

Linux のアプリケーションを Solaris 10 への移行する方法

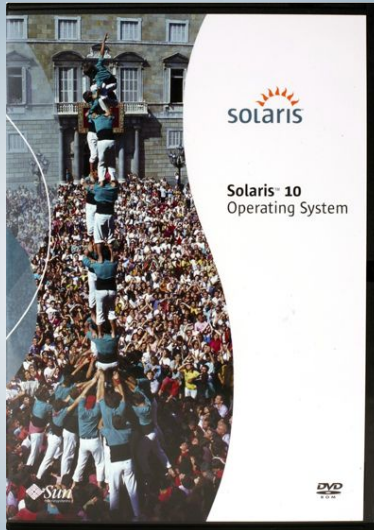
サン・マイクロシステムズ株式会社
ISV エンジニアリング
伊草英雄



Agenda

- なぜ Solaris 10?
- 実際の移行プロセス
 - > 開発環境の確認
 - > アーキテクチャの違い
 - > OS(Operation System) の違い
 - > Solaris のパッケージ作成
- ISV 様向けポータリングサポート
- 参照

なぜ Solaris?




- Dynamic Tracing (DTrace)
- 仮想化
- 高度なセキュリティ
- Predictive Self-Healing(予測的自己修復)
- ZFS
- 180 以上のオープンソースのアプリケーション
- 900 以上のシステムをサポート
- 互換性の保証 (source code & binary code)

Solaris 10 Operating System



solaris™

\$0



open source™

実際の移行のプロセス

実際の移行プロセス

- 使用しているサード・パーティーソフトウェアの確認 / 用意

http://www.sun.com/io_technologies/index.html

- 開発環境の確認
- アプリケーションソース上で Solaris と Linux との相違をチェック
- 機能テスト
- パフォーマンスの測定とチューニング
- ソフトウェアのパッケージ作成 / マニュアルのアップデート

開発環境の確認

SAMP vs. LAMP

Solaris Cool Stack

- Apache, MySQL, PostgreSQL, Perl, Ruby, PHP, ...
- その他のパフォーマンス向上用のライブラリも入っています
- Sun Studio 12 でコンパイルされ最適化されています
- インストールはダウンロードしてパッケージを追加するだけ

コンパイラ :GCC

- プラットフォームユーティリティまたはGNU の binutils パッケージと共に使用できます
- Linux と開発環境の互換性が良い

コンパイラ : Sun Studio

Sun Studio 12:

- 無料!
- Solaris と Linux に対応
- 使いやすい GUI
- 高い性能のライブラリ
- コンパイルする際には **`:-fast`** オプションと共に!
 - > パフォーマンスチューニングの最初の一步
 - > コンパイルしたアプリケーションが速い
- CMT および 64 bits 用に最適化されます
- 強力なデバッグツール、プロファイラー

どのコンパイラを使用すべき？

- アプリケーションの開発の為にどのコンパイラを使用するかは自由に選択することができます
- もしパフォーマンスの問題がクリティカルな場合：
 - 両方使用してみてください
 - そしてアプリケーションのパフォーマンスを比較してみてください
- コンパイラフラグはコンパイラにより変わります：**makefiles** を修正が必要になります



solaris™



- Solaris 10 の機能、さらに 150 のオープンソースのツールを提供
- Solaris 10 で提供されていない、多くの新機能を提供：最新のワイヤレスドライバ、xVM、StarSuite 8 や GNOME 2.20 に含まれる多くのデスクセットの向上
- 開発環境：Sun Studio コンパイラ、およびツール、NetBeans 統合開発環境、GlassFish アプリケーションサーバ、Java Platform、Standard Edition 6、Web アプリケーションスタック、データベース、その他

