Hewlett Packard Enterprise

DB システム開発者のためのサーバープラットフォーム移行ガイド

UNIX の世界に Intel アーキテクチャーのサーバーが進出し業界勢力図が変わり始めている現在の市場動向を分析し、そんな時代のサーバープラットフォーム選びはどうあるべきか、そのポイントを探ります。

《連載期間：2004 年 9 月～2005 年 1 月》

―目次―

第 1 回

将来を見据えたプラットフォーム選び

2004 年 9 月　テクニカルライター 田中敏夫

今使っているサーバーのリース期限が近付き、これからどうしようかと悩んでいるシステム担当者も多いのではないだろうか。あるいは、事業の拡大に伴ってサーバーで処理する情報量が増えたため、IT 設備を強化すべく、サーバーマシンのリプレースを検討している企業があるかもしれない。そこで問題になるのが、新たに導入するサーバー環境の選び方だ。サーバー市場では、2〜3 年後にはローエンドでも 64 ビットプロセッサーが一般的になると予想されている。また最近は、UNIX の世界に Intel アーキテクチャーのサーバーが進出し、これまでのサーバーベンダの勢力図が変わり始めている。そんな時代のサーバープラットフォーム選びはどうあるべきか。現在の市場動向を分析し、そのポイントを探る。

プラットフォームで決まるビジネスの将来

IT がビジネスに欠かせない重要な存在となった現在、そのプラットフォームとなるサーバー環境をどのように選択するかが、将来のビジネスの成否を大きく左右する。プラットフォームの選択を誤ると、サーバー環境全体の維持管理に非常に多くのコストがかかってしまうからだ。たとえば、拡張性に乏しいサーバーを選ぶと、数年経って処理能力が不足してきたときに、サーバーマシンをいくつも増設することになる。一般に、サーバーの台数が増えると、IT の所有コストが大幅に増加する。これは単に、追加するマシンのリース代が増えるというだけの問題ではない。社内のあちこちに分散し、複雑に構成されたサーバーを管理するため、より多くの人手が必要となり、それがそのまま人件費の増加につながるのだ。また、サーバーの台数が増えれば、マシンが占有するフロアの面積も増える。さらに、ミドルウェアやアプリケーションに追加の作り込みが発生したり、サーバー間のワークロード管理が増えたりするなど、予想外のコストがかさんでいく。そのようにして膨らんだ IT インフラの TCO は、自社本来のビジネスへの積極的な投資を阻害する要因になる。逆に、サーバーの選択がうまくいって、その後何年も大きな変更なしに IT インフラとして十分な能力を発揮し続けることができれば、自社の得意とするビジネスを伸ばすことにエネルギーを集中できる。
IT のプラットフォームとしては、OS も重要な要素となる。自社の将来を託す情報基盤として考えた場合には、この OS についても慎重に選択する必要がある。たとえば、Linux はオープンソースの OS としてサーバー市場でも導入が進んでいるが、SCO による Linux 訴訟問題は、現在も様々な企業を巻き込んで進行中だ。また、商用 UNIX であっても、競合他社との激しい競争に敗れ、市場から消えてしまうものもある。いったん導入したら簡単には変えられない OS であるから、企業の IT システムとしては、今後数年のスパンで安心して利用できるものを選択しなければならない。

こう考えると、IT プラットフォームそのものが、将来の自社の強みにも弱みにもなり得ると言える。したがって、サーバー環境を選択するときには、様々な角度から自社のニーズを見つめ直し、慎重に決断を下す必要がある。

市場の変革期である今こそ、新しいプラットフォームを考えるとき

現在は、OS を含めたサーバープラットフォームの変革期である。UNIX サーバー市場では、特にローエンドの領域でサーバーベンダ間のシェア争いが激しくなっている。これまで、この領域では Solaris サーバーが台数ベースでかなりリードしていたが、ここ数年は HPE や IBM といった競合ベンダとの競争が激化し、ローコストの IA-32／Linux サーバーの利用も特定分野では多くなってきた。このような状況を考えると、現在サーバーのリプレースを考えているユーザーにとっては、今がちょうど他のプラットフォームへ移行する良いタイミングではないだろうか。この機会に、将来性のあるプラットフォームへ乗り換えれば、今後のビジネスをスムーズに展開することができるだろう。

では、実際にサーバー環境をリプレースするとして、いったいどのような基準でサーバー選びをしたらよいだろうか？ 現在の UNIX サーバー市場の動向に照らしてみたとき、有力な答えの 1 つとして挙がってくるのが、IA（Intel アーキテクチャー）サーバーだ。中でも、最近になって新たに登場した 64 ビットプロセッサー搭載のサーバーは、今後のサーバーの主流になるものと見られている。

OS の幅広さが最適なソリューションを実現する

IA サーバーと言えば、ローエンドのサーバー市場で既に広く導入されている。しかし、これは 32 ビットプロセッサーを搭載した IA-32 と呼ばれるサーバー群である。ハイエンド UNIX の世界では、以前から 64 ビットコンピューティングが一般的だったが、最近になってインテルや AMD といったプロセッサーメーカーがサーバー市場向けに次々と 64 ビットプロセッサーを投入したことから、ローエンドサーバーの領域でも 64 ビット化の動きが本格化してきた。サーバー市場では、早ければ 2〜3 年後に 64 ビットが一般的になるのではないかと予想されている。

IA サーバーの最大のセールスポイントは、その価格性能比の高さにある。これは、64 ビットになっても基本的には変わらない。しかし、これに加えてもう 1 つ注目しておきたい点は、IA サーバーでは複数の OS がサポートされることである。IA サーバーなら、UNIX 以外の OS として Windows や Linux を選択できる。どの OS にも長所と短所があるが、OS を自由に選択できるのであれば、目的のソリューションに合わせて最適な OS を選択できる。メモリ搭載量、HA（ハイアベイラビリティ）構成、セキュリティ、スケーラビリティなどが重要であれば、商用 UNIX が有力な選択肢になるだろう。また、ローエンドサーバーであれば、Linux や Windows に移行するケースが多いかもしれない。

OS 選択の自由度の高さは、サーバー環境の将来性を考えた場合にも大きなポイントとなる。最初は UNIX で始めて、後から Linux や Windows に変えるということも可能だ。自社のビジネスニーズの変化に合わせて、柔軟にシステム構成を変えることができるのである。

これまでのプラットフォームに縛られることなく、必要とされるソリューションの実現に主眼を置いた選択ができるというのは、TCO の削減にとっても、自社のビジネスを伸ばすための積極的な投資に対しても、極めて有利に働くだろう。

今後のサーバー市場の動向を示す鍵は IA

既に述べたように、今、サーバー環境をリプレースするのであれば、64 ビット IA サーバーが注目株だ。中でも、Itanium を搭載したシステムについては、将来的にハイエンドからローエンドまですべての領域で主流になると見られている。図 1 を見てもらいたい。これは、今後のサーバー市場における RISC プロセッサーと Itanium プロセッサーファミリ（IPF）の出荷動向予測である。図中には出ていないが、Itanium の出荷数は 2005 年に POWER プロセッサーを、2006 年には SPARC プロセッサーを超えると予想されている。

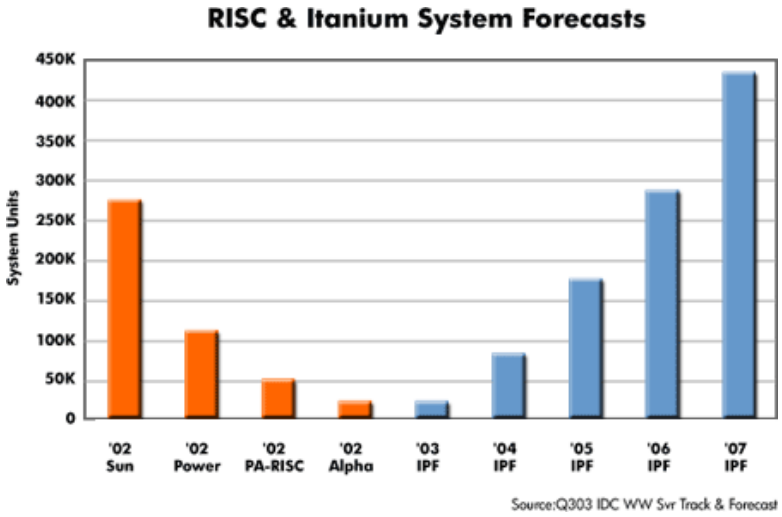


図 1 サーバー市場における Itanium システムの出荷動向予測

注：2003 年日本国内 UNIX サーバー売り上げベースで HPE 33.2% 、SUN 22.9% 、IBM19.0% 、富士通 17.0% のマーケットシェア

（出典：IDC Worldwide Quarterly Server Tracker, Q104 Unix OS/Risc & EPIC（IA-64）Server, Revenue）

また、Sun を除く大手サーバーベンダは、いずれも Itanium の採用を進めている（図 2）。これは、今後の Itanium の成長に対するサーバー市場全体の期待がいかに大きいかを示すものだ。

サーバー向けの新しい 64 ビット CPU としては、他にも Intel アーキテクチャーの Xeon があるし、Sun も採用している AMD の Opteron プロセッサーなどがある。これらのプロセッサーの特徴は、32 ビットアーキテクチャーとの高い互換性だ。既存の 32 ビットのソフトウェア資産を守れるという意味で強力な武器になるが、実はこの互換性は、32 ビットアーキテクチャーを拡張して 64 ビット対応としたことで得られたものである。つまり、最初から 64 ビットに最適化されて作られたアーキテクチャーではない。したがって、これらのプロセッサーのターゲットとなるのは、32 ビットシステムから移行するユーザーが大きな割合を占める。それに対し、真の 64 ビットシステムとして開発された Itanium が将来的にターゲットとする市場は、既に 64 ビットが使われてきたハイエンドの領域からローエンドの領域までと幅広い。Intel アーキテクチャーの UNIX サーバーとして考えたときには、やはり Itanium システムが有力候補となるだろう。

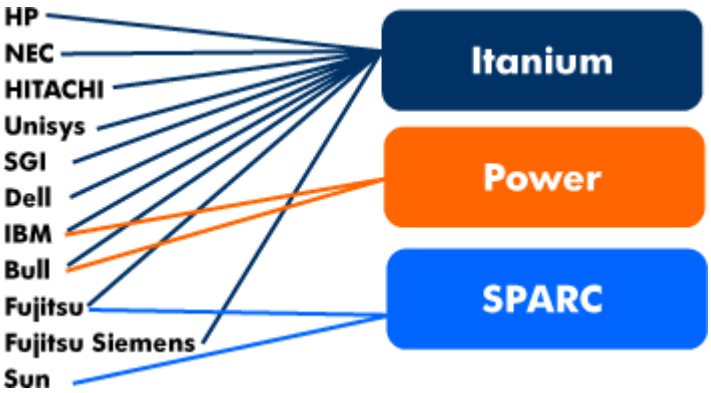


図 2 大手サーバーベンダの CPU サポート状況

新 64 ビット CPU の本命は Itanium ?

