

## コストメリットだけの導入は危険

PART

3

サービスレベルを意識した  
オープンソースDBの正しい適用法

企業システムにおいて、オープンソースプロダクトが存在感を増してきている。古くから普及している Web サーバーやメールサーバーなどは言うに及ばず、グループウェアやCRMなどの業務系アプリケーションでも勢力を拡大している。データベースの世界も例外ではない。今まで漠然とした棲み分けはあったが、ここへ来て

混沌とした状況になってきた。そのため、システムへの採用を検討する際に、いろいろと迷うケースも増えていることだろう。本稿では、商用DBとオープンソースDBを採用する際の判断基準について検討し、そこから見えてくるオープンソースDBの真のメリットを探ってみたい。

株式会社ビーブレイクシステムズ 上川伸彦 KAMIKAWA, Nobuhiko 鷲野秀次 WASHINO, Hidetsugu

## 無償版商用DBの台頭

ご存知のように、現在では商用DBのマーケットは、Oracle Database (日本オラクル)、SQL Server (マイクロソフト)、DB2 (日本アイ・ビー・エム) の3強が分け合う構図となっている。近年になって、これらのベンダ各社は、それぞれDB製品の無償版 (以下、無償版DB) を相次いでリリースした (表)。

各ベンダの無償版DBは、有償版の製品と比べて機能制限こそあるものの、開発専用というわけでもなく、システムの特性によっては (いろいろと議論の余地はあるが、

以下、「ローエンド」と呼ぶことにする)、十分運用に耐え得るスペックとなっている。

当然、各ベンダの意図としては、無償版から有償版へと容易にアップグレードできることから分かる通り、ユーザーがこれら無償版をエンタープライズ向けに導入し、将来的にハイエンド向けの有償版へとアップグレードしていくことを期待しているのだから、このような無償版提供の方向性は、ユーザーサイドから見ても商用DBを採用する敷居を下げるので、そのメリットは大きいと言えそうだ。

しかし、無償版DBをリリースしたベン

ダ各社の意図は「有償版への呼び水」だけののだろうか。そうではないと筆者は考える。躍進するオープンソースDBに対抗し、ユーザーや開発者の関心をオープンソースDBから商用DB (当然、自社製品) に呼び戻すという目的もあるはずである。

無償版DBが狙うような、ローエンドシステムを対象とするマーケットは確実に存在する。また、ローエンドだけにコスト低減への圧力も強いはずだ。これは、現在ローエンドシステムにオープンソースDBを採用する例が増えているという現象に如実に現われている。このオープンソースDBの存在感が大きいマーケットにおいて、ベンダ各社はオープンソースDBに対抗す

表：主要DBベンダの無償版DB

	Oracle Database 10g Express Edition	Microsoft SQL Server 2005 Express Edition	DB2 Universal Database Express-C
最大DBサイズ	4GB	4GB	無制限
最大メモリサイズ	1GB	1GB	4GB
最大CPU数	1CPU	1CPU	2CPU
稼働OS	Windows/Linux	Windows	Windows/Linux
提供開始時期	2006年2月	2005年12月	2006年1月
その他機能制限	<ul style="list-style-type: none"> <li>以下の機能については一部提供されない</li> <li>RACやOracle Clusterwareなどのスケールビリティ機能</li> <li>Redo Apply/SQL ApplyやRolling UpgradesなどのHA機能</li> <li>Advanced Security OptionやEncryption toolkitなどのセキュリティ機能</li> <li>Oracle Data MiningやOracle OLAPのBI機能</li> </ul>	<ul style="list-style-type: none"> <li>GUI管理ツールにはManagement StudioではなくSQL Server Express Managerという簡易版を利用</li> <li>以下の機能は提供されない</li> <li>バックアップログ配布</li> <li>フルテキスト検索</li> <li>SQL Agent job スケジュールリングサービス</li> <li>インポート/エクスポート</li> <li>レポートビルダ</li> </ul>	<ul style="list-style-type: none"> <li>オプション製品を購入することができない</li> <li>以下の機能は提供されない</li> <li>Spatial Extenderクライアントおよびサンプル</li> <li>Microsoft Cluster Serverサポート</li> <li>Informix データソースサポート</li> <li>レプリケーションデータキャプチャー (SQL レプリケーションおよびQレプリケーション)</li> <li>DB2 Web tools</li> <li>GSKit</li> </ul>

