

住友電工における OSSの基幹システムへの利用例

2006年 5月30日

住友電工情報システム(株)

岩佐 洋司

1. 住友電工の会社概要

創 業	1897(明治30)
資本金	968億(2006年3月)
売上高	単体 9,120億円、連結 20,071億円(2006年3月期)
経常利益	単体 293億円、連結 1,132億円(2006年3月期)
従業員数	単体 5,000人、連結 100,000人
製 品	(情報通信)ネットワーク機器、光ファイバ等 (エレクトロニクス)化合物半導体、フレキシブルプリントサーキット等 (自動車)ワイヤーハーネス、ディスクブレーキ等 (産業用素材)PC銅線、ダイヤ切削工具等 (社会インフラ)電力線、超伝導線等

2. 情報システム部門の概要

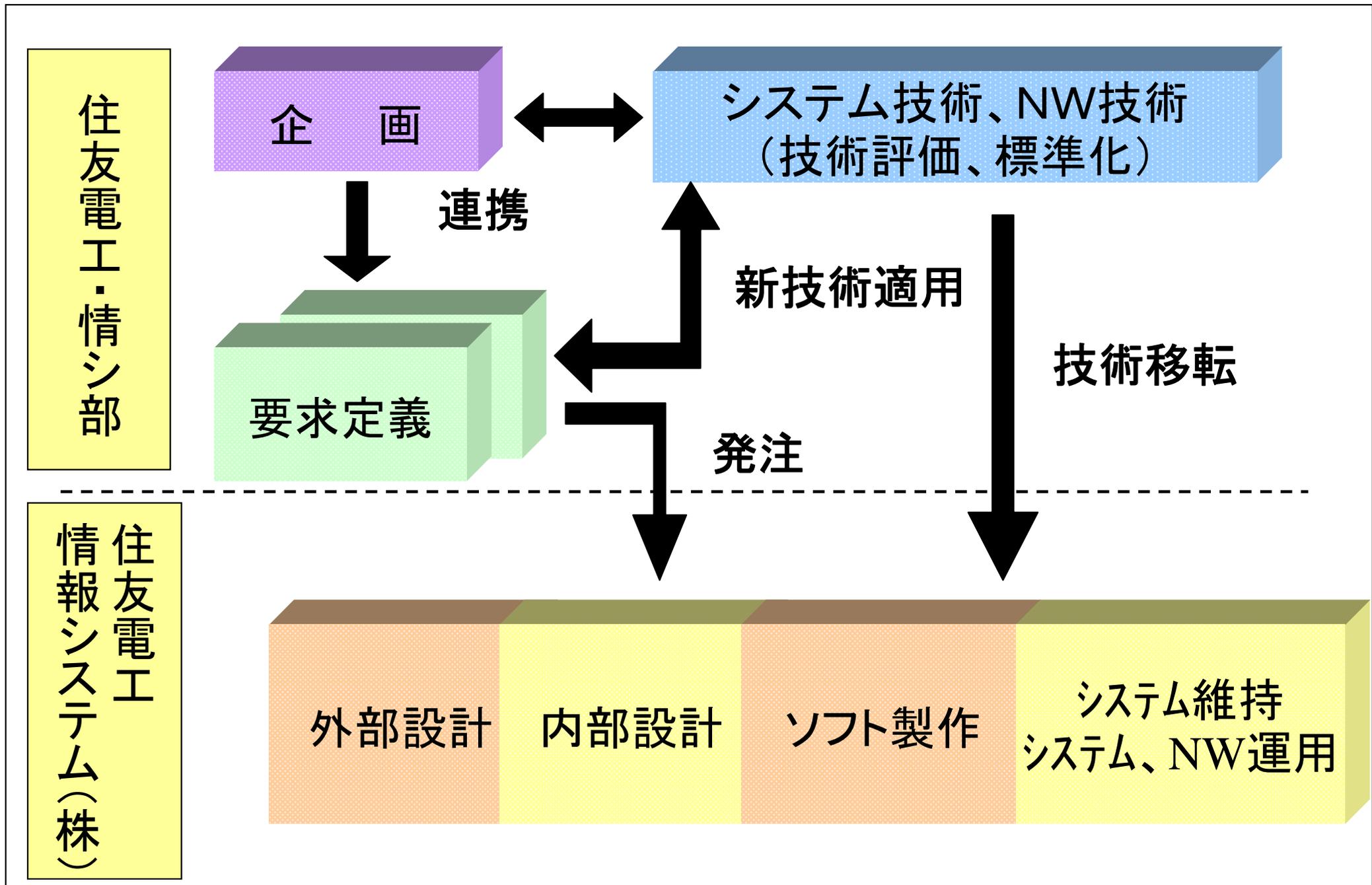
(1) 住友電工・情報システム部(50名)