Solaris Express Developer Edition
<http://developers.sun.com/sxde>

アーキテクチャの違い

エンディアン

- 情報をコンピュータのメモリーに蓄える方法にエンディアン(バイト順序)の違いがあります
- Sparc MSB: ビックエンディアン、 x86 LSB: リトルエンディアン
- エンディアン(バイト順序)の違いは問題を起こすことがあります。特に アプリケーションが、ハードウェア機能を最適に利用する低いレベルのコードを使う場合は注意が必要です

例) 0x01020304

Address:	100	101	102	103
X86:	00000100	00000011	00000010	00000001
Sparc:	00000001	00000010	00000011	00000100

ネットワーク上のデータの送信

- ビッグエンディアンとリトルエンディアンシステムの間での通信をプロトコルを使用せずに行うと問題が生じます
- 解決方法：
使用標準Cライブラリルーチンを使用する
`ntohl()`, `ntohs()`, `htonl()`, `htons()`.

ポインタを使用して数値データのそれぞれのバイトにアクセスする場合

X86/X64 と SPARC マシン上で以下のコードの結果の違いは？

```
int flagVal = 0X01020304;  
char *cptr = (char *)&flagVal;  
printf("Val is %d\n", *cptr);
```

解決方法：

1. 数値データ型より文字データ型を使用
2. Struct (構造体を使用)

アプリケーションデータの格納

- ファイル・システムは一般的に中立しており、ファイルの交換はSolaris OsのSPARC x86/x64 (Intel またはAMD) バージョン間において問題にはなりません。
- アプリケーションが複数プラットフォーム間で共有可能なロー・データを格納する場合、エンディアンに依存します

解決案:

- 1 テキスト・ファイルや文字列を使用して、アプリケーションによって定義づけられた、エンディアンに無関係なフォーマットでデータを格納する
2. ビッグエンディアンまたはリトルエンディアンのいずれかを選択肢、必要に応じてバイト・スワッピング (XDR などのイネーブリング技術の利用が考えられます) を行う

データ格納の順序の違い

☐データの格納の順序は、プラットフォームによって違いがあるため、特定の格納順序を想定したコードは移植できません

```
struct date {
    char year;
    char month;
    char day;
    char dummy; // the struct var size is 4
} planned_date, end_date;
if ( (*(long *)&end_date ) > (*(long *)&planned_date) )
{
    printf("Sorry, You missed the deadline to support Solaris x86 platform \n");
}
```

解決方法：

個々のフィールドを比較してください

読み込み・書き込みの構造

読み込みまたは書き込み時にデータサイズが必要な場合、異なるプラットフォーム上では配列やデータ型のサイズが異なるため、バイト数を指定するために演算子 `sizeof` を使用する必要があります

```
MyRecord myrecord;  
fread(&myrecord, 25, 1, fp);
```

解決策：

```
MyRecord myrecord;  
fread(&myrecord, sizeof(MyRecord), 1, fp);
```

OS の違い

OS(Operating System) の違い

- 32-Bit と 64-Bit の問題
- マルチスレッドプログラミング
- Solaris と POSIX APIs : セマフォの互換性
- C システムコール、ライブラリの違い
- ハードウェアの互換性

32-Bit と 64-Bit

- C 言語仕様

`sizeof(char) <= sizeof(short) <= sizeof(int) <= sizeof(long)`

- 個々のデータ型のサイズは処理環境 (OE) によって使用されるデータモデルで決定されます。

C Data Type	ILP32 Data Model	LP64 Data Model (size in bits)
long	32	64
pointer	32	64
long double	64	128

32-Bit と 64-Bit (続き)

- 32-bit から 64-bit への変換

理想的にはコンパイルのしなおしと 64
ビットライブラリとのリンクのしなおし
だけ

- API が 32-bit と 64-bit ビットの間で異なる場合は `#ifdef` 文でコードの切り替えが必要です

32-Bit と 64-Bit (続き)

Tips

- int 、 long 、 およびポインタデータ型の交換可能な使用はデータ損失を防止するために訂正されるべきです。
- ユニオンと構造体は、データ型のサイズが正しいか確認が必要です
- いつも意図されている結果が出るように、明示的なキャストを用いてください
- 固定の値の変わりに **sizeof()** を使用しててください

マルチスレッドプログラミング

- Solaris のスレッドと POSIX のスレッド
- ほとんどのスレッド APIs は互換性があります

POSIX

`pthread_create()`

`pthread_exit()`

...