Itanium は、インテルがハイエンドサーバー市場への進出を目指し、長い年月をかけて開発してきたプロセッサーだ。32 ビットの延長としてではなく、真の 64 ビットプロセッサーとして完全に一から設計された新しいアーキテクチャーである。強力な浮動小数点演算能力、UNIX、Windows、Linux など複数 OS のサポート、そして IA ならではの優れた価格性能比が売りだが、これまでの UNIX サーバーで主流であった RISC アーキテクチャーと根本的に異なるのは、EPIC （Explicitly Parallel Instruction Computing： 明示的並列命令コンピューティング）と呼ばれる技術が導入されていることである（図 3）。

従来のコンピューティングアーキテクチャーでは、CPU で処理される命令を並列実行するために、プログラムの並列性をプロセッサー内部で動的に判断していた。しかし、この方式では並列性を発見するための技術に限界があり、命令を並列実行するための実行ユニット群に大量のアイドル時間が生じ、プロセッサーの処理能力を十分に使い切ることができなかった。EPIC アーキテクチャーでは、並列性の探索ロジックをコンパイラ側に実装し、効率的に並列実行できるマシンコードを生成させる。これによって、実行ユニットの利用効率が高まり、同クロック数でより多くの命令を処理できるようになる。

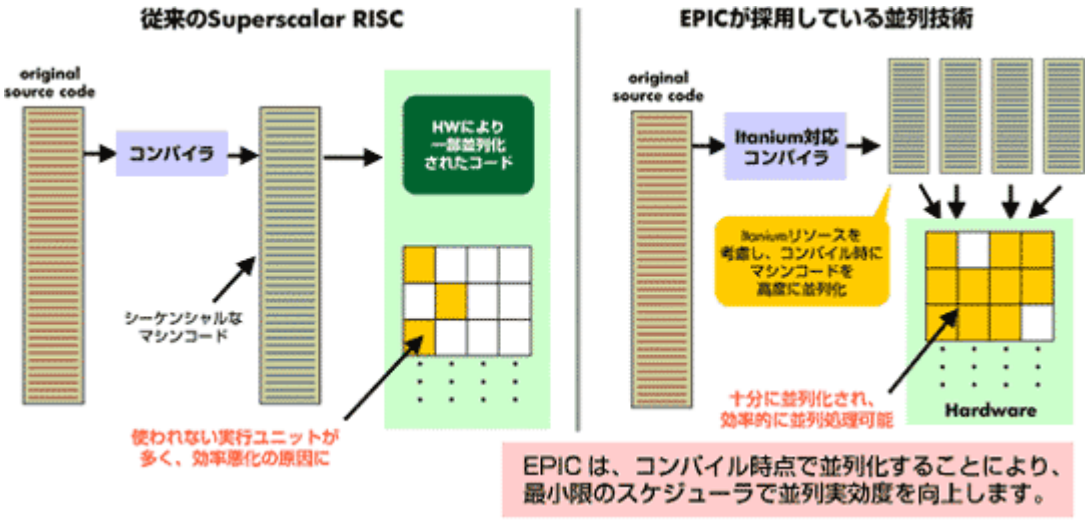


図 3EPIC アーキテクチャー

従来のプロセッサーアーキテクチャーの限界が命令実行の並列化にあったことから、EPIC は現在のコンピューティングの限界を打ち破る技術として、業界全体から注目されていたのである。この技術を実装した Itanium プロセッサーが多くのベンダによってサポートされ、今後、サーバー製品のメインストリームになると予想されているのも不思議ではないだろう。

Itanium 搭載マシン選択の基準

先ほど見たように、Itanium マシンをラインナップに持つベンダは、Sun を除いて主要なベンダのほぼすべてに渡っている。ユーザーとしては、選択肢が多いに越したことはないが、どれを選択すればいいか迷いやすくなるのも事実だ。たくさんの選択肢の中から最適なものを見つけるには何かしらの基準が必要だが、では Itanium マシンを選ぶ場合の基準とはいったい何だろうか？

Itanium プロセッサーについて語るときには、やはり HPE を外すことはできない。そもそも、Itanium プロセッサーはインテルと HPE の共同開発によるものだからだ。このような経緯から、HPE はどのベンダよりも Itanium プロセッサーの特性についてよく知っていると言える。実際、Itanium 搭載サーバーを最初に市場に送り出したのは、他ならぬ HPE 自身である。Itanium プロセッサーを成功させるため、長い間研究を重ねてきた成果は、図 4 に示した TPC-C ベンチマークの結果からも明らかだ。

昨年の 11 月には、Itanium 搭載サーバー（Integrity サーバーシリーズ）が TPC-C ベンチマークで世界初の 100 万トランザクション／ 分を達成し、本稿の執筆時点（ 2004 年 7 月 3 日現在）も IBM の UNIX サーバーとともに、ランクの上位を Itanium 搭載システムが占めている。また、Itanium プロセッサーでは、IA-32 サーバーに比較してパフォーマンスが 2.2 倍向上するとインテルから発表されているが、これが実現すれば、その競争力は他の CPU を圧倒すると思われる。

さらに、Oracle ユーザーが IA サーバーにリプレースする場合には、HP 社と Oracle 社の 20 年来の提携関係にも注目したい。つい先日（ 2004 年 6 月 30 日）には、日本ヒューレット・パッカード 、日本オラクル、シスコの共同検証事業が発表されたばかりだ。この共同検証では、日本オラクル内に HP の Itanium 搭載サーバーを 50 台と、StorageWorks EVA 5000 などを使った 35TB のストレージを設置し、メインフレームの置き換えや、Oracle 10g を使ったグリッドコンピューティングの検証／ノウハウの蓄積を、1 年近くに及んで実施する。Oracle を使用した実運用システムにおいては、そのプラットフォームの 37%が HP-UX を利用しているという報告もある。TPC-C ベンチマークにおいて世界で初めて 100 万トランザクション／分の記録を樹立したのも、Oracle と組み合わされた HP の Itanium 搭載システムである。このような両社の密接な協力体制は、今後の Oracle＋HP Itanium 搭載サーバーの成長を支える大きなアドバンテージとなるだろう。

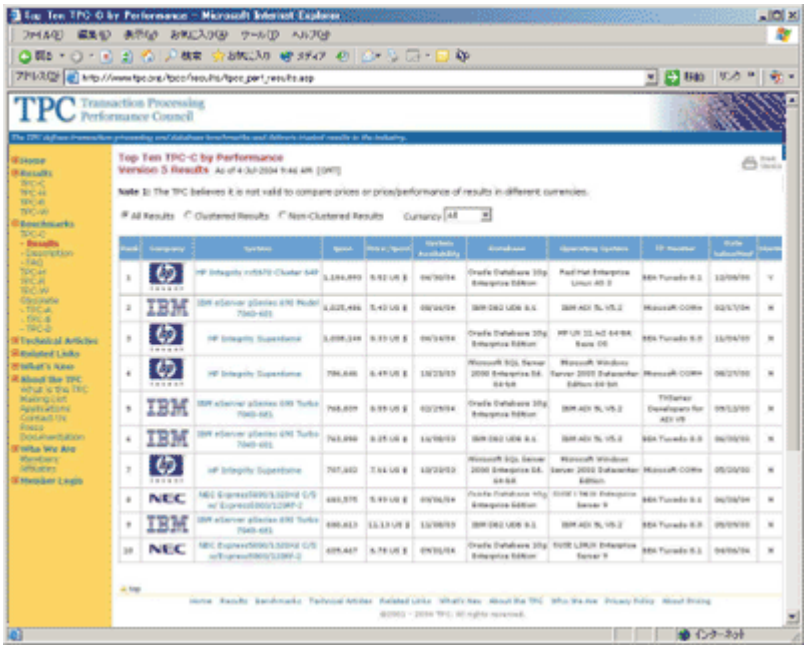


図 4TPC-C ベンチマークのランキング（2004 年 7 月 3 日現在）

異種プラットフォームへの移行に落とし穴はないのか？

実際に IA サーバーがいいと思っても、やはり異なるプラットフォームへの乗換えを決意するには、かなり勇気が要るものだ。新しい環境に移行するかどうか決断するとき、最も大きな懸念材料となるのは、既存のアプリケーションをどうするかということだ。数年前に構築したシステムを再利用するか、あるいはスクラップ＆ビルドするかの判断がまず必要になる。以前構築したシステムが C/S システムであれば、Web ベースへの移行も検討が必要であろう。既に数年運用したシステムをそのまま移行するには、同一ベンダプラットフォームのアップグレードでも、アーキテクチャーの異なるプラットフォームへ移行する場合でも、OS のレベルから入れ換えが必要だ。当然、その上に乗るミドルウェアやアプリケーションも再ビルドしなければならず、部分的にソースコードの修正が必要になるかもしれない。場合によっては、システムや仕様を再検討せざるを得ないケースもある。また、いったん新しい環境へ移行した後も、Web サーバー、アプリケーションサーバー、データベースサーバーの各コンポーネントは常にバージョンアップが繰り返されていく。セキュリティ要件を満たそうとして、いずれかのコンポーネントを最新のものにすれば、それが原因で業務アプリケーションに支障が出るかもしれない。

そんなプラットフォームの移行に不安を覚え、二の足を踏んでいるシステム担当者も少なくないはずだ。そこで、次回以降は、異種プラットフォームへの移行の具体例として、Solaris から HP-UX へのアプリケーションポーティングの事例を紹介する。第一線で活躍する SIer の指導の下、これまでにポーティングの経験のなかったシステム担当者が、移行作業の A から Z までを体験した。この記事を読めば、異種プラットフォームへの移行が実際にはどのようなものか、おおよその雰囲気は掴めるだろう。読者が移行を検討するうえでの参考になればさいわいだ。

【コラム】TPC-C で好成績を収める Integrity サーバー

Integrity サーバーは、日本ヒューレット・パッカードが販売する Intel® Itanium®プロセッサ搭載サーバーである。同社が提供する PA-RISC サーバーの今後を担う主力サーバーであるが、最小ではラックマウント型の rx1600、最大では大型ハウジングラックサイズの Superdome と多様なラインナップが揃っており、小オフィスのファイルサーバーからデータセンターでのホスティングサービスに至るまで、サーバーとしてのあらゆる用途をカバーする。最大の特徴は複数の OS に対応していることで、UNIX（HP-UX）、Linux、Windows のいずれの OS も利用できる。また、独自のパーティション技術によって、1 台のハードウェア上で複数の OS を同時運用することもできる。このパーティション技術を利用すれば、これまで用途別/OS 別に複数台用意していたサーバーを Integrity サーバー1 台に集約でき、今話題のサーバーコンソリデーションを実現できる。既にインテルより発表された Itanium プロセッサの新シリーズ（コードネーム Montecito）もサポートされるとともに、十分なバス帯域幅を持っているので、将来にわたって最新のサーバー環境を提供可能だ。

第 2 回

実践！＜Solaris→HP-UX/Linux＞アプリケーション移行ガイド　Part.1

2004 年 10 月　テクニカルライター 大戸 英樹

今回からは、Solaris 上で稼動していた Web アプリケーションを HP-UX 上で動かすためにはどんな作業が必要かということを、具体的なシナリオを追いながら見ていきます。ここでは、HP-UX にそれほど詳しくない担当者がポーティング作業を行うという状況を想定しています。実際の現場でぶつかりそうな問題点も網羅していますので、ポーティングを実践するときの助けになるでしょう。実践編 Part.1 の今回は、ポーティングのシナリオとサーバー構築までを説明します。

担当者 S
とある企業の IT 関連部署に勤務しています。Solaris 担当なのですが、ここのところエンドユーザーとの打ち合わせの後には必ず、何か考えている様子でした。この実験に多いに期待している反面、不安でいっぱいのようです。

同僚 H
担当者 S と同年代。先日、HP-UX に関する教育コースに参加して、Integrity サーバーを知りました。そこを見込んで、HP-UX についてアドバイスをすることになりましたが、「最初から聞くなよ。まず自分でやってみな」と、担当者 S を一喝しています。

ポーティングのシナリオ

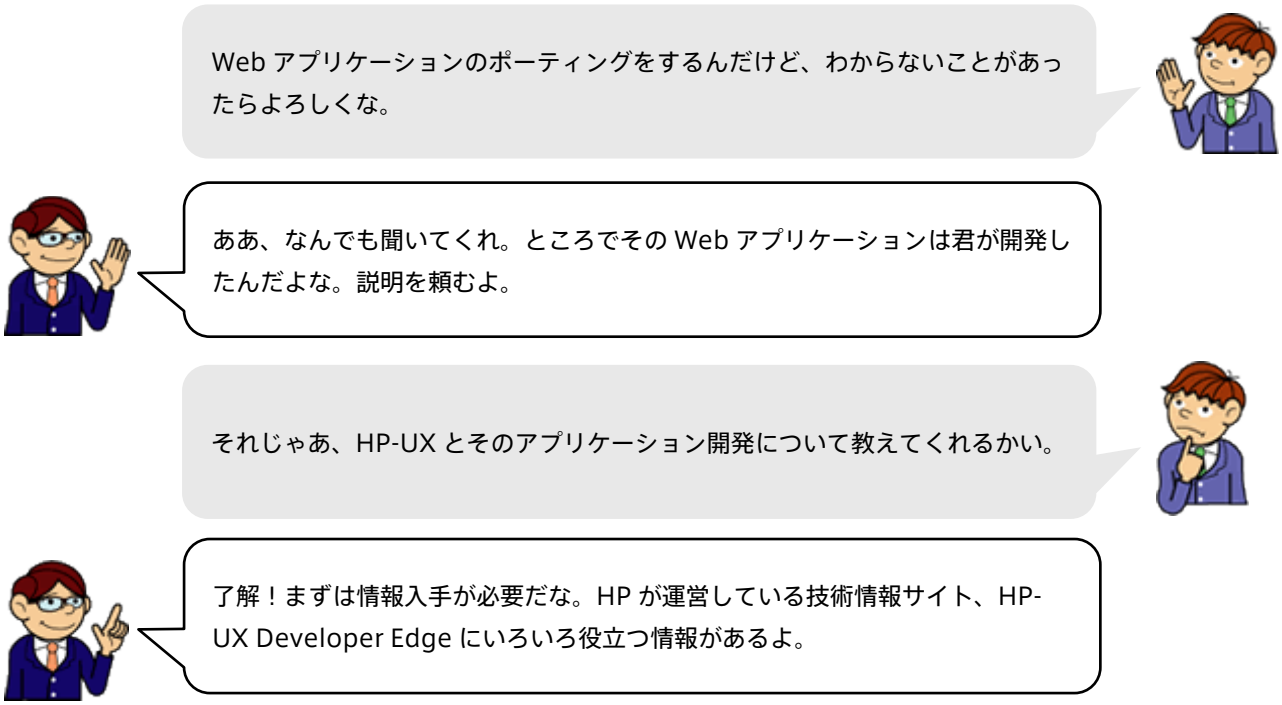
ある企業が Web サーバーの置き換えを検討しています。既存の Sun Enterprise 250 サーバーは導入からすでに 4〜5 年がたち、リース期間の終了も間近になった今では、そろそろ現状に対応できなくなっているからです。現有資産としての Web アプリケーションや、Oracle マスターをはじめとした技術者のスキルのことを考えると、引き続き Sun の新しいサーバーを導入するのが最も無難な選択ですが、今回はパフォーマンスやセキュリティのさらなる向上を目指して、別のアーキテクチャーを採用することにしました。新たに導入を予定しているのは、Intel Itanium プロセッサー搭載の Integrity サーバーrx2600 です。