- ・情報システム企画
- ・各事業部及び国内、海外関係会社のシステム化企画、推進
- ・システム技術／ネットワーク技術

(2) 住友電工情報システム(株)(250名)

- ・1998年設立
- ・業務システムの設計、構築、保守
- ・パッケージ・ソフトの開発、販売
- ・コンピュータ・センターの運用

情報システム部門の体制



3. オープンシステムへの取り組み経緯

年度	方式	OS	言語	DB
70年代	汎用機集中処理 (ミニコンの活用)	IBM S/370 NEC ACOS4 DEC PDP11	COBOL (BASIC)	IMS ADBS
80年代	汎用機分散設置	IBM MVS/VSE NEC ACOS2		DL/I、DB2 ADBS
91～94	分散処理(telnet)	UNIX	Informix4GL	Informix
95～96	クライアントサーバー処理		Developer2000	Oracle
97～98	Webシステム	Windows NT	Cold Fusion	
99～04		Linux/Tomcat	Java	DB2, Oracle
05～				PostgreSQL

- ・クライアントサーバー処理方式の期間が短い
- ・Webシステムへの早期取り組み
- ・現在、約600台のサーバーが本番稼働(Linux, UNIX, NT)

4. 業務システムの考え方

受注	各事業部システム	電線営業システム			
設計			電線各事業部システム
製造						
外注						
出荷						
原価管理						
購買	購買システム					
物流	物流システム					
経理	経理システム					
人事	人事システム					
総務	総務システム					