る (もちろん他社製品にも対抗する) 戦略をとっていると言えるだろう。

こうしたベンダ各社の動きは、マーケットにおけるオープンソースDBの認知度や存在感が非常に高まっていることの証拠と言えるのではないだろうか。

オープンソースDBの  
高機能化

一方のオープンソースDBにおいては、PostgreSQL (<http://www.postgresql.org/>) とMySQL (<http://www.mysql.com/>) の2強でほとんどのシェアを分け合う構図と言って良いだろう。ここで、両者について簡単に見ていこう。

## PostgreSQL

PostgreSQLは、BSDライセンスにより配布されているオープンソースのRDBMSである。米カリフォルニア大学バークレー校の研究プロジェクトにて開発されたPOSTGRESを起源とする。RDBMS実装のパイオニア的な役割を担ってきたが、ここにきて商用DBに比肩する機能を備えるようになってきた。Windowsに対応した時期がバージョン8以降と最近であることもあり、LinuxなどPC-UNIXでの導入実績が多い。

2006年12月5日にバージョン8.2をリリースし、エンタープライズ向けの機能をさらに強化した。具体的には、主に以下の性能改善、機能追加がなされている (性能改善には特に力が入っている)。

- オンライントランザクション処理の性能が20%向上
- メモリ上およびディスク上のソート処理の高速化
- マルチプロセッサ環境のスケールビリティ改善
- ウォームスタンバイデータベース

- アプリケーションがDBに書き込み中でもインデックス構築が可能
- 標準技術への準拠 (SQL:2003の機能が追加)

これらのほかにも200以上の新機能や改良が盛り込まれているが、特にエンタープライズシステムへの適用を視野に入れた機能強化が目立っている。

## MySQL

MySQLは、スウェーデンのMySQL ABの創業者によって開発されたオープンソースのRDBMSである。オープンソースの標準的プラットフォームを意味するLAMP (Linux, Apache, MySQL, PHP/Perl/Python) という言葉に採用されるなど、世界で最も有名なオープンソースDBとして知られる。

検索処理や更新処理の速さを重要視して開発されており、特に更新よりも参照の頻度の高いアプリケーションに向くとされている。そのため、Webシステム (一般的に参照系の頻度が高い) への導入実績が多い。なお、MySQLは、ほかのオープンソースプロジェクトと異なり、営利団体「MySQL AB」が保持していて、GPLライセンスと商用ライセンスのデュアルライセンスとなっている。

2005年10月にバージョン5.0がリリースされ、現在はMySQL EnterpriseとMySQL Community Serverという2系統が用意されている。MySQL Enterpriseは、その名の通り、性能面やセキュリティ面でエンタープライズ向けの機能が備わっている。GUI管理画面ではネットワークやシステム負荷の現状などをグラフで確認でき、各種アラートの表示など、稼働中のシステムの問題をいち早く正確に把握できる。有料で各種のサポートも提供されている。

一方のMySQL Community Serverは、従来のGPLライセンスで提供されるものだ。

広がるオープンソースDB  
の適用分野

オープンソースを使用したシステムについての調査結果 (図1) を見てみると、オープンソースは、今まではWeb系のような (多少の語弊はあるが) 信頼性がそれほど高くなくても特に大きな問題にならないシステムに導入されてきた。逆に言うと、基幹系のようなクリティカルなシステムではあまり導入されていなかったが、基幹システムへの適用に対する期待については「高まっている」と見ることができる。

一方、PostgreSQLやMySQLといったオープンソースDBの最近の動向からも、エンタープライズシステムへの適用を強く意識した機能強化が続いていることが読み取れる。これは、まさに次のような好循環が生まれていることの証と言えるだろう。

- オープンソースDBの機能強化がユーザーの関心を引き、「基幹系への導入」の期待を高めている
- 「基幹系への導入」という期待に応える方向でオープンソースDBの開発が進められている