今回の実験では、既存の Sun サーバーの Solaris7 上で稼動している 2 種類の Web アプリケーションを、Integrity サーバーの HP-UX 上にポーティングします。

実験の担当者 S は、ここ数年、Solaris7 を担当していた技術者で、Solaris7 以外のことはほとんど知りません。そこで、HP-UX に詳しい同僚 H の手を借りつつ実験を進めることになります。HP-UX へのポーティングを完了させるまでにはいくつかの問題にぶつかりましたが、同僚 H のアドバイスのおかげで、既存の Web アプリケーションをほとんど修正することなく HP-UX にポーティングすることができました。今回から数回に分けて、このポーティング作業の一部始終をご紹介します。

対象となる Web アプリケーションの特徴

今回のシナリオでは、「Web セミナー受付システム」と「会員管理・メール配信システム」という Web アプリケーションをポーティングします。



- **「Web セミナー受付システム」アプリケーション**

この Web アプリケーションでは、どのようなセミナーがあるのかを検索して、その詳しい内容を閲覧できます。閲覧者は、気に入ればその場でセミナーを予約することができます。会員登録と登録内容の変更も Web 上で行うことができます。また、セミナー主催側は、このシステムを利用して受講者にメールを送付できます。

- **「会員管理・メール配信システム」アプリケーション**

この Web アプリケーションは、会員管理(会員登録・検索・外部ファイル読み込み・メール配信起動/停止)やサービス管理(メール配信・予約管理・サービス時間管理・サービス内容管理)をするためのシステムです。

対象となる Web アプリケーションの特徴

まず、今回のポーティングで扱う 2 種類のサーバーの違いを見てみましょう(表 1)。ポーティング元の Solaris サーバーとポーティング先の HP-UX サーバーは、販売時点の平均的なスペックのものを用意しています。

	Solaris	HP-UX
機種名	Sun Enterprise 250	Integrity サーバー rx2600
CPU	UltraSPARC IIs 400MHz×1	Itanium 1.5GHz×2
メモリ	512MB	4GB
ハードディスク	17.7GB×1	72GB×2
OS	Solaris 7	HP-UX 11i v2(B.11.23)

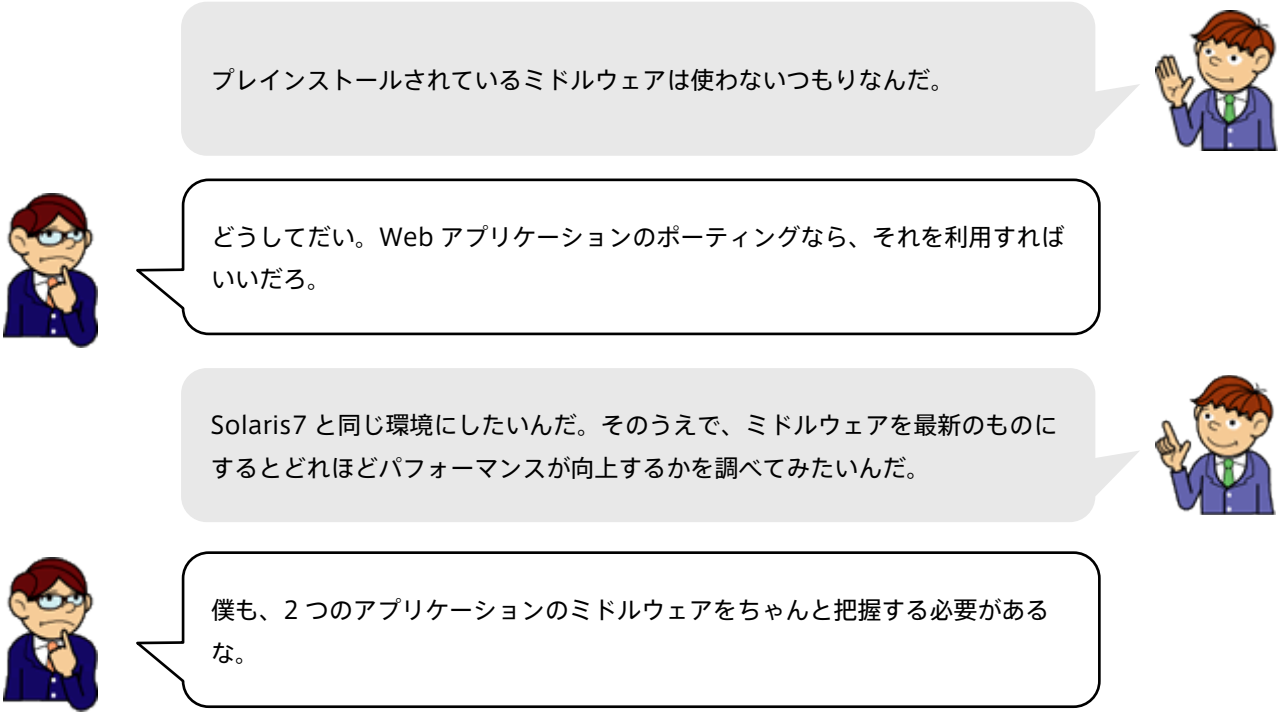
表 1 サーバーのスペック比較

対象となるミドルウェアの特徴

今回のシナリオで新しい Web サーバーを構築するために必要なミドルウェアは HP-UX に標準でバンドルされています。

- HP-UX Apache-based Web Server
- HP-UX Tomcat-based servlet engine
- HP-UX Webmin-based administration
- HP-UX XML Web server tools

これらのミドルウェアは、メモリ管理の強化、chroot を使用したセキュリティの強化、secure SSL のパフォーマンス向上などが施されたものです。



ただし今回の実験では、UNIX マシンである HP-UX サーバーにオープンソースのミドルウェアをあらためてインストールし、Oracle9i を走らせたうえで、既存の Web アプリケーションを動かすことにします。それでは、今回ポーティングする 2 つの Web アプリケーションに必要なミドルウェアを見てみましょう。

「Web セミナー受付システム」のミドルウェア

今回の実験では、Solaris7 上でこの Web アプリケーションを動かすために使用していたのと同じ種類のミドルウェアを HP-UX 上にインストールします。ただし、Apache は 1.3 系ではなく、マルチプロセス・マルチスレッド設計の Apache 2.0.49 を選択しました。また、Servlet コンテナには、ApacheJServ ではなく、現在 Servlet コンテナの主流となっている Tomcat を利用します。そのため、Apache とのコネクタとして JK2 のインストールも必要となります。JK2 での接続方法は、一般的な Socket 接続としました。また、Oracle のバージョンは、対応 OS から検討し、Oracle9i Release2 としました(表 2)。

	Solaris	HP-UX
Web サーバー	Apache 1.3.9 Solaris	Apache 2.0.49 HP-UX
Servlet コンテナ	ApacheJServ-1.0fc1	Tomcat4.1.30
コネクタ		JK2-2.0.4
DB	Oracle8i(8.1.7.0.0)	Oracle9i Release2 (9.2.0.2.0)
JDBC ドライバ	Oracle JDBC Driver 8.1.7.0.0	Oracle JDBC Driver 9.2.0
J2SE	J2SDK 1.2.1_02a	SDK & RTE 1.4 Itanium 版(1.4.2.03)

表 2 「Web セミナー受付システム」のミドルウェア一覧

「会員管理・メール配信システム」のミドルウェア

Apache のバージョンについては、「Web セミナー受付システム」と同様です。PHP については、Solaris7 上ではページごとに PHP3 と PHP4 が混在していましたが、Apache 2.0.49 に対応しているのは PHP4 だけなので、すべてのページに PHP4 で対応するようにしました。また、PostgreSQL はポーティング作業時点(2004 年 5 月)で最新のものを用います(表 3)。

	Solaris	HP-UX
Web サーバー	Apache 1.3.9 Solaris	Apache 2.0.49 HP-UX
PHP モジュール	PHP 3.0.18 および 4.1.2	PHP 4.3.5
DB	PostgreSQL 7.1.3	PostgreSQL 7.4.2

表 3 「会員管理・メール配信システム」のミドルウェア一覧

ポーティング手順の整理

今回のポーティングの担当者 S は、実際の作業に入る前に、ポーティングの手順を次のように整理しました。

1. **サーバー構築**
   HP-UX のインストールおよびネットワーク設定やパッチの適用などを行う。

2. **「Web セミナー受付システム」に必要なミドルウェアのインストール**
   Apache、Servlet コンテナ、Java SDK、Oracle のインストールおよび動作に必要な設定を行う。

3. **「Web セミナー受付システム」のインストール**
   Web アプリケーションのインストールおよび設定を行う。
   アプリケーションの動作に必要なミドルウェアの設定も適宜行う。

4. **「会員管理・メール配信システム」に必要なミドルウェアのインストール**
   PHP、PostgreSQL のインストールおよび動作に必要な設定を行う。
   Apache は「Web セミナー受付システム」でインストールしたものを利用するため、ここではインストールしない。

5. **「会員管理・メール配信システム」のインストール**
   Web アプリケーションのインストールおよび設定を行う。アプリケーションの動作に必要なミドルウェアの設定も適宜行う。

6. **サーバー運用設定**
   実際に Web アプリケーションを稼働させてからの問題点を修正する。

7. **性能測定**
   マイクロソフト社の Web 性能測定ツールを利用して性能測定を行う。

手順はだいたいこんなものだろう。
まずはサーバー構築からだ。

サーバー構築は、OS のインストールとネットワーク設定、それから OS へパッチを当てる必要がある。

詳細は、このインストールガイドどおりにすればいいんだな。

そのとおり。このガイドには既知の問題についても細かく載っている。ミドルウェアのインストールの際にも役立つよ。

それでは、ポーティングの具体的な手順を見ていきましょう。

担当者 S は、Integrity サーバーに付属している『HP-UX 11i バージョン 2 インストール/ アップデートガイド』の手順に従って、HP-UX 11i v2 をインストールし始めました。サーバー構築は、ガイドの手順どおりに行えば何の問題もありません。ここでは、Solaris7 との違いを Tips として示すのみとします。

1 日目：サーバー構築(1)

まずサーバーに HP-UX をインストール(図 1)し、ネットワークを含めた環境を構築しました。

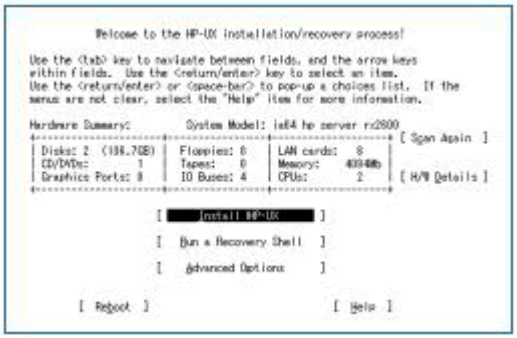


図 1：HP-UX インストール初期画面

Tips1 HP-UX では、ifconfig の引数に-a パラメータは使用不可能

HP-UX では、ifconfig の引数にインターフェース名を指定しなければなりません。そのため、まず、インターフェース名を調べる必要があります。インターフェース名を調べるには、landiag や lanscan-v コマンドを使用します。

```
#landiag
Enter command: lan
Enter command: display
(省略)
Description = lan0 Intel PCI Pro
10/100Tx Server Adapter
[10BASE-TX,HD,AUTO,T
(省略)
```

2 日目：サーバー構築(2)