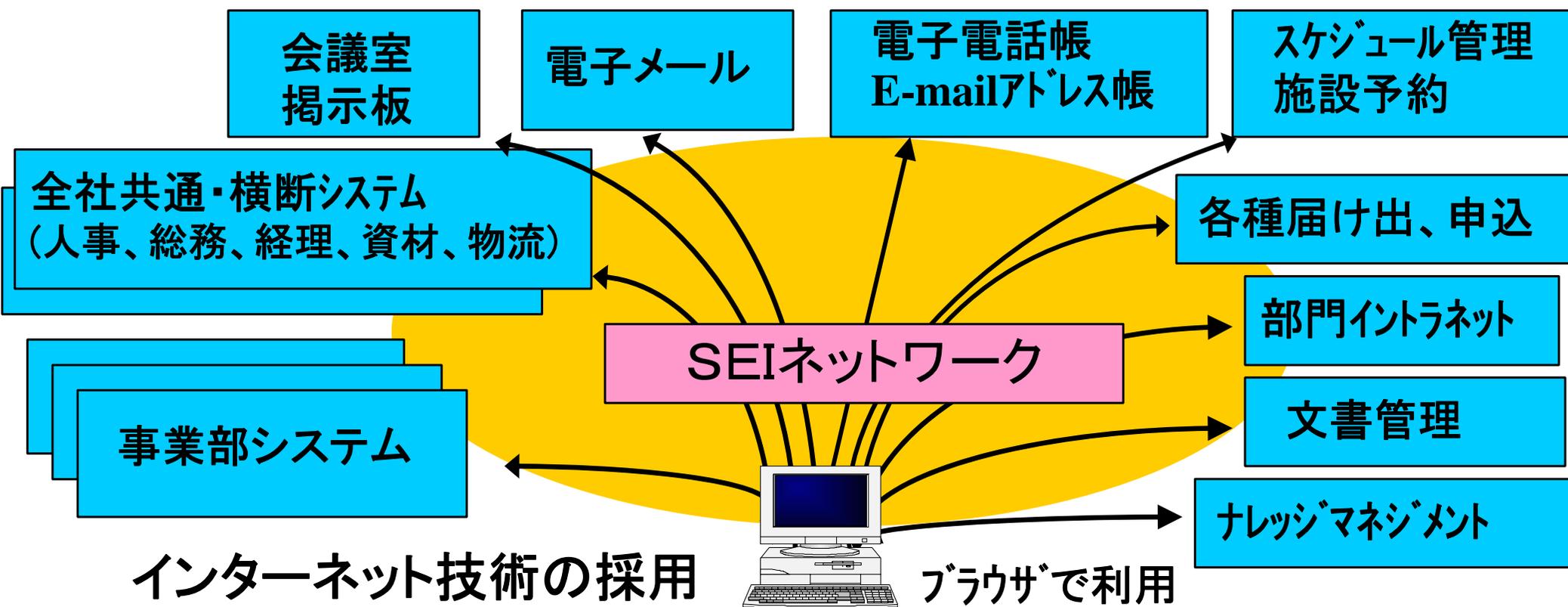
Linux

UNIX等

汎用機

5. イン트라ネットによるシステム基盤

— 1台のPCブラウザ(*)で全ての業務を実行可能とする —



① TCOの削減

- ・ブラウザ、メールソフトのみで業務が可能
- ・クライアントPCの管理工数削減

② 最新技術の導入が容易

- ・インターネット技術の進歩が早い
- ・費用が安い

97年：イントラネットシステムによる基幹システム開発方針決定

(*) ブラウザはNetscape、IE、FireFox等、一般によく使われているもので動くこと

6. システム開発の生産性向上の取り組み経緯

	取り組み事項	言語	成果
70年代 80年代	汎用機での生産性向上の 取り組み	COBOL (BASIC)	汎用機での生産性向上
1991	Informix4GL用ジェネレータ開発	Informix4GL	開発フェーズ 生産性 30%UP
1994	T字形ER設計手法導入 ——佐藤正美氏(DOA導入)		D2000
1997	ファンクションポイントの導入	ColdFusion	計測方法の見直し
1999	楽々Framework 自社開発 (View,Controller)	Java	ユーザインタフェース・コンポーネン (Struts相当の部品)
2001	システム開発プロセス改善(CMM)		CMMレベル3(03年4月)