また、先日、IPA (独立行政法人情報処理推進機構) が行っている「OSS性能・信頼性評価プロジェクト」の成果が発表さ

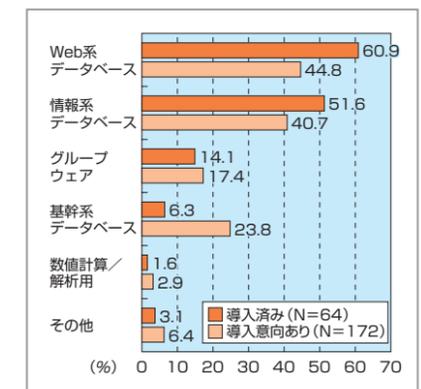


図1：オープンソースプロダクトを導入済み、あるいは導入意向のあるシステム (矢野経済研究所の発表記事より)

れたが<sup>※1</sup>、評価結果によると、「PostgreSQLやMySQLといったオープンソースDBはCPU数のスケラビリティが格段に向上し、チューニング次第で大幅な高速化や大規模対応が可能」とのことである。

詳細な情報や具体例はここでは省略するが、「OSSミドルウェアの適用における課題はもはや性能ではなく、信頼性や運用に関する要件を考慮した具体的な案件ベースの対応になってきている」とIPAはコメントしている。すなわち、オープンソースDBが商用DBに性能で劣る領域は、確実に狭くなってきているのである（チューニングの難易度や運用性などもあり、一概に言い切れるわけではないが）。

### 商用DBかオープンソースDBかの選択ポイント

ここまでで述べたように、商用DBのローエンドDB市場への進出、そしてオープンソースDBのハイエンドDB市場への進出が顕著となっている。すなわち、商用DBとオープンソースDBのカバー領域が確実に重なりつつあり、さらに重なる部分が拡大し続けていることが激戦に拍車をかけていると言えそうだ。

ではユーザー側は、この状況をどのようにとらえれば良いのだろうか。商用製品か

注1：http://ossipedia.ipa.go.jp/capacity/CS0702010349/

オープンソース製品かで迷ってしまうようなケースも増えていると思う。そこで以降では、その選択ポイントとして以下の2点に着目して考察してみたい。

- ① コスト…システム開発/運用に関わるトータルコストが適切かどうかという判断
- ② サービスレベル…システム要件（機能やパフォーマンスなど）を実現できるかどうかという判断

上記2点に着目すると図2のようなグラフを描くことができ、ここからは次のようなポイントが見えてこよう。

#### 商用DB

- 一定水準のコストがかかるが、サービスレベルにかかわらずコスト増減が少ない
- サービスレベル（使用機能）にかかわらず、基本的にライセンス料がかかる
- サービスレベルを上げても、運用サポート機能などプロダクトの機能として含まれる部分が多い

#### オープンソースDB

- 最小コストは低いが、サービスレベルに従ってコストが増加する
- ライセンス料が不要
- サービスレベルを上げると、プロダクト機能以外の機能の実装、チューニン

グや複雑な運用などが必要となる

つまり、商用DBかオープンソースDBかの選択ポイントとしては、グラフの交点を分岐点と考え、対象システムのサービスレベルが「オープンソースDBが有利」の領域（分岐点より左側）なのか、「商用DBが有利」の領域（分岐点より右側）なのかを見極めることが重要ということになる。

### サービスレベルによる選択ポイント

このように、商用DBはライセンス料はかかるが、サービスレベルが上がってもチューニングなどの手間が少なく、開発/運用にかかるコストの増大は少ない傾向にある。一方のオープンソースDBはライセンス料は無料だが、サービスレベルが上がるにつれチューニングなどに手間暇をかける必要があり、開発/運用にかかるコストが増大していく傾向にある。

極論を言うと、商用DBの生命線は「開発/運用にかかるコストを抑えることで、ライセンス料をどこまで挽回できるか」になる。この「開発/運用にかかるコスト」については、システムのサービスレベルが高くなればなるほどオープンソースDBとの差が広がるので、ライセンス料を挽回できるチャンスが大きくなる。

したがって、商用DBの命題は「プロダクトを高機能化することによってマーケット全体のシステム要件を、より高度なサービスレベルへと導くこと」となる。そうすれば、高度なサービスレベルを提供するには商用DBを用いるしかない（トータルコストが低い）という構図を作れるということだ。さらに言うと、オープンソースDBが有利な領域に無償版DBを投入している、と考えることもできよう（図3）。