Tips2 HP-UX と Solaris7 とのネットワーク設定ファイルの違い

Solaris7 では IP アドレスやサブネットマスクなどを別々のファイルで設定しますが、HP-UX では 1 つのファイルを使用します。

	Solaris	HP-UX
IP アドレス	/etc/hosts	/etc/rc.config.d/ netconf
サブネットマスク	/etc/netmasks	

ホスト名	/etc/hostname
デフォルトゲートウェイ	/etc/defaultrouter
インターフェース名	/etc/hostname.<インターフェース名>
hosts	/etc/hosts
DNS 設定	/etc/resolv.conf

表 4 ネットワーク設定ファイルの違い

サーバー構築が完了し、これから本格的なポーティングに入ります。

Linux へのポーティング
サーバー構築編

今回の実験終了後、RedHat Linux AdvancedServer2.1 for Itanium を Integrity サーバーへインストールし、同じような実験をしてみた。OS のインストールは、RedHat Linux のインストール画面の指示どおりでとくに問題なく終了した。

- **ネットワーク設定**

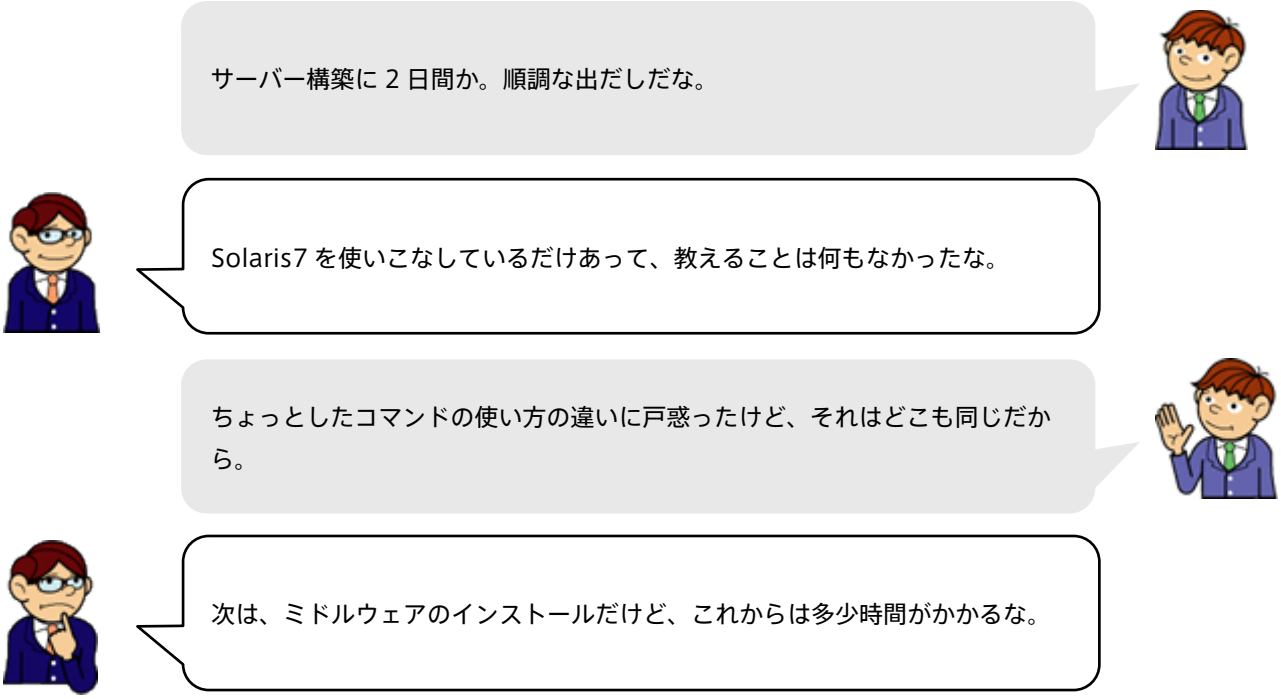
ネットワーク設定は、OS インストール時にすべて完了するため、あらためての作業は発生しなかった。Solaris7 と RedHat Linux のネットワーク設定ファイルの違いを、表 5 にまとめる。

- **必要なパッチの適用について**

RedHat Linux でも、OS にパッチを適用する必要がある。RedHat Linux では、RedHat Network を使用することになるが、これにはアカウント登録が必要である。

	Solaris	RedHat Linux
インターフェース名	/etc/hostname<インターフェース名>	/etc/sysconfig/network-scripts/ifcfg-eth<番号>
IP アドレス	/etc/hosts	
サブネットマスク	/etc/netmasks	
ホスト名	/etc/hostname	/etc/sysconfig/network
デフォルトゲートウェイ	/etc/defaultrouter	
hosts	/etc/hosts	
DNS 設定	/etc/resolv.conf	

表 5 Solaris7 と RedHat Linux のネットワーク設定ファイルの違い



第 3 回

実践！＜Solaris→HP-UX/Linux＞アプリケーション移行ガイド　Part.2

2004 年 11 月　テクニカルライター 大戸 英樹

前回は、「Web セミナー受付システム」と「会員管理・メール配信システム」という 2 つの Web アプリケーションに対するポーティングのシナリオを概観しました。また Integrity サーバーに HP-UX 11i v2 をインストールし、ネットワーク等の環境構築を進めてきました。実践編 Part.2 の今回は、ミドルウェアのインストールを行います。Apache Web サーバーをはじめ、Tomcat と JK2、そして Oracle を HP-UX 上でインストールする手順を見ていきます。

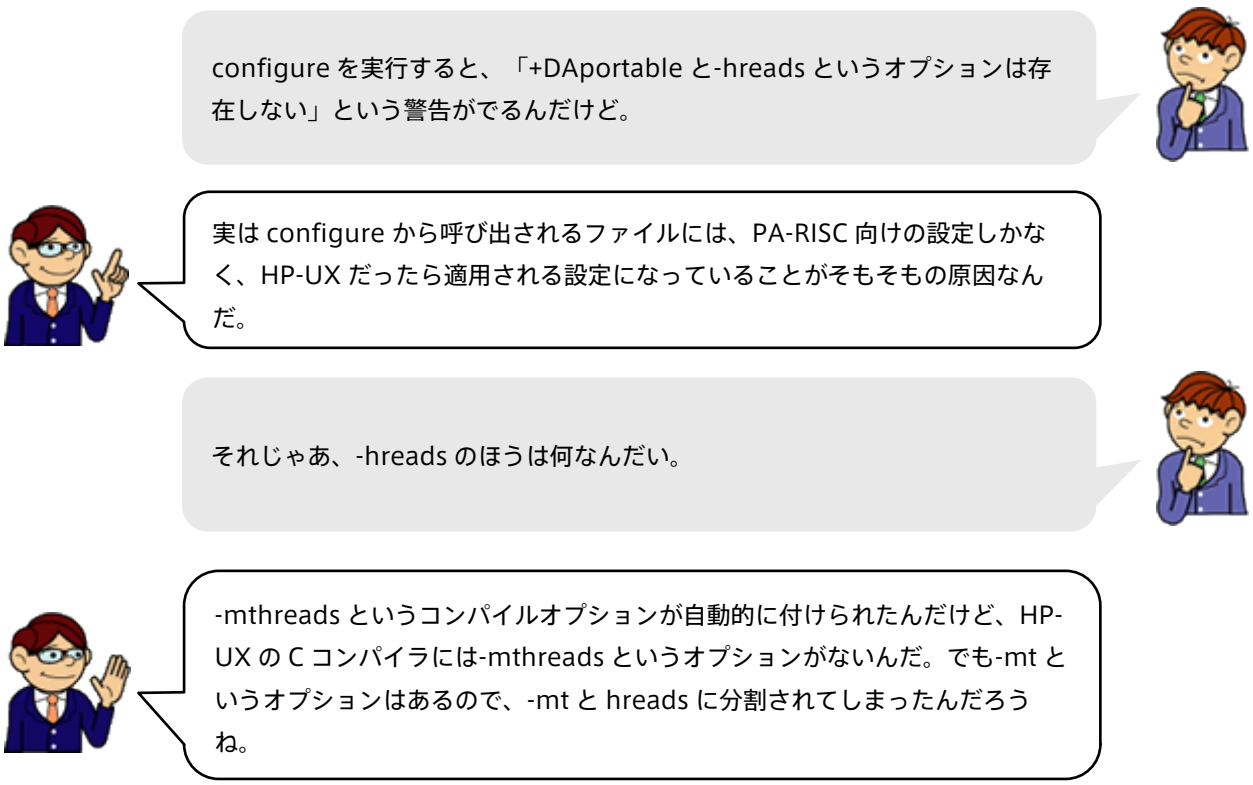
第 2 段階：「Web セミナー受付システム」のミドルウェアのインストール

3〜4 日目： ミドルウェアの設定(1)

まず Web サーバーApache と J2SE をインストールします。

Apache のインストール

Apache は HP-UX にプリインストールされていますが、今回は新しいバージョンの Apache をソースからビルドすることにしました。ソースアーカイブをダウンロードして configure を実行すると、「+DAportable と-hreads というオプションは存在しない」というエラーが出ました(ログ-1)。



この問題を解決するには、コンパイルオプションを Tips3 のように変更します。

```
configure:3434: cc +DSitanium2 + -Ae +Z +DAportable -mthreads -DHPUX11 -D_REENTRANT -
D_XOPEN_SOURCE_EXTENDED -L/usr/local/src/httpd-2.0.49/srclib/apr-util/xml/expat/lib conftest.c >&5
cc: warning 901: unknown option: `+DAportable': use +help for online documentation.
cc: warning 901: unknown option: `-hreads': use +help for online documentation.
```

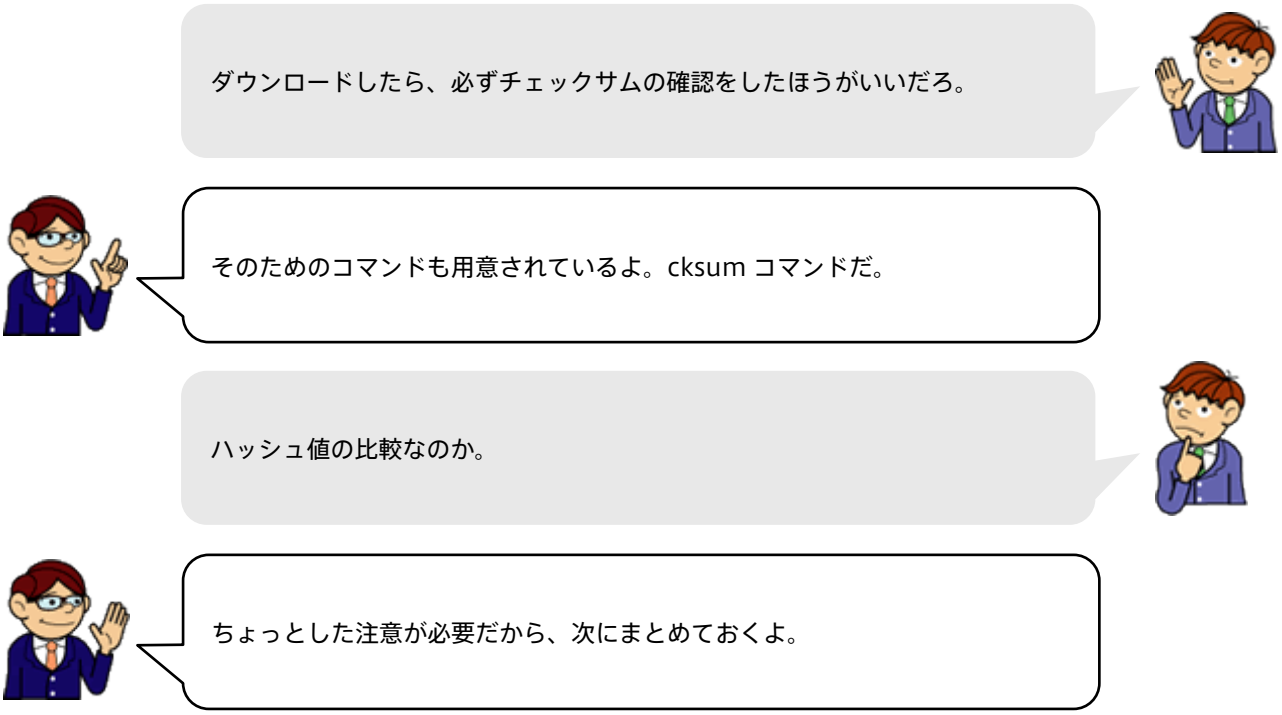
Tips3 HP-UX での configure のオプション
インテル®Itanium®プロセッサ(以下、Itanium)が搭載された Integrity サーバー上でオープンソースのミドルウェアをコンパイルする際には、コンパイルオプションの CC に、-AC99 を追加してください。また、HP-UX には、-mthreads というオプションはありません。コンパイルオプションの LIBS に-lpthread を指定してください。