2003	組立型開発の開始 (コンポーネント元年)		業務用コンポーネント300以上
	ビジネスロジックの部品化		販売管理システム
2005	CMMIレベル5の取り組み開始		

7. Linux/Javaの採用 – 1999年に決定–

(1)Linuxの採用

1) メリット

- ① ハード、データベース・ソフトが安価
 - ・PCサーバで稼働
 - ・データベース・ソフトがUNIXに比べ安価
- ② システムの安定性
- ③ オープン・ソース → 今後の主流と判断

2) 課題

Linux上の運用、開発ツールが少ない

(2)Javaの採用

1) メリット

- ・ソフトの部品化に優れる
- ・ベンダーフリー

2) 課題

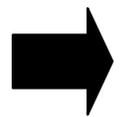
- ・オブジェクト指向技術者の不足(従来は4GL他を使用)

8. Javaフレームワーク(楽々Framework)の開発 ¹⁰

(1) 基本コンセプト

— DOAとオブジェクト指向の融合 —

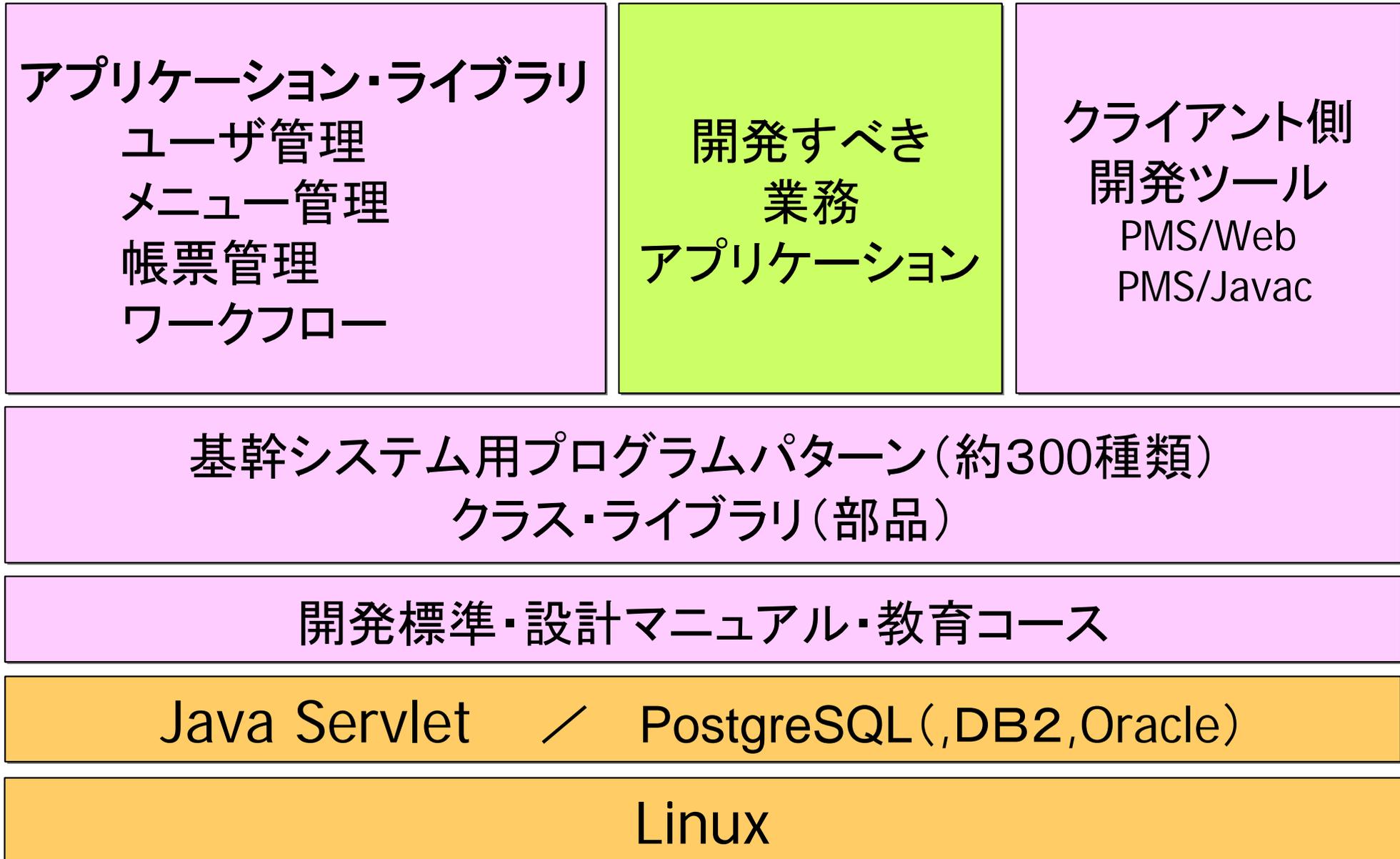
- ・ DOA(Data Oriented Approach)による設計
確立されたデータベース設計手法、仕様の安定性
プログラムの単純化
→1FP当たりのコーディング量の大幅削減
- ・ オブジェクト指向のメリット
再利用可能な部品の創出
 - 画面出力
 - メインプログラム