これに対してオープンソースDBの今後の課題は、プロダクトの機能品質を上げる

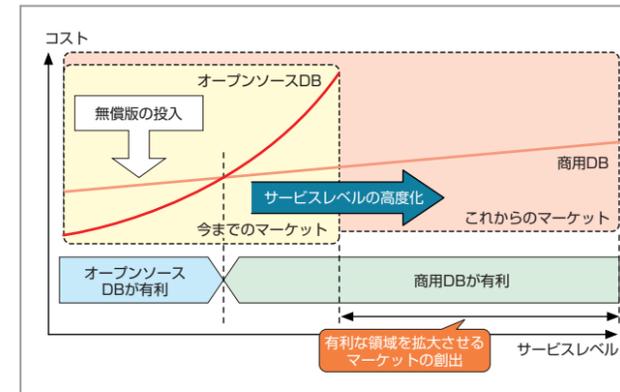


図3：商用DBの狙い

ことだけでなく、導入事例やノウハウ、開発者を増やすことにより、あるサービスレベルを実現する場合の開発/運用にかかるコストをさらに低く抑えることである。そうすれば、より高いサービスレベルを実現する場合でも、オープンソースDBがユーザーの選択肢に入ってくることだろう（図4）。

そのような視点で見ると、IPAによってオープンソースソフトウェアの性能/信頼性評価や、学校や教育現場へのオープンソース導入の推進など、オープンソース普及促進の活動が行なわれているのも見逃せない動きである。導入事例は確実に増え、その結果、次々に開発案件が後続し、ノウハウが蓄積され、ノウハウを持った開発者も増えていくという好循環を後押ししている。また、PostgreSQL技術者認定試験やMySQL認定試験など、オープンソースDBの資格制度が普及してきていることも、好循環の推進力となっている（図5）。

ここまで、商用DBかオープンソースDBかについて、サービスレベルによる選択ポイントを交えて説明してきた。しかしながら、商用DBとオープンソースDBの「分岐点」がどこなのかについて、一概に言うことはできない。これは、各プロダクトの機能や商用DBのライセンス形態が刻々と変わるのに加え、各プロジェクトの要件はさまざまだからである。

したがって、この機能を使う場合は商用

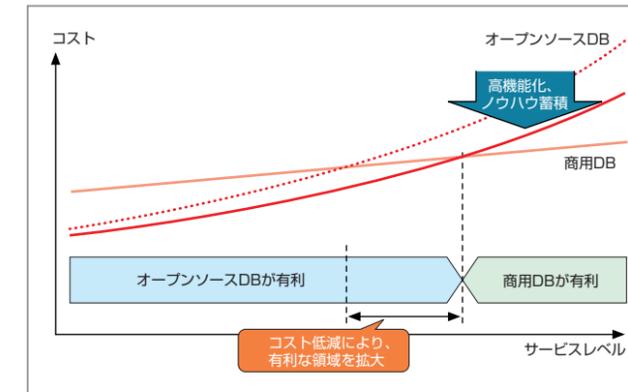


図4：オープンソースの狙い

DB、コストをこれ以下に抑えるにはオープンソースDB、というような固定的な線引きというよりも、常に「分岐点」を意識し、各プロジェクトの要件（システム要件のほか開発/運用メンバーのスキルなども含む）を明確にした上で、コストやサービスレベルをきちんと見極める目が重要となってくる。

### オープンソース採用に立ちはだかる現場の壁

一般論としては、ここまで説明してきたような棲み分けになっているのだが、筆者の経験上、実際の開発現場では必ずしもそのような棲み分けが当てはまるわけではないようだ。

例えば、商用DBを採用しているシステム開発の現場で、現場の開発者はもとより、関係者の誰に聞いても「なぜ商用DBを使っているのか」という疑問に答えられない人がいなかったり、見積もり時やシステム設計時のデータベース選択の際に、何の疑問もなく当たり前のように商用DBの導入が決定されるなど、このようなことは誰しも経験があるのではないだろうか。