その後、正しいオプションを付けて、configure を再実行しました。このままインストールは問題なく終了したように見えましたが、調べてみるとシェアードオブジェクトが作成されていませんでした。これは Apache 2.0.49 の configure をはじめとして、HP-UX 11i v2(11.23)への対応がされていないことが原因と思われます。そこで buildconf を実行して config-ure を作り直すことにしました。しかし、buildconf は内部で autoconf と libtool を用いますので、この 2 つを先にインストールする必要があります。さらに、autoconf には m4 が必要です。また libtool は、HP-UX に付属する make を使用してコンパイルすると make check で多数のエラーが出るため、GNU Make をインストールする必要がありました。

最終的に、GNU make、GNU m4、GNU auto-conf、GNU libtool、Apache の順にインストールを行いました。

J2SE のインストール

HP-UX に付属している J2SE のバージョンが 1.4.1 であったため、最新バージョンの 1.4.2 を日本ヒューレット・パッカードの Web サイトからダウンロードしました。チェックサムを確認した後、root ユーザーになり swinstall を実行しました。



Tips4 HP-UX の cksum コマンド
HP-UX の cksum コマンドではチェックサムを調べることができます。このアルゴリズムは 32bit Cyclic Redundancy Check を採用しているため、MD5 チェックサムと直接比較することはできません。また、swinstall で depot のパスを指定する際は、相対パスでなく絶対パスで行う必要があります。

5〜6 日目： ミドルウェアの設定(2)

Tomcat のインストール

担当者 S が Tomcat4.1.30 のソースアーカイブをダウンロードし、HP-UX 付属の tar で展開したところ、展開が失敗して、@LongLink という名前のファイルが複数できました。これは HP-UX(Solaris7 も同様) に付属している tar が長いファイル名に対応していないからです。この問題を回避するために、GNU tar をインストールしてから再度展開しました。

また、ビルドを行うために Ant をインストールしました。BUILDING.txt によると、Tomcat4.0 は Ant のバージョン 1.5 を要求しています。そこで Apache Ant Project の Web サイトのアーカイブから apache-ant-1.5.4-bin.zip をダウンロードし、jar コマンドを使用して展開しました。Ant 以外のアーカイブをすべて同じディレクトリに置いた後、build.properties.sample を build.properties という名前でコピーして編集し、ビルドを行いました。その後、ant dist ターゲットを実行して完全なパッ

ケージを作成し、root ユーザーで dist ディレクトリをインストールディレクトリにコピーして、Tomcat のインストールを終えました(図 1)。

図 1：Tomcat の初期画面

JK2 のインストール

次は、Apache と Tomcat を連携させるためのコネクタをインストールします。今回は、jakarta-tom-cat-connectors-4.1.30-src ではなく、別途 JK2 2.0.4 (jakarta-tomcat-connectors-jk2-src-current.tar .gz)のソースアーカイブを Jakarta の Web サイトからダウンロードしました。しかし、configure が Itanium 用の HP-UX 11i v2(11.23)には対応していなかったので(PA-RISC 用の HP-UX には対応済み)、Tomcat の Web サイトからパッチを入手してビルドしました。
Apache の apxs により、mod_jk2.so ファイルは modules ディレクトリにコピーされるため、あとは httpd.conf に「LoadModule jk2_module mod-ules/mod_jk2.so」を追加して JK2 のインストールを終えました。

Oracle のインストール

Oracle9i Release2 のインストールの前に、ソフトウェアとデータベースのマウントポイント用論理ボリュームを追加しなければなりません。

HP-UX の論理ボリュームってなんだい。

HP-UX ではディスクスペースを論理ボリュームという単位で分けて、論理ボリュームマネージャが管理するんだよ。

それでいったい、どうやって論理ボリュームを使えばいいんだ。

論理ボリュームは、ソフト RAID に似てるんだ。こんなふうに使えばいいんだよ。

Tips5 HP-UX の論理ボリュームの設定方法

論理ボリュームを使うためには、物理ボリュームデバイス上にあるパーティション(/dev/dsk/c0t0d0 など)をボリュームグループにアサインします。このボリュームグループ内の合計スペースを 1 つまたは複数の論理ボリュームとして分割することにより、OS がこれらの領域をパーティションのように利用できるようになります。

サーバー構築時にすでに 43GB を使っています。残りのディスク領域を 10GB ずつに分け、ソフトウェアマウントポイントとデータベースマウントポイント用に割り当てます。論理ボリュームの用意ができたら、Oracle のインストールに進みます。
Oracle のインストーラを起動する前に、次の作業を済ませておきます。

- **UNIX グループとアカウントの追加**
  UNIX グループに oinstall と dba を追加し、追加したグループの UNIX アカウントとして oracle ユーザーを追加します。技術資料『HP-UX 11i v2(11.23) ＋Oracle9i DB 構築手順』には、グループ属性を設定するようにという記述があります。グループ属性を設定することにより、Oracl9i のパフォーマンスが向上します。
- **カーネルパラメータの変更**
  カーネルパラメータを変更するには、SAM から Kernel Config(kcweb)というツールを起動します。kcweb ではパラメータどうしの関連性もチェックされており、関連する項目の値を先に変更しておかなければ変更できない値もあります。
- **CD-ROM デバイスのマウント**
  『Oracle9i for UNIX Systems インストレーションガイドリリース 2(9.2.0.1.0)』には、CD-ROM のマウントには pfs_mount コマンドを使うと書かれていますが、マウントポイントを/SD_CDROM/ とすると、アンマウントに失敗することがあります。『HP-UX 11i v2(11.23)＋Oracle9i DB 構築手順』では、mount コマンドを使うように指示されているため、mount コマンドを使って CD-ROM をマウントします。

あとは、Oracle のインストーラを起動して Oracle のドキュメントどおりに行えば、インストールは完了です(ログ-2)。

・ログ-2 Oracle9i 起動時のログ

```
SQL*Plus: Release 9.2.0.2.0 - Production on 月 Jul 12 15:41:14 2004
Copyright (c) 1982, 2002, Oracle
Corporation. All rights reserved.

SQL> アイドル・インスタンスに接続しました。
SQL> ORACLE インスタンスが起動しました。

Total System Global Area 320356472 bytes
Fixed Size 753784 bytes
Variable Size 285212672 bytes
Database Buffers 33554432 bytes
Redo Buffers 835584 bytes
データベースがマウントされました。
データベースがオープンされました。
SQL> Oracle9i Enterprise Edition Release
```

```
9.2.0.2.0 - 64bit Production

Database "ora" warm started.
```

「Web セミナー受付システム」のミドルウェアのインストールは、これですべて完了しました。

Linux へのポーティング
ミドルウェアインストール編「Web セミナー受付システム」

- C コンパイラについて

ミドルウェアのソースのコンパイルをする際に、RedHat Linux に同梱されていた C コンパイラの GCC2.96 ではなく、GCC 3.4.0 をあらためてインストールして利用した。この GCC 3.4.0 は、Itanium 上での高速化が期待されている。

- Oracle9i のインストール

Oracle9i のインストール時に注意しなければならないのは次の点だ。このサーバーは、RedHat Linux のインストールの際に、ハードディスクの容量一杯にパーティションが切ってあったため、HP-UX の時とは異なり 2 台目のハードディスクに Oracle のソフトウェアマウントポイントとデータベースマウントポイントを作成しようと考えた。2 台目のハードディスクはまだ未使用だったため、fdisk を実行し 2 台目のハードディスクを参照すると、全体が 1 つの GPT パーティションに見えた。GPT とは、GUID パーティションテーブルの略で、MBR などのパーティションスタイルの一種なのだが、現時点では fdisk で GPT 上のパーティションテーブルを見ることはできない。そのため、これを削除して MBR でパーティションを切り直してしまった。本来は、parted というコマンドを使って、GPT 上のパーティションを操作することが推奨される。

第 4 回

実践！＜Solaris→HP-UX/Linux＞アプリケーション移行ガイド　Part.3

2004 年 12 月　テクニカルライター 大戸 英樹

前回は、「Web セミナー受付システム」のミドルウェアのインストールを行いました。実践編 Part.3 の今回は、前回に続いて「Web セミナー受付システム」のインストールを行います。さらに、「会員管理・メール配信システム」に必要な Postgre SQL や PHP などのミドルウェアのインストール、そして「会員管理・メール配信システム」をインストールする手順を見ていきます。

第 3 段階：「Web セミナー受付システム」のインストール

ミドルウェアの準備ができたので、次は「Web セミナー受付システム」Web アプリケーションをインストールします（図 1）。

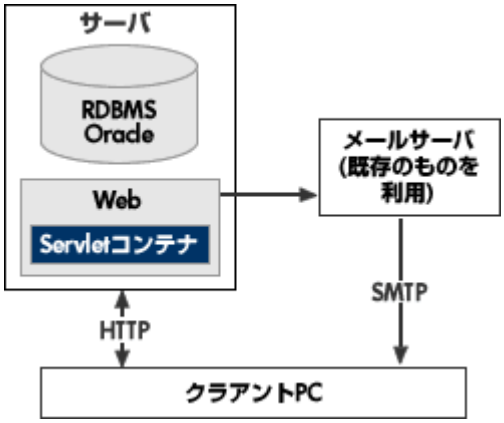


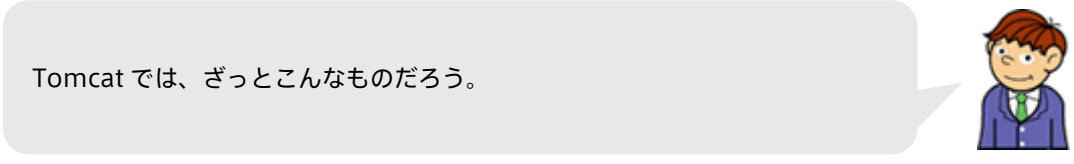
図 1：Solaris7 上の Web セミナー受付システム構成図

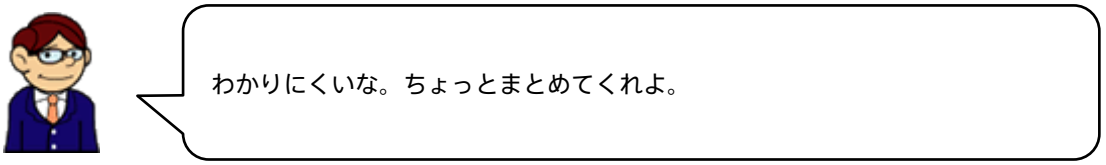
7 日目：「Web セミナー受付システム」の設定

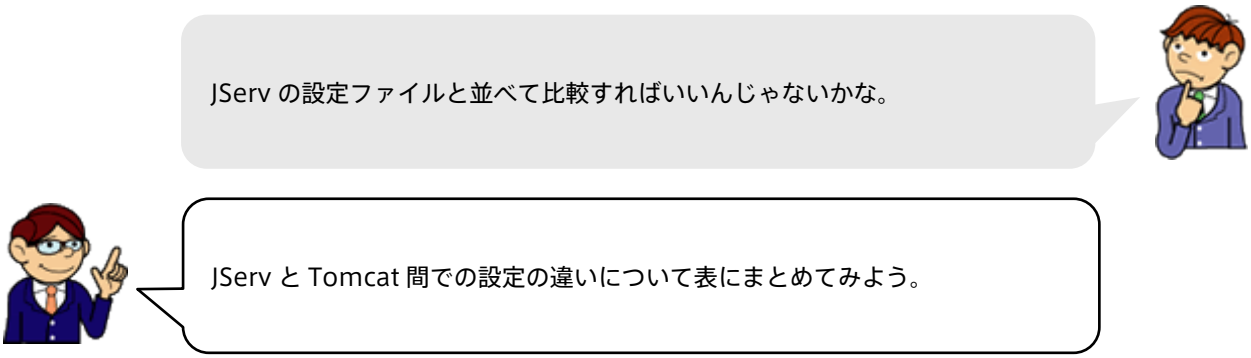
ミドルウェアのインストールが完了しているので、Web アプリケーションは問題なく動くはずです。ここでは、この Web アプリケーションを HP-UX にインストールする際に、Solaris7 にインストールしたときの手順どおりにはいかなかった点だけまとめておきます。

- **ページの文字化け**
  Apache の httpd.conf に AddDefaultCharset が iso-8859-1 と設定されていました。これはデフォルトのコンテンツのキャラクタセットを指定するもので、Apache1.3.12 から使われるようになったものです。必要のない設定のため、その行をコメントアウトしました。