基幹システムを短期間で、低コストで構築する

DOA+ コンソーシアム発足(2003年12月)

(2) 楽々Frameworkの構成

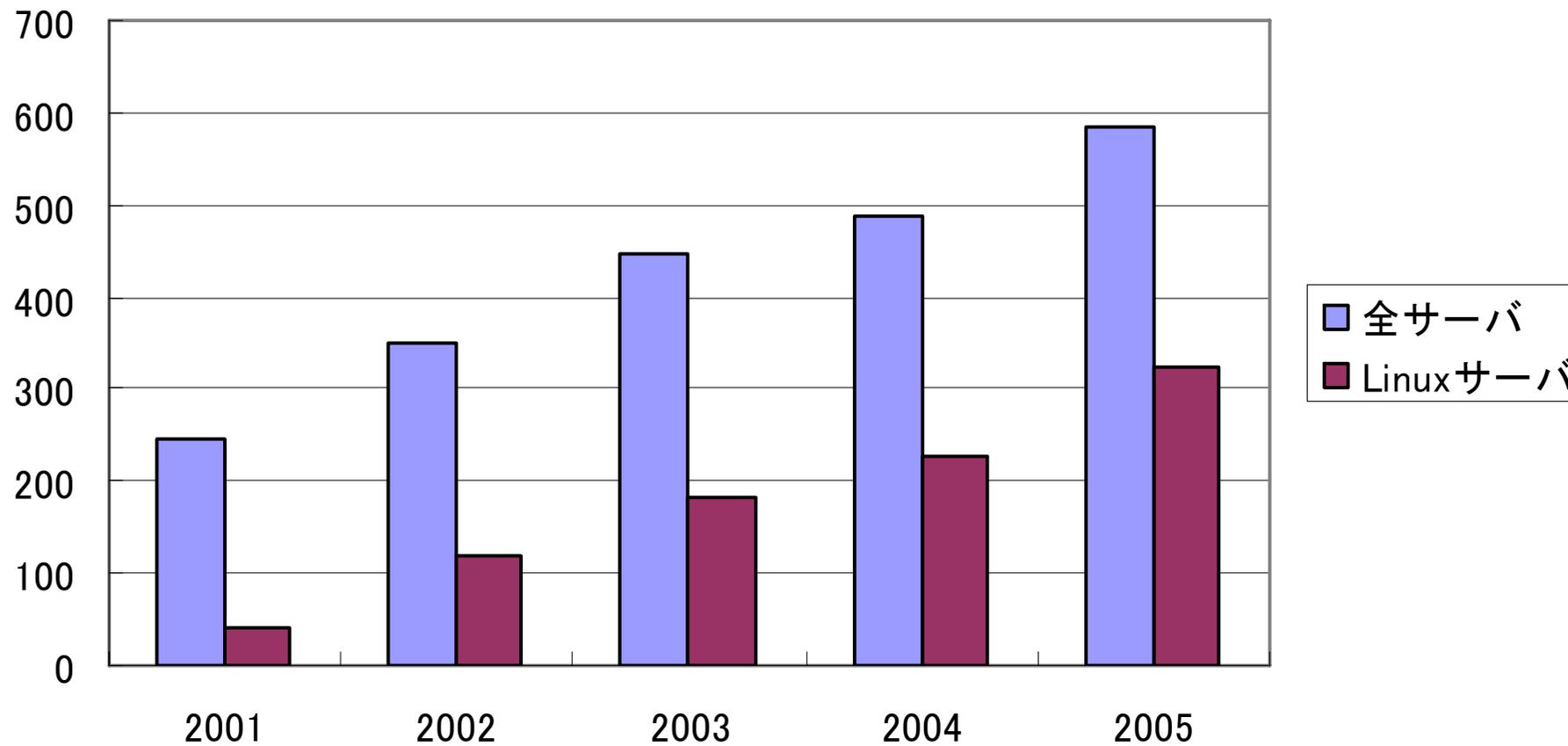


10. Linux/Java による開発事例

システム名	システム概要	稼働時期
購買ECシステム	購入依頼から入荷検収までの一連の社内購買業務と取引先とのデータ交換	2000年 7月
超硬工具EC	超硬工具のポータル・サイト	2000年10月
Web受注システム	Web-EDIを利用した受注システム	2000年10月
原価管理システム	設計及び原価管理	2000年10月
生産管理システム	工場の生産管理システム	2001年 4月
SCMシステム	販売、製造、物流管理システム	2001年 6月
全社勤惰管理システム	各種工場勤務形態に対応した勤惰管理	2001年 9月
生産管理システム	工場の生産管理システム	2001年10月
固定資産管理システム	償却計算、異動廃却処理	2002年10月
海外工場システム	中国工場の生産管理システム	2002年11月
全社起案・申請システム	決裁電子化(ワークフロー)汎用システム	2002年12月
社内ポータル	企業ポータルシステム、グループウェア	2003年 4月
グループ人事・給与システム	グループ共通人事管理、給与計算	2003年 4月
グループ経理システム	グループ共通経理システム	2004年 4月