これは、データベース（というよりもデータ）の特性によるところが大きいと考えられる。そもそもオープンソースプロダクトは、いわゆるWebサーバーなどでは

すでに「使って当然」というような風潮になっている。Apache HTTPサーバーなどはその好例だろう。

これに対して、DBサーバーは前述のような棲み分けを考えなければならないのだが、実際にオープンソースが使われる例はまだまだ少ない。やはり、Webサーバーのように「処理を行なう」ものではなく、「データを保存する」ものだからだろうか。

データベースは、データを保存するのが目的なので、刻々と適宜行なわれる処理ではなく、管理されるデータがすべてとなる。（これも多少の語弊があるが）何かあってもプログラムを修正すれば何とかなるWebサーバーとは違い、何かあったらそもそもその目的である「保存してあるデータ」がなくなる（読めなくなる）かもしれない。DBサーバーでは、この「何かあったら」の不安の割合が非常に高いのである。

このように、「データを保存する」とい

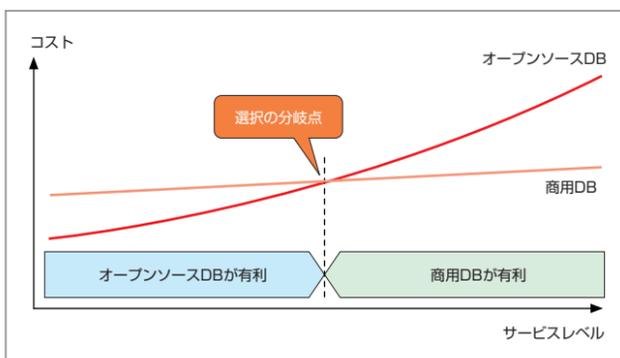


図2：商用DBかオープンソースDBかの選択ポイント

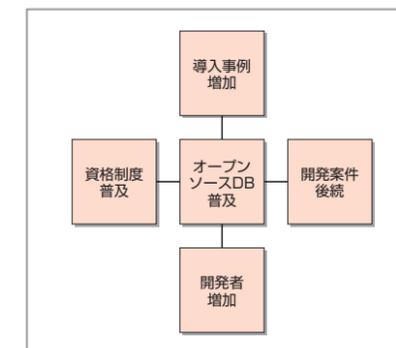


図5：オープンソースDB普及の好循環

エンタープライズでの  
オープンソース使いこなし術

う観点に立てば、商用DBのほうが「何となく安心」できるというのはうなずける。今日飲むワインを選んでくれるソムリエと、数年後に飲むワインを保存しておくワインセラーを選ぶとしよう。あなたならどちらを慎重に選ぶだろうか？ その感覚と似ていると思う。

この「何かあったら」という漠然とした不安は、リスク分析により不安の原因を具体化した後に、リスクへの対策案を提示することによって、ある程度は払拭できる。しかし、それでもなお、理詰めでは越えられない壁というのは存在するようで、「何となく安心」を選択するユーザーがいることも事実である。

サービスレベルの  
考え方の浸透

では、DBサーバーに関しては商用DBが最良の選択なのだろうか。もちろん、そうではないだろう。最近では、サービスレベルの考え方が浸透してきているので、サービスレベルが上がればコストも上がり、サービスレベルを下げればコストも下がる、という意識が高まっている。

すなわち、「コストの大小によらず、ある程度のコストをかければ、もれなく最高品質のものが手に入る」というよりも、「コストに応じて品質（サービスレベル）が上下する」というのが常識となりつつある。その意味では、ある程度品質を下げられる場合には、コストを抑えることができるオープンソースが最良の選択である、ということも言えると思う。

余談になるが、少し前だとファーストフード店で高級レストラン並みの味やサービスを求めるユーザーが、システム開発の現場では多かったように思う（もちろん原因はユーザーだけにあるのではないが）。この手の問題は、サービスレベルの考え方が浸透すれば、ある程度は解決すると筆者

は期待している。

もちろん、必ずしも「オープンソース＝サービスレベルが下がる」というわけではないが、サービスレベルを低くできるシステムに対しては、オープンソースが使いやすいというのは、導入事例を見ても分かるとおりの事実だと思う。ちなみに、当然のことながら、サービスレベルを下げることで、そのシステムを使う業務へのインパクトがあってはならない。そこはまさにSIerの腕の見せ所になるのではないだろうか。