Solaris

`thr_create()`

`thr_exit()`

...

セマフォ API

- Solaris の POSIX との互換性

POSIX

sem_destroy()
sem_init()
sem_post()
sem_trywait()
sem_wait()

Solaris

sema_destroy()
sema_init()
sema_post()
sema_trywait()
sema_wait()

C言語のライブラリ、システムコールの違い

- システム・コールには幾つかの違いがあります
- 検証ツール
LinCAT

便利なツール : LinCAT

- LinCAT (Linux Compatibility Assurance Toolkit)
 - > C/C++ ソースファイルを分析するツール
 - > 移植問題を識別するために使用
- LinCAT は以下の情報を出力します
 - > ファンクションコールの名前、位置、および頻度の名前
 - > ソース中で移行の際に問題になりそうなライン
 - > 移行の際の問題を解決することの解決策
 - > 互換性がないコール - ファンクションコールデータベースを元にツールは解決策を示します
- LinCAT は以下から入手できます
<http://developers.sun.com/solaris/downloads/lincat/index.html>

ハードウェアの互換性

- HCL (Hardware Compatibility List) を確認してください。

<http://www.sun.com/bigadmin/hcl/>

- Solaris DDI/DKI はほとんどの命令セットアーキテクチャ (ISA) および多くのバスアーキテクチャ問題からデバイス・ドライバを分離しています。
- X86 デバイス・ドライバは急速に増えています！

Solaris のパッケージ作成

- **Prototype** – パッケージ情報ファイルを作成します
 - > アプリケーションのためのファイルを配置
 - > `pkgproto [path] > prototype`
 - > `pkginfo` – パッケージの情報を記述したファイルを用意
 - > `copyright` – コピーライトファイル作成
- `pkgmk` – インストール用のパッケージを作成
- `pkgchk` – パッケージの整合性のチェック
- `pkgadd/pkgrm` でパッケージのインストールのテスト

まとめ

Solaris 10 へ移行のプロセス

- 使用しているサード・パーティーソフトウェアの確認 / 用意
- 開発環境の確認
- アプリケーションソース上で Solaris と Linux との相違をチェック
- 機能テスト
- パフォーマンスの測定とチューニング
- ソフトウェアのパッケージ作成 / マニュアルのアップデート

ISV 様向けポーティングサポート

- 目的：ISV 様と Sun のマーケットシェアの拡大のための技術サポート
- プロジェクトベースで新規アプリケーションの Solaris 10 への移行をサポート
- 無償！
- Solaris SPARC マシン、x86 マシンが無料でもらえるキャンペーン：“Port Now!” が始まります
 - > 新しいアプリケーションを Solaris 10 へのポーティングして頂くと 1 社に 1 台マシンをプレゼント
 - > 詳しい情報は SDC(Sun Developer Connection) のメーリングリストから送られます

参照

- マニュアル
 - > <http://docs.sun.com>
- Solaris Hardware Compatibility Lists
 - > <http://sun.com/bigadmin/hcl/>
- Solaris Coolstack
 - > <http://cooltools.sunsource.net/coolstack>
- Freeware for Solaris
 - > <http://solaris-freeware.saix.net/index.html>
- SDC (Sun Developer Connection)
 - > <http://sdc.sun.co.jp/portal/index.html>
- Application Packaging Developer's Guide
 - > <http://docs.sun.com/app/docs/doc/817-0406>



Thank you!

サン・マイクロシステムズ株式会社
ISV エンジニアリング部
伊草英雄