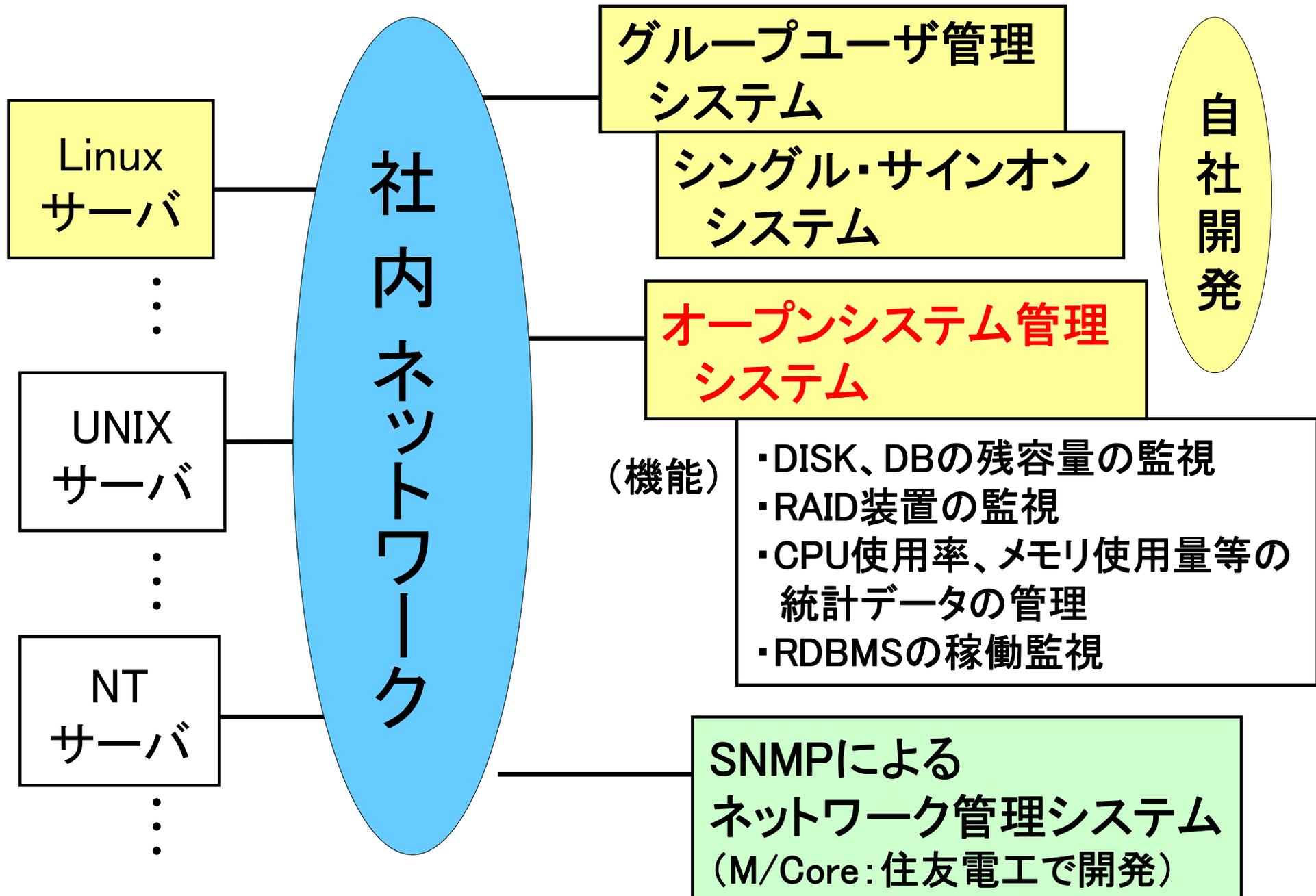
80システム、300台強のLinuxサーバ

11. Linuxサーバー台数の推移

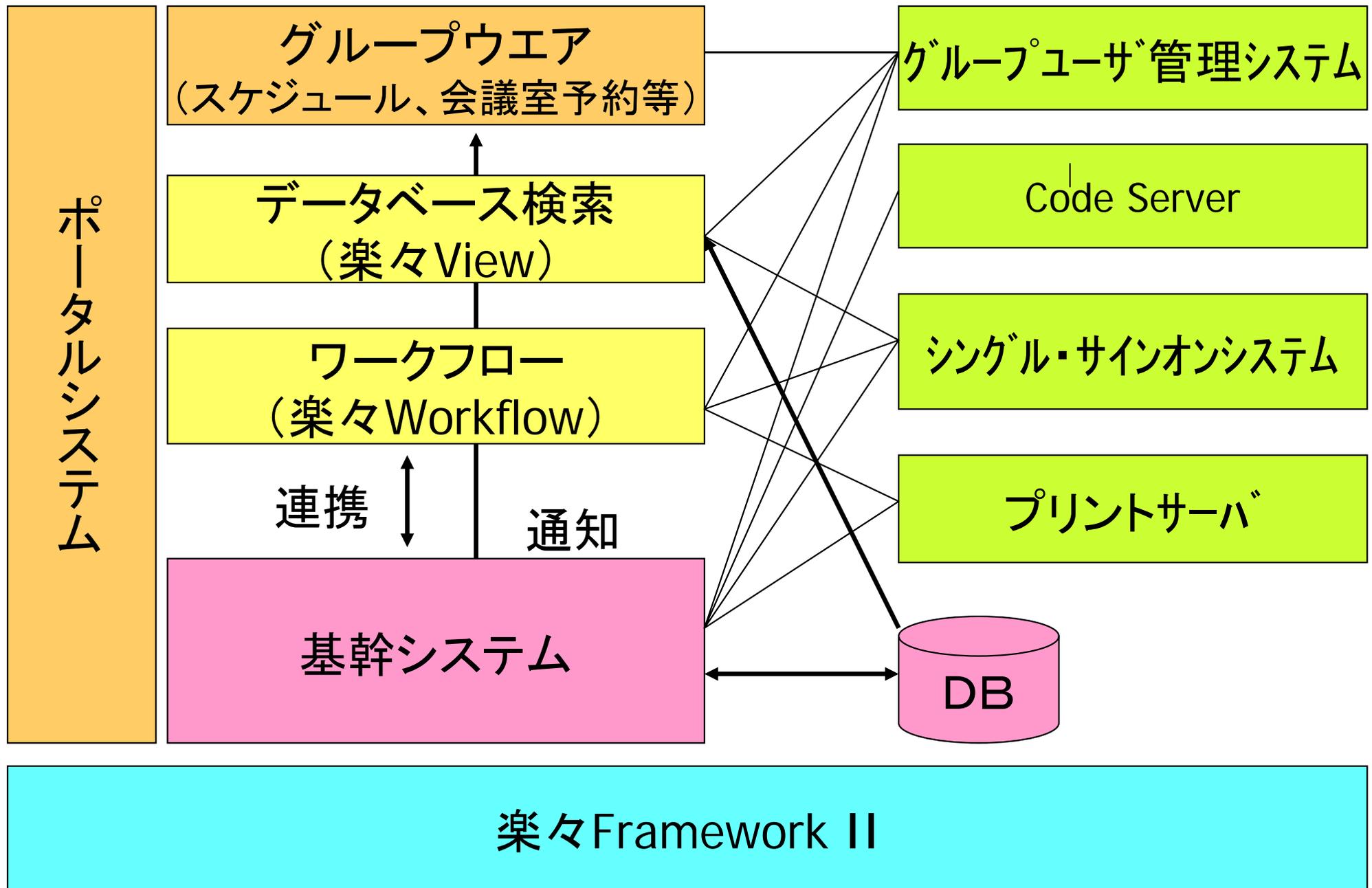


12. オープンシステムの管理とインフラシステム

(1) オープンシステム管理システム



(2) インフラシステム (全て Linux + Java)