商用DBとオープンソース  
DB導入のケーススタディ

さて、筆者の関わったプロジェクト（以下、PJ）で、サービスレベルの検討により、データベースを商用DBからオープンソースDBに切り替えた事例があるので紹介したい。紹介するのは、ある顧客における2つのPJで、1つ目のPJが商用DBを採用し、2つ目のPJがオープンソースDBを採用した。

2つのPJでは基本的な部分が同じような機能であったこともあり、1つ目のPJで構築したシステムを2つ目のPJでもそのまま使えないかということが検討された。この動きは、ちょうど1つ目のPJが安定稼働に入ったところに2つ目のPJがスタートしたことも追い風となったのだが、細かい要件の違いにより断念するに至り、別のPJとしてスタートすることになった。

では、2つのPJに共通する要件を見ていこう。2つのPJともに、治験の被験者を登録／管理するシステムである。

先に、一般の人には馴染みのない用語、治験と被験者について簡単に説明しておこう。「治験」とは、新しい薬が誕生するための最終過程に、有効性と安全性の評価を行なう制度を指す言葉である<sup>※2</sup>。

新しい薬が誕生する全体のプロセスは次のとおりだ。まず薬の成分が発見される

と、基礎研究や非臨床試験が行なわれ、その後「治験」に入る。治験が終了するとその結果が国によって審査され、審査に通ると市場に出回るのである（さらに、市場に出回った後で、再度、安全性や効き目を確かめる試験が行なわれる）。一般的に、成分の発見から審査の承認までに、およそ10年程度はかかるものであり、その中でも治験にかかる期間は3～7年と一番長いと言われている。

この治験において、薬の候補を服用するのが被験者である。治験によっては、被験者となれるかどうかについて、年齢や性別、検査の結果などで厳しく制限される。また、治験内容についての説明と同意（インフォームドコンセント）がないと、そもそも被験者になることはできない。そのため、被験者が同意した日付や検査結果の数値などについて、きちんと管理することが求められる。

このように、被験者を登録／管理するというのは、治験において非常に重要であり、治験システムが備えるべき基本的な機能だと言える。2つのPJでは、この基本的な機能部分について、かなり似通った要件を備えていた。

2つのPJで本システムを使った業務の流れを簡略化して説明すると、次のようになる。

- ① 被験者が治験参加の意思を示す
- ② 被験者が治験に適合するかを判断する
- ③ 被験者に治験の薬剤を割り付け、被験者が治験薬を服用する
- ④ 何か異常が発生した場合には、治験薬の服用を中止する
- ⑤ 上記について、適宜、進捗状況を確認する

つまり2つのPJ固有のシステム要件は、

注2:実際には、「薬の候補」を用いて国の承認を得るための成績を集める臨床試験のことを治験と呼ぶことが多い。

次のようなものであった。

- 被験者を登録できること
- 被験者が治験に適合するかどうかを判定すること
- 被験者に薬剤を割り付けること
- 適合しなかった被験者、薬剤を服用した被験者など、被験者を集計して表示すること

もちろん、このほかにも細かい要件はあったが、基本的には非常に簡素なシステムと言える。ただし、このPJ固有の要件以外に、治験特有の要件として、データの変更履歴をすべて残すというものがあつた。つまり、治験システムでは、データがどのように蓄積（入力／更新／削除）されていったかが非常に重要なのである。

もっとも、治験終了後にデータを解析することにより薬剤の効果を明らかにし、当局の申請／承認プロセスを経て薬として認可されることを考えると、データの正当性を明らかにしなければならないという意味で、上記のような要件が発生するのは当然であろう。

また、両PJともに、医師がWebを介して直接システムを操作するので、使用できる時間帯をできるだけ長くする必要があつた。これは、診療時間内にシステムを操作するのではなく、診療時間が終わった後にシステムを操作したいという強い要望から

