、ネットワーク側に問題があると判定し、IDC・インターネットサービス部に調査を依頼した。その結果、FireWall サーバでメモリ不足が発生し、リクエスト処理を正常に行えないことが判明した。FireWall のメモリパラメータの変更を 2003 年 5 月 30 日に行った結果、上記障害がほとんど発生しなくなった。

表 2 仮想 WEB 監視でのタイムアウト回数

| | 2002 年 9 月～ 2003 年 5 月まで | 2003 年 6 月～ 2003 年 9 月 29 日まで |
|----|-----------------------------|----------------------------------|
| 回数 | 40 | 3 |

5. 2 連絡・通知手段の改善の効果

障害時に監視ツールから配信されるメールにより、以下の効果が得られ顧客への信頼感もアップした。

- 障害をリアルタイムで検知することが可能となり、最大 5 分以内での障害対応が可能となった
- 迅速な初期対応により、顧客への報告もスムーズにいくようになった
- 統一されたメールフォーマットにより、顧客との情報共有がうまく図れるようになった

5. 3 システム修正時のバリデーション作業効果

diff コマンドによるモジュール比較を行うことによって本番系とテスト系のモジュールに差が生じることは無くなった。また、ファイル転送ツールのウィンドウのスクリーンショットを取ることで、モジュール置換え作業の結果を確認することが出来、作業自体のバリデーションを行うことが可能となった。

6. 新たな問題

このような改善によって、障害の件数は少なくなったが、逆に SE コールの件数が爆発的に多くなってしまった。多いときで1日約200通もの SE コールメールが会社の端末および製薬 ASP メンバーの携帯電話に届けられたため、確認作業に追われ、業務に集中することができなくなってしまった。更に重要な SE コールを見逃してしまう可能性もあった。監視ツールで設定している閾値が厳しかったことが原因であったため、運用に合わせた適正な閾値への調整を行った。

表3 閾値の調整

| 項目 | 調整前 | 調整後 |
|-------------------------|---------------|-------|
| 仮想 WEB 監視 (レスポンスタイム) | 7 秒 | 15 秒 |
| CPU 使用率 | 5 分間平均 90% | なし |
| 空きメモリ ※トータル 2 GB | 1GB | 200MB |

仮想 WEB 監視ではネットワークトラフィックやサーバ負荷により瞬間的にレスポンスが悪くなることがあったため、レスポンスタイムの閾値を7秒から15秒に増やした。CPU 使用率は、特に Windows 機では、夜間バッチや DB バックアップ時に CPU 利用率が非常に高まることから、SE コール自体の設定を外した。空きメモリは、1週間連続稼働を行った場合、1GB を下回ることが度々あり、またメンテナンス時間でなければマシンの再起動ができないため、閾値を1/5に小さくした。

これらの調整によって、無駄な SE コールの件数が大幅に減り、重要な障害時以外は SE コールの発生件数はほぼ0件となった。

7. 現状の問題点と今後の課題

監視の充実や連絡手段の改善を行った結果、現在では稼働率 99.9%の非常に安定したシステムを構築することができた。顧客の信頼度も増し、サーバ管理作業に費やす時間も削減することができた。

障害時に担当者の携帯電話にも SE コールメールを配信することで、深夜あるいは休日でも障害状況を即座に把握することが可能となったが、製薬 ASP メンバー全員が社外にいる場合、本社ビルもしくは武蔵中原の IDC に駆けつけるまでに時間がかかってしまう。IDC・インターネットサービス部との連携や、自宅 PC から PostMaNet サーバへアクセスできるような環境を構築する必要がある。

また、プログラムのバグを、モジュール修正後の動作確認で検知できなかったことが度々あった。これを防ぐために自動テストツールの導入を検討中である。

MSP サービスの一つである「評価/性能レポート」についても、運用監視当初よ

り，月毎の日付別，時刻別，曜日別の利用状況を調査し，顧客への報告を実施している．
今後も継続して信頼性向上への取り組みを行っていききたい．

参考文献

- [1] 「MR ガイド 情報 109」，(株)ミクス，1996 (ISBN4-89587-321-8)
- [2] 「MSP とは？」，<http://www.msp-sp.net/>