13. 現在利用している主なOSS

分類	ソフトウェア	バージョン
OS	SUSE Linux	SLES9
	RedHat Linux Server	RHLE3.0
実行環境	Java(＊)	J2SE5.0
	Tomcat	5.5
開発ツール	楽々Framework II(＊)	3.0
	+Eclipse	3.1
DB	PostgreSQL	8.1
	MySQL(一部)	4.0
ブラウザ	FireFox	1.5

(＊)OSSではない

14. OSS導入で留意したこと、トラブルなど

14.1 Linuxの導入

(1) 留意したこと

- ・システム技術部門で事前に新バージョンを導入、評価
そこでOKとなったバージョンを開発部門にリリース。
- ・バージョンにより、ハードウェアのサポート(デバイスドライバ)
が制限される場合がある。
- ・**ディストリビューション固有の機能は極力使わない。**

バージョン	導入	1999	2000	01	02	03	04	05	
RedHat 6.2/SE	○	→							
RedHat 7.0	×								
RedHat 7.1	×								
RedHat 7.2	○					→			
RedHat 7.3	○				→				
RedHat 8.0	○						→		
SUSE LINUX 9	○							→	

(2) Linux システムで発生したトラブル

1) デバイス・ドライバによるトラブルが大半

① UPSドライバ

2 CPUマシンで電源トラブル(開発機)

② RAIDカード・ドライバ

OSのインストールに3週間要した

2) Linuxのトラブルでないが

HDDのハード障害で苦勞

14. 2 Tomcat の導入

(1) 特徴

無償利用可能(再配布の制約も少ない)

(2) 評価

5年間の基幹システムの利用でほとんどバグがない

(3) トラブル

Tomcat ではなく、Java(JDK/JRE) のオプティマイザによる不具合が発生

14.3 PostgreSQLの導入

(1) 1999年からパイロットシステムでOSS RDBMSを活用

PostgreSQL --- 全社ポータルシステム(1万人が利用)

MySQL --- オープンシステム管理システム

シングルサインオン・システム

(2) PostgreSQL 対 MySQL の評価

PostgreSQL	MySQL
<ul style="list-style-type: none"> ・高機能(再帰SQLは未実装) ・フリーソフト(コミュニティによる運営) 	<ul style="list-style-type: none"> ・高速作動、シンプルな機能 ・外販時は有償ライセンス

(3) 2005年--今後の新規システムは、PostgreSQLに決定

(後述) **PostgreSQLを用いた正規化の実験評価**

現時点の標準システム

Linux, Tomcat, PostgreSQL, FireFox, Java(楽々Framework)

15. OSS活用の評価

(1)ハードウェア及び基本ソフト(OS、DBMS)の
大幅費用削減

(2) システムの安定性

Linux導入後、6年経過するが、本番稼働後のシステムトラブルはない。

(3) 最新技術の導入が容易で、IT関連費用が安くなる。

- ・OSSの技術進歩が早い。
- ・64ビットマシン等の技術進歩をいち早く享受できる。
- ・まず実験、評価することが容易にできる。

(4) クライアント管理費用の削減

- ・OSSのブラウザ、メールソフトのみで業務が可能

課題

クライアント(デスクトップ)のOSS対応

16. OSSの今後の活用

(1) 基本ソフトはOSS

1) サーバ

Linux、WAS (Tomcat等)

RDB (PostgreSQL等)

2) ブラウザ

FireFox

(2) 64ビットマシンの性能を生かしたシステム構築

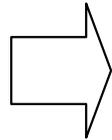
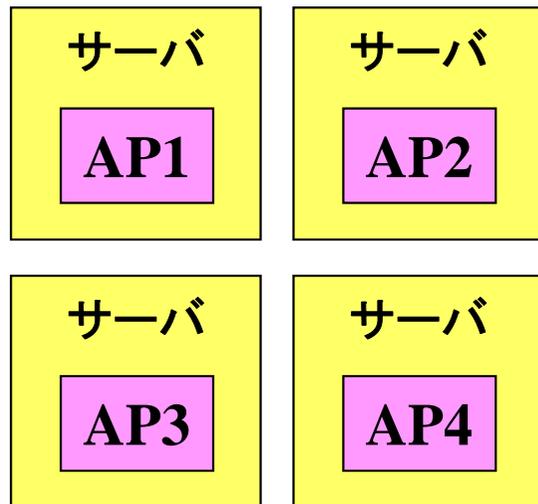
- ・CPU能力、大きなメモリー空間の活用
- ・サーバ統合