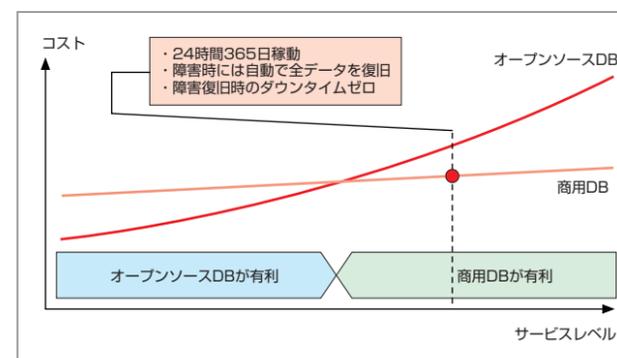


図6：商用DBプロジェクトの状況

である。

さて、細かい要件はいろいろあるにせよ、両PJともに「データを確実に蓄積する」「サービス時間を延ばす」という要件が特徴的なポイントとなっていた。またデータ量や同時アクセス数については、特に大規模というわけではなく、むしろ小規模と言えるレベルであった（被験者数：100～数百名程度、同時アクセス：10～数十程度）。

すなわち、両PJでの要件で非常に重要なものがトランザクション量やボリュームではなく、耐障害性や可用性だということが分かっていただけたかと思う。

## 商用DBプロジェクト

商用DBプロジェクトでは、下記のようなソフトウェア構成でシステムを構築することになった。この構成に決まった一番の理由は、ユーザーにおけるそれまでの運用実績であった。

OS : Windows  
DB : Oracle9i  
AP : Tomcat  
Web : Apache

このPJでは、サービスレベルの検討よりもシステム構成が先に決まった感じであった。そのため、サービスレベルについては、次のように上記構成で可能な最高レ

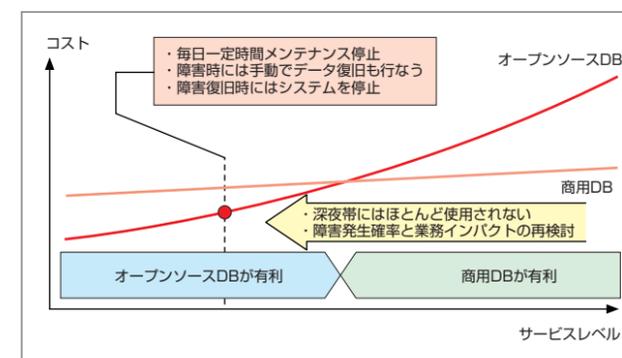


図7：オープンソースDBプロジェクトの状況

ベルに決まった。

- 稼働時間：24時間稼働
- データ保障：最後のトランザクション終了時点まで自動復旧
- 耐障害性：正副2台のホットスタンバイ。障害復旧時のダウンタイムゼロ

このサービスレベル程度を図6のようなグラフで描けるとすれば、結果論になるが、本PJにおいては商用DBの選択肢が適切であったと判断ができる。

オープンソース  
DBプロジェクト

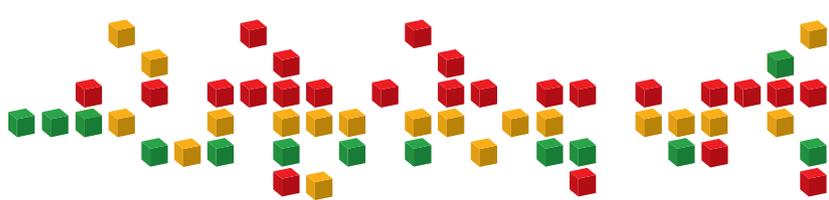
一方のオープンソースDBプロジェクトでは、次のようなソフトウェア構成でシステムを構築した。本PJにおいてはコスト低減の圧力が強く、オープンソースの採用を強く要請されたこともあり、類似PJとして上で説明した商用DBプロジェクトがあるにもかかわらず、違うシステム構成を採用する運びとなった。

OS : Linux  
DB : PostgreSQL  
AP : Tomcat  
Web : Apache

また、オープンソースDBを採用するので、サービスレベルを商用DBプロジェク



# エンタープライズでの オープンソース使いこなし術



トと同レベルにすると、開発／運用コストが高くなる恐れがある。これは、図6を見ればお分かりいただけるだろう。そのため、本PJでは、サービスレベルを、次のように見直すことにした(図7)。

- 稼働時間：0:00～6:00をメンテナンス時間として、システム停止
- データ保障：前日バックアップ取得時点まで(それ以降は手動復旧)
- 耐障害性：正副2台のホットスタンバイ。障害復旧には、システムを止めて復旧作業を行なう