- **Servlet の設定変更**
  Servlet コンテナが JServ1.0fc1 から Tomcat4.1.30 に変わったため、アプリケーションの設定ファイルを書き変える必要があります。Solaris7 では、JServ の設定は、httpd.conf からインクルードされる jserv.conf と、jserv.conf から指定される jserv.properties に記述されています。
  jserv.conf は、Apache から JServ へ転送する URI の指定、jserv.properties ファイルの指定、プロトコルとポートの指定などを行っています。Tomcat では、これらを workers2.properties ファイルや、servlet.xml ファイルに記述しなければなりません。
  jserv.properties は、クラスパスやライブラリの指定、ログの設定、アプリケーションの登録などを行っています。Tomcat では、これらも servlet.xml ファイルに記述します。







JServ と Tomcat 間での設定の違いについて表 1 にまとめました。

- **Oracle の設定**
  Solaris7 上の Oracle8i では永続表領域を割り当てることができましたが、Oracle9i では不可のため、一時表領域とすることにしました。その点を除けば、データベースのインポート／エクスポートには何の問題もありませんでした。
  これで、「Web セミナー受付システム」のインストールは完了です。HP-UX 上のアプリケーション画面（図 1）をご覧ください。

<参考> Linux へのポーティング - Web アプリケーションのインストールについて

Web アプリケーションのインストールは、HPUX へのインストールと、工程・内容ともに何ら違いはなかった。

図 2：Web セミナー受付システムの画面例

表 1：JServ と Tomcat の設定の対比

設定項目	JServ の設定ファイル	Tomcat の設定ファイル
Servlet コンテナへ転送する URI	httpd.conf	workers2.properties

Servlet コンテナの動作設定ファイル	デフォルトでは jserv.properties	server.xml
アプリケーションの登録	jserv.conf 内の ApJServMount 項目で指定	server.xml 内の Context タグで指定
アプリケーションごとの設定	jserv.conf で指定されるプロパティファイル	アプリケーションディレクトリの WEB-INF/web.xml
Apache との通信プロトコルの指定	jserv.conf	server.xml の Connector タグ内で指定

第 4 段階：「会員管理・メール配信システム」のミドルウェアのインストール

次に、2 つ目の Web アプリケーション「会員管理・メール配信システム」に必要なミドルウェアをインストールします。

Apache はすでにインストール済みですので、そのまま利用します。

8〜11 日目： ミドルウェアの設定（1）

今回の実験で最も時間がかかったのが、Postgre SQL のインストールでした。PostgreSQL のサイト（日本語サイトはこちら）にあるソースアーカイブ postgresql-7.4.2.tar.gz をダウンロードし configure しました。その際のエラーをひとつずつ解消していきましょう。

- **readline ライブラリが見つからない**
  HP-UX では readline ライブラリが用意されていません。この場合、HP-UX 用にビルドされたオープンソースのソースコードやバイナリが用意されている http://hpux.connect.org.uk/ から HP-UX 11.23 対応バイナリを入手することも一つの手です。なお、readline ライブラリがないと Postgre SQL の一部のツールで編集機能が制限されますが、特に動作に支障があるわけではないため、今回は readline ライブラリを使用しないことにしました。--without-readline オプションを付けて再度 configure を行います。

- **zlib ライブラリが見つからない**
  readline ライブラリと同様に、HP-UX では zlib ライブラリが標準では用意されていません。こちらも特に動作に支障はないため、今回は使用しないことにしました。--without-zlib オプションを付けて再度 configure を行います。

- **flex と bison のインストール**
  flex と bison が必要だという警告が出ました。この 2 つがインストールされていませんでした。そこでまず bison-1.875 のソースアーカイブを GNU サイト（http://www.gnu.org/）からダウンロードし、展開後、configure を実行しました。すると「GNU m4 が必要です」というエラーが出ました。GNU m4 は「Web セミナー受付システム」のポーティング時にインストールしてあるので、インストールされているディレクトリを PATH に追加し、もう一度 bison-1.875 をインストールしました。続いて、GNU サイトにある flex-2.5.4 をダウンロードしてインストールしました。

- **make 時のエラー**

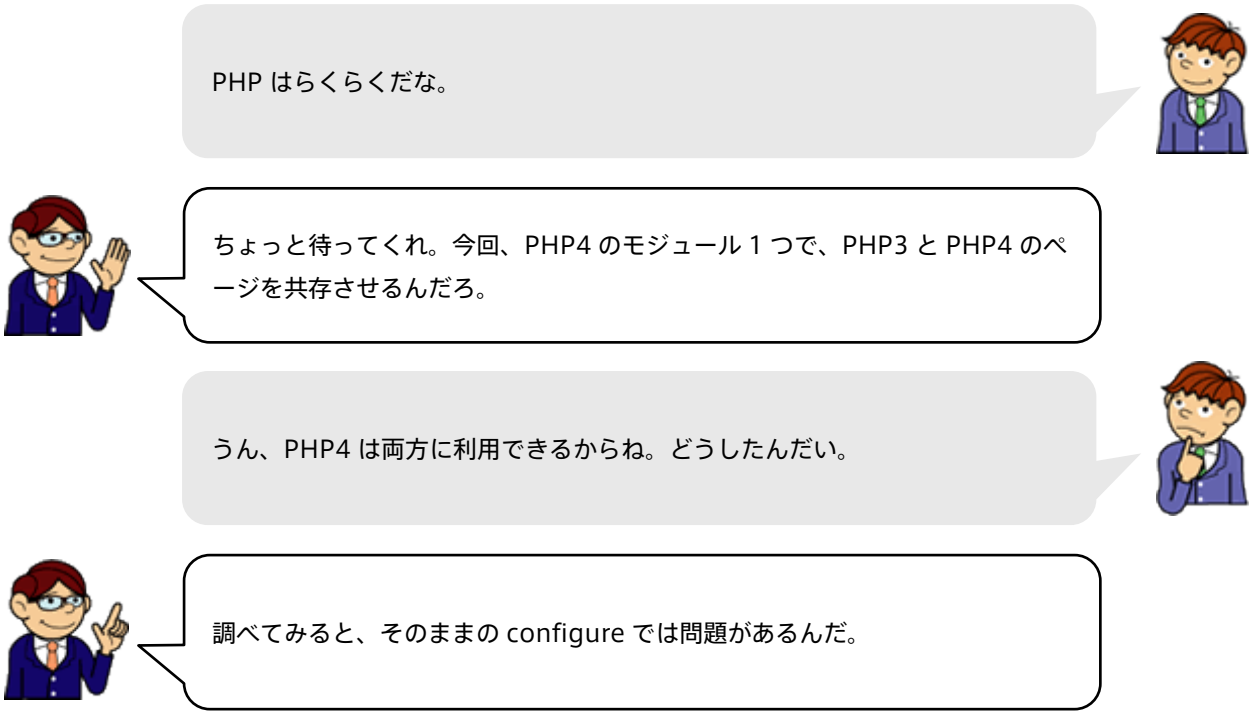
PostgreSQL に付属している configure を実行したところ、問題なく Makefile ができたので、「Web セミナー受付システム」のポーティング時にインストールしてある GNU Make を使って構築を行おうとしたところ、途中でエラーが出てしまいました。
make check によると「float8 のエラーのメッセージが違う」ということでしたが、PostgreSQL のドキュメントを参照し、特に問題はないと判断し、インストールを続行しました。続いて SAM でユーザーpostgres を追加し、データベースの初期化とユーザーpostgres のデータベースを作成しました。psql により対話シェルを起動し、テーブルが正しく作られているかを確認したあと、createuser ＜ユーザー名＞で、データベースオブジェクトの管理者アカウントを追加しました。

12〜14 日目：ミドルウェアの設定（2）

次は PHP をインストールします。PHP のサイトからソースアーカイブ（php-4.3.5.tar.gz）をダウンロードし、gunzip と tar でアーカイブを展開しました。PHP を構築する際には、PostgreSQL 用の DB モジュールの作成を行う必要があるため、あらかじめ PostgreSQL 用のライブラリパスを設定しました。

- **PHP4 の configure 時の注意点**

今回の実験では、PHP4 のモジュールを使って PHP3 と PHP4 のページを共存させるため、config-ure にオプションを付けて実行する必要があります。



<参考> Tips6 PHP3 と PHP4 のページの共存方法

PHP4 のモジュールを使って PHP3 のページと PHP4 のページを共存させる場合は、configure に--enable-versioning を付ける必要があります。

これで Makefile は無事にできましたが、ここから多少変則的なことをしなければなりませんでした。その要点を次にまとめます。

- **PHP のインストール**
  GNU make を実行すると、途中で「リンカが PostgreSQL のライブラリを見つけることができず、そのため PostgreSQL のシェアドライブラリを作ることができない」という警告が出ました。担当者 S はとりあえず、この警告を無視して、PHP のインストールを進め、シェアドライブラリは本体のインストール後に構築することにしました。

- **シェアドライブラリの構築**
  PHP のインストール終了後、シェアドライブラリを構築するために aclocal を実行しましたが、「コマンドが見つかりません」というエラーが出ました。aclocal のインストールがまだでした。GNU automake に aclocal が含まれているので、GNU サイトからダウンロードしてインストールしました。その後、再度 aclocal を実行してシェアドライブラリを構築しました。

- **PHP 設定ファイルの編集**
  PHP の設定ファイルの雛形ファイル（php.ini-dist） の名前を、/usr/local/lib/php.ini に変更して内容を編集し直しました。

- **phpinfo.php ファイルの作成**
  下記の内容の phpinfo.php ファイルを/opt/ apache2/htdocs/ディレクトリに作成します。

```
<?
phpinfo();
?>
```

ここまでの作業を終えたら、http://＜サーバーの IP アドレス＞/phpinfo.php を開いて、PHP の設定情報が表示されることを確認します。これで PHP のインストールは完了です。

<参考> Linux へのポーティング - ミドルウェアインストール編

「会員管理・メール配信システム」

- PostgreSQL について

PostgreSQL のインストールをするための configure の処理ログの中で「インストールされている Bison が古すぎます。PostgreSQL は Ver1.875 以上の bison を必要とします」という警告が出た。GNU サイトから bison-1.875 のソースアーカイブをダウンロードし、インストールした。