(3) Linuxサーバ統合

1台のPCサーバで複数システムを稼働

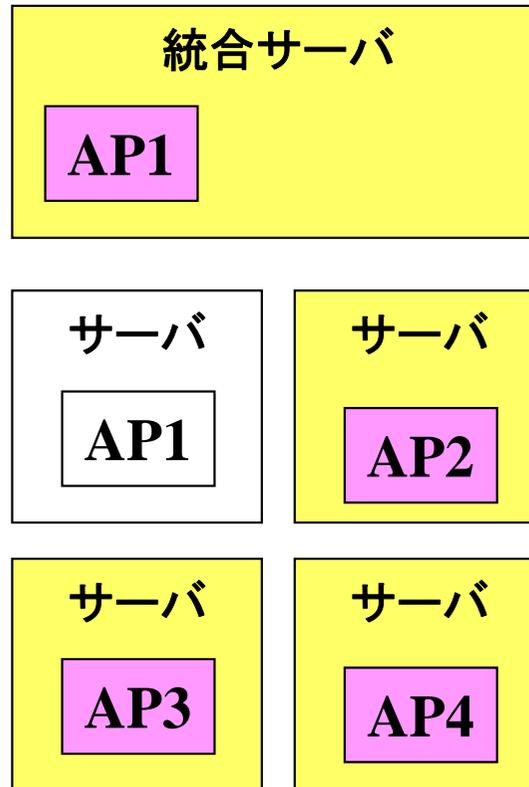
Linuxサーバ統合の取り組み

今まで



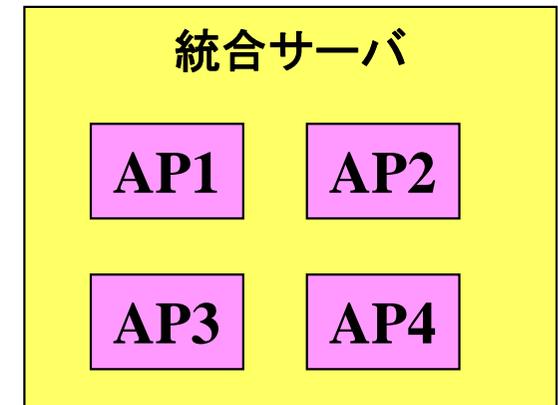
今後

ステップ1



今後

ステップ2



PostgreSQLを用いたパフォーマンス実験

出典:「DOA+コンソーシアム 第3分科会公開データより」

DOA+ Consortium Website - Mozilla Firefox

http://www.doaplus.com/html/bun03_20051101.html

第3分科会

サイトマップ お問い合わせ

- HOME
- DOA+
- DOA+とは
- 研究報告書
- 記事
- コンソーシアムご案内
 - コンソーシアムの目的
 - 会則
 - 組織
 - 活動内容
 - 入会のご案内
 - お問合せ
 - 個人情報保護
- 会員向けページ
 - 第1分科会
 - 第3分科会
 - 会員情報の確認

実証実験1：非正規化すると本当に速いのか

正規化と非正規化の応答速度実証実験

「正規化するとレスポンスが悪くなる。」という誤った認識から抜け出せませんか？
 正規化したRDBにちょっとした工夫を施すだけで、JOINによるパフォーマンスの劣化が起こる心配はありません。むしろ、すばらしいパフォーマンスを見せてくれます。

DOA+コンソーシアムでは、数々の無償セミナーを通じて正規化によるパフォーマンス劣化の誤解を糾してきました。しかし、百聞は一見にしかず。今回、実証実験を通じて我々の主張が正しいことを世に問い直そうと思立ち、「正規化vs非正規化」の論拠を示すことになりました。

【実験結果】
 5000万件の正規化されたデータを非正規化により500万件にしてもほとんどパフォーマンスの向上は得られず、逆に、Joinする場合は非常に遅くなることがわかった。

- 正規化されたDBは高速に動作する。
 Indexを適切に利用することで正規化されたDBは非正規化されたDBよりも高速に動作する。
- 非正規化されたDBはJOINすると非常に遅くなる。
 非正規形のDBを使っている技術者はJOINが遅くなることを体験しており、「正規化するともっと遅くなる」と誤解している。

→詳しい内容は[こちら](#)。
 →実証実験に使用したテーブルレイアウト

上記資料をこちらからダウンロードできます
【資料ダウンロード】

No.	タイトル	作成者	資料
1	正規化と非正規化の応答速度実証実験	第三分科会ワーキンググループ	配布資料1.583KB (PDFファイル)
2	実証実験に利用したテーブルレイアウトと生成用SQL	第三分科会ワーキンググループ	配布資料21KB (ZIPファイル)

実験	処理	正規形	非正規形
実験1	未手配検索	2ms	2ms
実験2	1件表示	3ms	2ms
実験3	発注数量集計	91ms	14020ms (*1)

(*1) 実験3の結果は 7,010ms であるが、全項目を取得する為に2回SQLを発行する必要がある為、2倍している。

DOA+コンソーシアム
 で公開
www.doaplus.com