これならば、オープンソースDBを採用した場合でも、コストを抑えた現実的な運用が可能だということが言えよう。これは「オープンソースDBを採用する＝コスト低減」というイメージにとらわれず、サービスレベルを落とさないと総コストが膨れ上がるというサービスレベルの考え方があったからこそ、実現できたと言って良いだろう。

## オープンソースのメリット

上で紹介した事例では、ユーザー企業で次のようなメリットが生まれた。

- 低コストでの開発(単純に、データベースのライセンス料を無償とすることができた)
- サービスレベルや業務フローの見直し(直接的なメリットではないが、オープンソースDBの採用を検討したことで、サービスレベルや業務フローを再検討して最新化できたのはメリットでもある)

また、ユーザー企業以外にも、SIerにとっては、オープンソースやサービスレベルを組み合わせて多種多様な提案が可能に

なることで、提案バリエーションが増えるというメリットがある。今までの「食材が高いから料理も高い」世界に加えて、「安い食材が手に入るようになったので料理も安い」世界が生まれたことで、純粋に「料理人の腕前=SIerの提案力/構築力」が問われるようになってきている。このSIerにおけるメリットは、同時にユーザー企業の選択の幅が広がるという意味で、ユーザー企業のメリットにも直結していると筆者は考えている。

## 注意点および デメリットとリスク

当然であるが、オープンソースDBの導入については、そのメリットだけでなく注意点やデメリットについても知っておかなければならない。

まず注意したいのは、いわゆる「技に溺れる」ことである。技に溺れるとは、ここで言うと難しい技術や運用方法を採用することにより、実用に耐えないシステムになってしまうことである。もちろん、これはオープンソースDBに限った話ではないが、現状では同じことをやろうとすると、商用DBよりもオープンソースDBのほうが手間がかかることが多いので、技に溺れやすくなるのである。

さらに言うと、オープンソースDBの一時的なコストメリット(ライセンス料が無償)に目を向けすぎて、現実に即した実現可能性や運用コストに目が行かなくなると、本番稼働後にトラブルが多発する可能性が高まる。最近では、オープンソースが一種の「流行り」になっているので、何が何でもオープンソースというのではなく、現実に即した冷静な判断が求められる。

次に、デメリットとリスクであるが、やはり基本的に「自分の身は自分で守らなければならない」という点は押さえておかなければならない。システム要件として非常

に特殊な要素があったり、長期にわたってデータベースの使用が予想される場合などは特に注意が必要である。

例えば、何かトラブルが発生した場合に、SIerがしっかりしている、あるいはサポートサービスを提供している企業があるということであれば、対応は可能かもしれないが、基本的にはユーザー側の自己責任と考えておいたほうが良いだろう。

同様に、SIerがユーザーにオープンソースDBの採用を提案する場合も、こうした注意点やデメリット/リスクを総合的に検討した上で、ユーザーに提案すべきである。

\* \* \*

以上、本パートでは、商用DBとの比較を交えながらオープンソースDBのメリットやデメリットを見てきたが、いかがだっただろうか。

要求されるコストとサービスレベルから選択の「分岐点」を冷静に見極めることで、オープンソースDBを商用DB並みに適用できることを簡単な事例とともに示した。今後こうしたアプローチによる採用事例が増えることで、さらにノウハウが蓄積され、低リスクでもより多くのメリットを享受できるオープンソースDBシステムの普及が進むものと筆者は考えている。

DBM

### 上川伸彦(かみかわのぶひこ)

(株)ビーブレイクシステムズ技術担当取締役。大規模向けデータベース製品の開発、介護業界、電子部品業界でのEDI標準メッセージなどのオープンスタンダード仕様の策定や、SOA構築基盤の設計に携わる。現在は、社内外の技術コミュニティ活動を推進する立場にある。

### 鷲野秀次(わしのひでつぐ)

Javaテクノロジスト。ASP、C++などでの業務システム開発や独自言語でのERPパッケージ開発を経て、自分の技術的な限界を知るために上京、Java開発専門のビーブレイクシステムズに転職し、現在、Java設計者として業務に携わっている。また、仕事面だけではなく社内サークルの中心的な存在としても活躍している。