- PHP について

PHP のインストールでは、HP-UX 時と同様、--enableversioning をつけて configure した。

第 5 段階：「会員管理・メール配信システム」のインストール

ミドルウェアのインストールが完了したので、次は「会員管理・メール配信システム」Web アプリケーションをインストールします（図 3）。

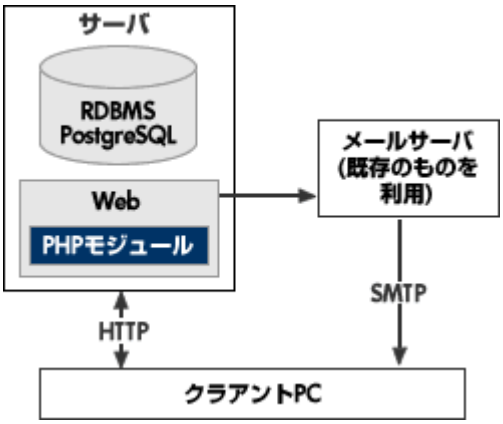


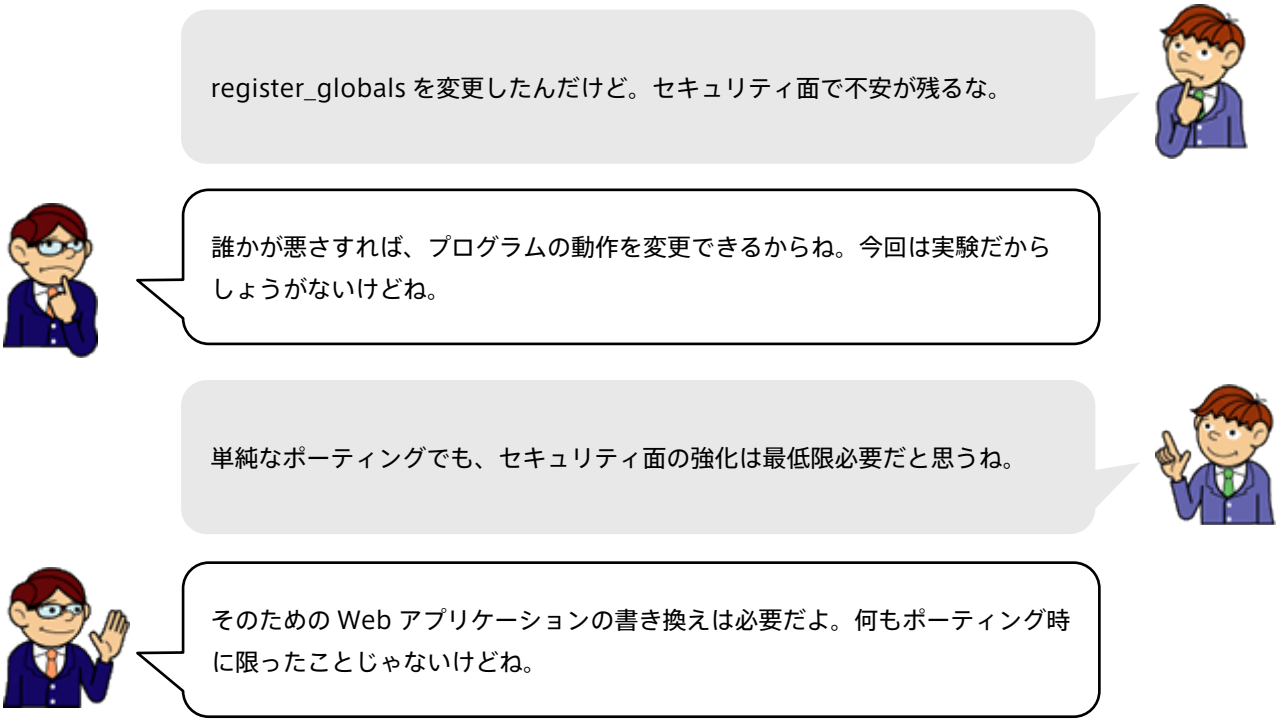
図 3：Solaris 上の会員管理・メール配信システム構成図

15〜17 日目：「会員管理・メール配信システム」の設定

ここでも、調整する必要があった点だけをまとめて記述します。

- **PostgreSQL の datetime 型を使用している PHP の変更**
  datetime という日付時刻型は、新しく用意された timestamp 型に移行していましたが、引き続き利用することができました。しかし PostgreSQL 7.3 になった段階で datetime 型は完全に廃止され、datetime 型を利用しているプログラムは書き換えないとエラーが出るようになっています。この実験では対象プログラムは 700 を超えるため、書き換えプログラムを Perl で作成し、自動的に書き換え処理を行いました。

- **register_globals の変更**
  PHP3 で書かれたプログラムを PHP4 のモジュールでそのまま動作させるためには、register_ globals を変更する必要があります。次の内容を php.ini に追加する必要があります。
  `register_globals = On`



<参考> グローバル変数のデフォルト設定

register_globals を On にすると Environment・GET・POST・Cookie・Server の各パラメータをグローバル変数として登録できるようになります。このディレクティブは PHP3 ではデフォルトで On になっていましたが、グローバル変数が悪用される可能性があるために、PHP4.2.0 以降ではデフォルトで Off になっています。

これで 2 つ目の Web アプリケーションのインストールが完了しました。ポーティング作業はこれで終了です。

<参考> Linux へのポーティング - Web アプリケーションのインストールについて

この Web アプリケーションのインストールも、HP-UX へのインストールと、工程・内容ともに何ら違いはなかった。

図 4：会員管理・メール配信システムの画面例

最終回

実践！＜Solaris→HP-UX/Linux＞アプリケーション移行ガイド　Part.4

2005 年 1 月　テクニカルライター 大戸 英樹

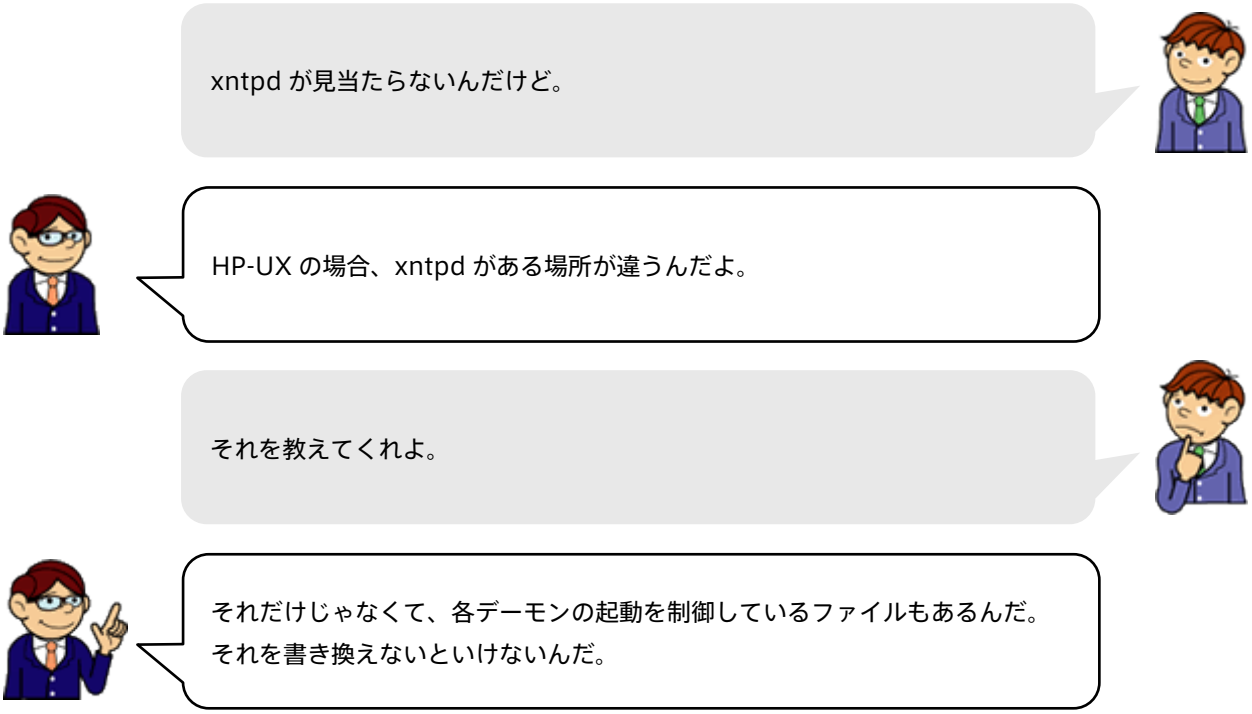
前回までで、「Web セミナー受付システム」と「会員管理・メール配信システム」に必要なミドルウェアおよびシステム本体のインストールは終了しました。最終回となる今回は、サーバーの運用設定に関するヒントを紹介します。また、システムの性能テストを行い、今回のポーティング全体について総括します。

サーバー運用設定

ポーティング作業は終わりましたが、実際に運用するときには、各デーモンの設定やアクセス制限をする必要があります。ここでは、サーバーの運用設定に関するヒントをまとめて紹介します。

NTP 設定

正確な時刻を表示するため、NTP の設定を行います。NTP サーバーの設定を行うには、/etc/ntp.conf を編集しなければなりません。このファイルの編集後、/sbin/ディレクトリにあるデーモン xntpd を起動します。



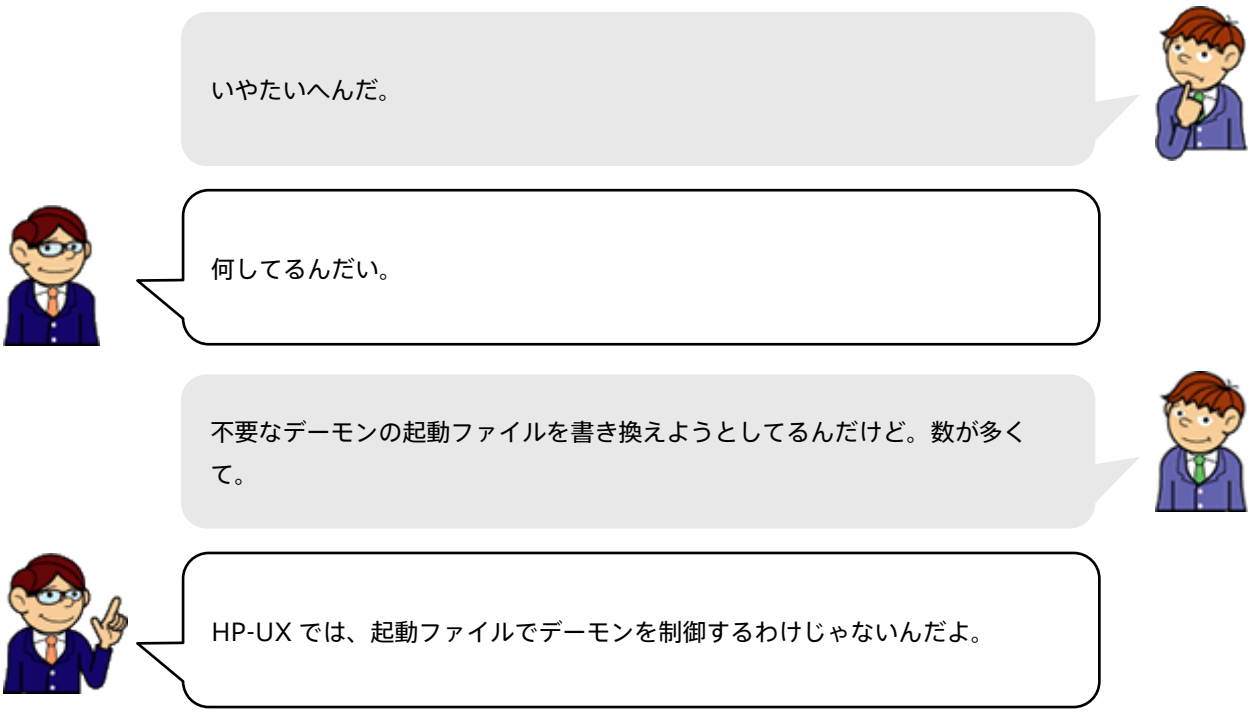
Tips8 起動ファイルの場所と環境変数スクリプト

HP-UX では、起動ファイルは/etc/ディレクトリではなく/sbin/ディレクトリに存在します。

/sbin/init.d・・・・・・・・・・・・・実行スクリプト
/sbin/rc*.d（*は 0～4 の数字）・・・・実行レベル別のリンクファイル
/sbin/rc*.d/ディレクトリのファイルは/sbin/
init.d/ディレクトリの実行スクリプトにシンボリックリンクが設定されています。
また、HP-UX では、/sbin/init.d/ディレクトリの起動スクリプトから/etc/rc.config.d/ディレクトリの環境変数スクリプトを読み出しています。この/etc/rc.config.d/ディレクトリの環境変数スクリプトの中に記述されているパラメータによりデーモン起動の制御を行っています。このパラメータがデーモンを起動させないように記述されていると、/sbin/init.d/ディレクトリの起動スクリプトを実行してもデーモンは立ち上がりません。

不要なデーモンを止める

HP-UX ではデフォルトで様々なデーモンが起動するように設定されています。セキュリティホールやリソースの無駄使いなどの理由から、不要なデーモンを止める必要があります。HP-UX では、起動時に動作するデーモンを/etc/rc.config.d/ディレクトリ内の環境変数スクリプトで制御します。



Tips9 不要なデーモンを止める場合は

HP-UX では、/sbin/rc*.d/（*は 0～4 の数字）ディレクトリの起動ファイルで起動時に動作するデーモンを制御するのではなく、/etc/rc.config.d/ディレクトリにある環境変数スクリプトで制御します。実際には、/sbin/rc*.d/ディレクトリの実行ファイル（/sbin/init.d/へリンクされている）から/etc/rc.config.d/ディレクトリの環境変数スクリプトが読み込まれ、その中にある Start 用パラメータの数値によりデーモンを起動するか判断します。Solaris のように rc.*d/ディレクトリにファイルがあるからと言って、起動時にデーモンが立ち上がるとは限りません。

サーバーの状態監視

サーバーを適切に運用するためには、サーバーのディスク容量と適用済みのパッチ情報を確認する必要があります。

Tips10 サーバーのディスク容量の確認

Solaris7 では、サーバーのディスク容量を確認するためには、df -k を実行しますが、HP-UX では、bdf コマンドを使用します。Solaris7 では、サーバーのディスク容量を確認するためには、df -k を実行しますが、HP-UX では、bdf コマンドを使用します。

Tips11 適用済みのパッチ情報の確認

HP-UX で適用済みのパッチ情報を確認するときは、/usr/sbin/swlist -l patch *.*,c=patch を実行します。

セキュリティの確保

最低限のセキュリティを確保するために、サーバー上でアクセス制限を行う必要があります。

Tips12 アクセス制限

Solaris7 では、TCPWrapper によって FTP のアクセス制限を行いますが、HP-UX では TCPWrapper をインストールする必要はありません。デフォルトで実装されている inetd.sec の機能を使用します。設定を行うには、/var/adm/inetd.sec を編集し、次のような形式の項目を記述します。
[Service] [allow|deny] [address|network] [address|network] ・・・・

たとえば、192.168.1.0/24 のネットワークと 10.10.10.10 のホストからの FTP アクセスを拒否するには、次のように記述します。

ftp deny 192.168.1.* 10.10.10.10

ファイルを保存するとすぐに設定が反映されます。

Tips13 root ログインの制限

HP-UX では、/etc/securetty ファイルで root ログインを制御しています。このファイルはデフォルトでは存在しないので、自分で作成することになります。このファイルに console と記述すると、ネットワーク経由での root ログインが不可になります。

Linux へのポーティング サーバー運用設定編

■NTP の設定について

RedHat Linux で、NTP サーバーの設定を行うために、HP-UX と同じように、/etc/ntp.conf の内容を編集した。ただし、編集した設定を反映するためには、etc/init.d/ntpd restart を実行しなければならない。NTP サーバーと同期が取れているかどうかの確認は、ntpq -p を実行する。

■standalone 型デーモンの停止方法

RedHat Linux では、chkconfig というコマンドで、デーモンの起動設定を行った。chkconfig --list で、各デーモンがどのランレベルで起動するかを確認してから、次のように設定すればよいということだった。
chkconfig -- level < ランレベル> < デーモン名><on | off>

実際に、Canna をランレベル 2〜5 の時に起動させないようにした。

/sbin/chkconfig --level 2345 canna off

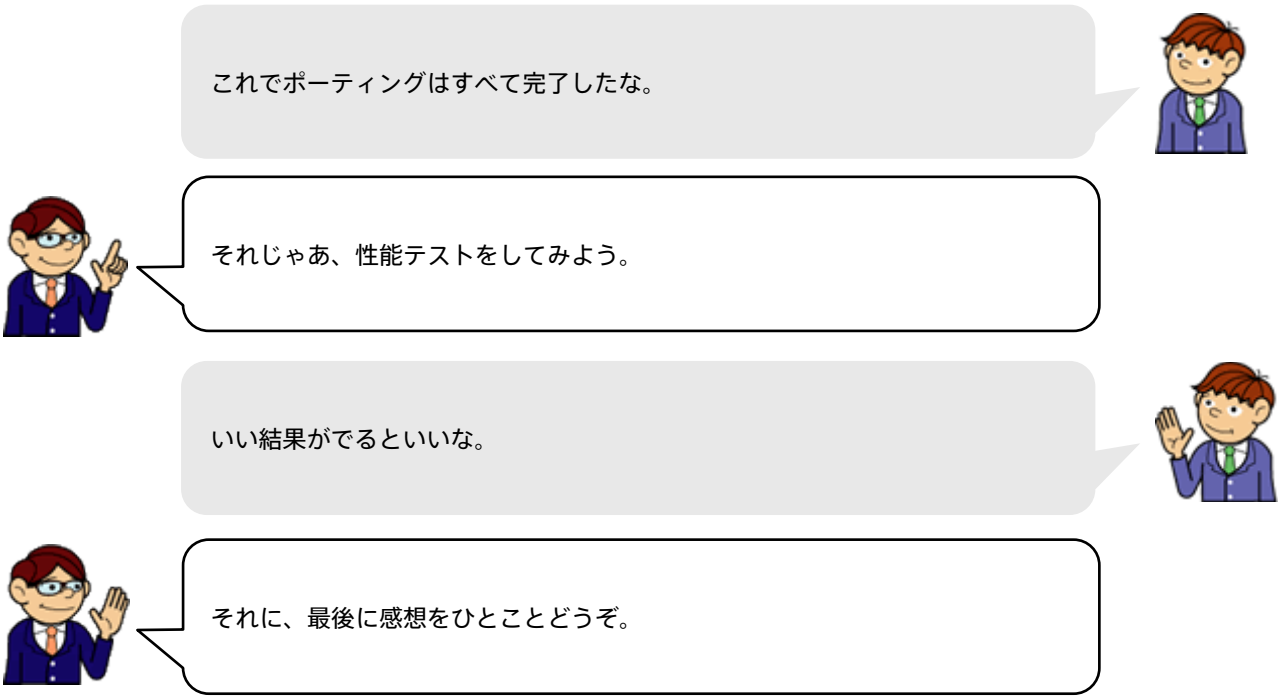
■inetd で制御するデーモンの停止方法

RedHat Linux では、inetd ではなく xinetd を使う。/etc/xinetd.d/ディレクトリにデーモンごとのファイルがある。このファイルを編集して、disable=yes とすればデーモンは起動しない。デフォルトでは、すべて disable=yes なので、逆に、起動したいデーモンのファイルだけ disable=no と書き換えた。

■アクセス制限について

RedHat Linux でアクセス制限をする場合は、/etc/xinetd.d/ディレクトリにあるデーモンごとのファイルに、「only_from = < IP アドレス>という記述を加える。設定反映のために、/etc/init.d/xinetdrestart を起動した。

まとめ



ポーティングにかかった日数

ミドルウェアを新しくしたことで、機種には依存しない問題が起きました。今回のポーティングにかかった日数は、表 7 のとおりです。ミドルウェアのインストールに時間がかかったのは前述した理由によります。

作業内容	日数
サーバー構築	2 日
「Web セミナー受付システム」ミドルウェアのインストール	4 日
「Web セミナー受付システム」のインストール	1 日
「会員管理・メール配信システム」ミドルウェアのインストール	7 日
「会員管理・メール配信システム」のインストール	3 日

表 7 それぞれの作業ごとの日数

性能測定

今回の性能測定では、マイクロソフト社の「MS Web Application Stress Tool」を利用し、秒間あたりのアクセス数およびコンテンツの取得に要した時間を計測しました。テスト方法は 30 個のスレッドを使用して、コンテンツを取得し続けるというものです。それぞれのテスト内容は表 8 のとおりです。

テスト結果は表 9 のとおりです。1 行目の req/sec は秒当たりのアクセス数、2 行目の ms はコンテンツのデータを取得し始めた時間を指します。秒当たりのアクセス数は大きいほど良好な結果であることを意味し、コンテンツデータを取得し始めた時間は小さいほど良好な結果を意味します。

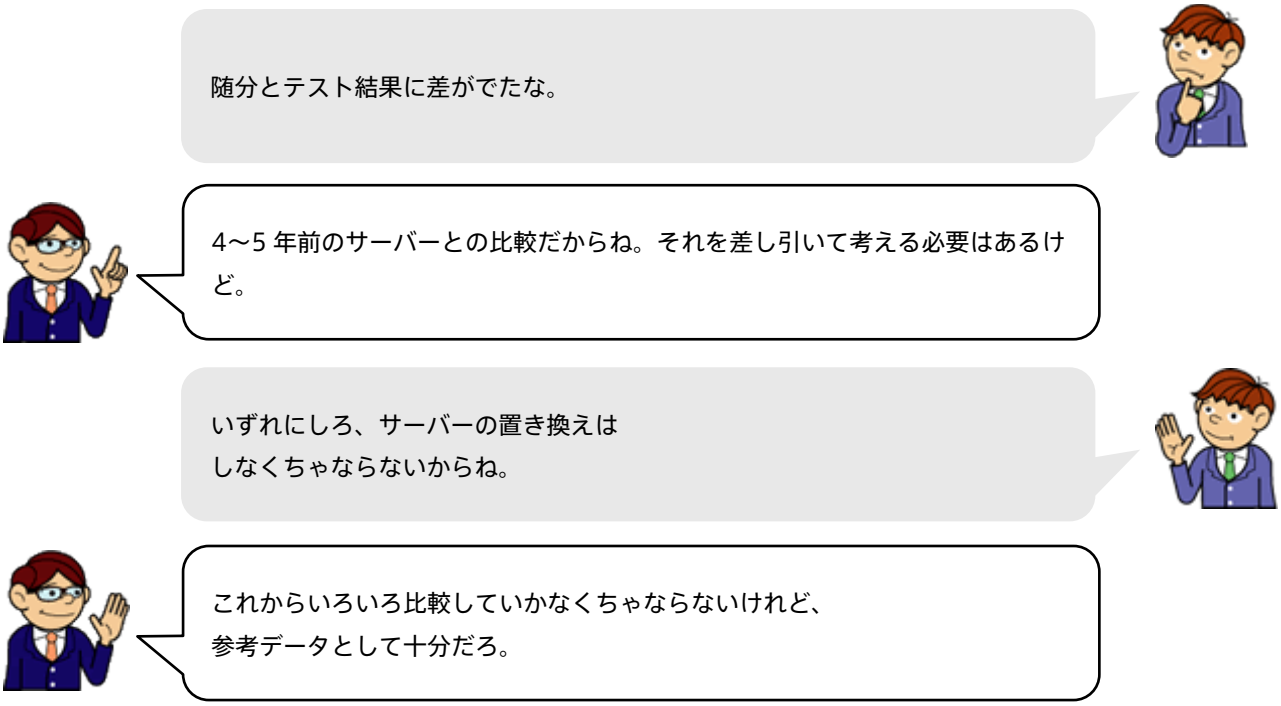
秒あたりのアクセス数の性能向上率は 3.5 倍から 34 倍となり、非常に大きなパフォーマンス向上であることが確認できました。

テスト番号	テスト対象
テスト 1	静的な HTML コンテンツ
テスト 2	決められた文字列を出力する PHP のコンテンツ
テスト 3	データベース（PostgreSQL）に対して select 文を実行し、その結果を表示する PHP のコンテンツ
テスト 4	決められた文字列を出力する Servlet のコンテンツ
テスト 5	データベース（Oracle）に対して select 文を実行し、その結果を表示する Servlet のコンテンツ

表 8 性能測定テストの内容

テスト番号	Solaris	HP-UX
テスト 1	520.80 req/sec 56.30 ms	1854.02 req/sec 14.27 ms
テスト 2	260.28 req/sec 113.93 ms	3005.67 req/sec 4.08 ms
テスト 3	117.77 req/sec 252.92 ms	1687.47 req/sec 16.12 ms
テスト 4	2.85 req/sec 6,494.7 ms	97.04 req/sec 27.14 ms
テスト 5	3.78 req/sec 5,935.2 ms	89.59 req/sec 62.32 ms

表 9 性能測定テストの結果



担当者の感想

今回のポーティング作業を行った担当者 S は、HP-UX サーバーを扱うのが初めてだったため、HP-UX 特有のディレクトリ構造やコマンドオプションという点で少々つまずきましたが、同僚 H のアドバイスのおかげで無事に 2 種類の Web アプリケーションを移行することができました。HP-UX のインストール自体は、Solaris を扱ったことのあるエンジニアならば問題なく行うことができるでしょう。今回の実験は新しい CPU と OS との組み合わせということで、オープンソースのミドルウェアのコンパイルに手間がかりましたが、今後は Web の情報も増えていくと予想されますし、今回行ったような作業もパッチとして提供されると思われます。Oracle の利用に関しては、ドキュメントが充実していることもあり、特に問題になるような点はありませんでした。性能測定テストの結果を見れば HP-UX サーバーの優位は実感できるので、パフォーマンス面を重視するならば HP-UX サーバーは有力な選択肢と言えるでしょう。

Linux へのポーティング RedHat Linux へのインストールの総括

RedHat Linux での測定結果は、表 10 のとおりであった。
静的コンテンツを扱ったテスト 1、PHP を扱ったテスト 2 および 3 の結果を見る限り、HP-UX の C コンパイラ(aCC)は、RedHat Linux の C コンパイラ(gcc)よりも、最適化において優秀なことがわかった。Java を扱ったテスト 4 およびテスト 5 の結果では、RedHat Linux の Java の方が、HP-UX の Java よりも良好な結果が出た。この結果には理由がある。RedHat Linux の J2SE は、BEA 社の WebLogic JRockit 1.4.2SDK for Linux(64bit)をインストールしたが、この JRockit は、ポーティング作業時点（2004 年 5 月）で、フリーのものでは最も高度な最適化を行う SDK だろうと思われる。HP-UX と BEAWebLogic Server 8.1 との組み合わせで測定していたら、結果はまた違ったものになっていただろう。

テスト番号	RedHat Linux
テスト 1	1407.25 req/sec 19.52 ms
テスト 2	2189.68 req/sec 11.24 ms
テスト 3	1426.88 req/sec 18.87 ms
テスト 4	154.19sec 22.49 ms
テスト 5	123.87 req/sec 93.47 ms

表 10 RedHat Linux の測定結果

HP-UX

www.hpe.com/jp/hpux

Hewlett Packard Enterprise