

RISCプロセッサを用いたJava実行環境

ー JavaOSを実装したネットワーク・コンピュータ

小森裕之

Java プログラムを動作させる組み込み機器は、CPUとしてJavaチップを使うのが理想といわれている。しかし、現在はまだ量産されていない。また、ターゲット・アプリケーションによっては、汎用CPUにJava VMを搭載したほうがよい場合もある。ここではSPARCliteを使用した評価ボードにJavaOSを実装した例を紹介する。ネットワーク・コンピュータとして動作させた。(編集部)



はじめに

Javaは米国Sun Microsystems社が開発した言語およびその実行環境である。1995年に初版が完成して以来、インターネットを通じて広く世界に普及している。

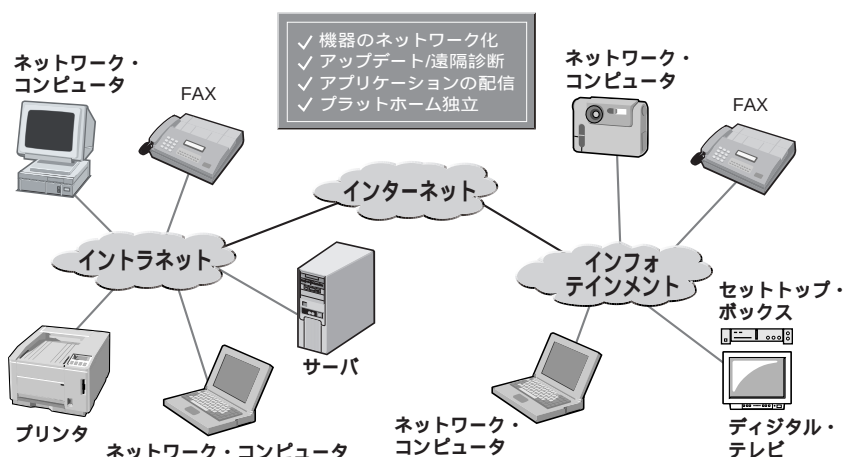
Javaの特徴としては、以下の3点があげられる。

- (1) プラットフォームに依存しない言語仕様
- (2) インターネット/イントラネットとい

った標準的なネットワーク環境でプログラムを利用可能

- (3) 生産性が高いオブジェクト指向言語

こうした特徴は、組み込み機器の開発にもメリットをもたらす。つまり、開発環境の構築・入手が容易になり、分散開発がスムーズに行える。また、ハードウェア変更に伴う移植作業や機能追加による既存処理の手直しなどのバック・ログを解消できる。開発効率が上がり、製品サイクルを短縮できる。



【図1】Javaのビジョン

また、今後の組み込み機器は、情報家電などの分野に見られるように、スタンドアロンよりもネットワークが前提となってきたが、その実現において、Javaは最適な技術と考えられる。たとえば、Javaプログラムをネットワークを介してやりとりすることで、オンデマンド技術による新しいサービスの配信や分散処理といった応用も考えられる。

このような経緯から、現在では組み込み機器分野におけるJavaの重要性が認識され、また要求も高まってきている。

RISCプロセッサへのJavaの搭載

筆者らは、1997年にJavaをネットワーク・コンピューティングの中核技術と位置づけて、さまざまなプラットフォームにおけるJava技術の推進を図ってきた。たとえば、組み込み用32ビットRISCコントローラSPARCliteシリーズへのJava搭載を、Javaチップの開発に先立って開始した。

○組み込みJavaの本命はJavaチップ

組み込み機器分野では、ターゲットとなる製品により、資源が制約される場合や消費電力が仕様に影響するが多い。このため、Javaを実行するCPUとしては、Javaチップが有効だろうといわれている。

たとえば、汎用CPUでJavaを高速に実行するための技術としてJIT(Just In Time)がある。これはJavaのバイト・コードを実行の前に最適化(コンパイル)して、その結果をメモリに格納するため、コンパイルが不要なJavaチップに比べて、余分なメモリが必要になる。また、高い周波数で動作するCPUを使えばJavaプログラムの実行は速くなるが、当然、消費電力が大きくなる。つまり、低消費電力で最大のJava実行性能が出せるのが、Javaチップなのである。

○汎用CPUのJava実行環境の意味

それでは汎用CPUにJavaを搭載する意味はどこにあるのだろうか。

【表1】JavaOSの最小実行環境

OS	CPU	ROM	RAM
JavaOS for NCs	100MHz以上(32ビット)	4M～8Mバイト	4Mバイト (SPARC版で最小8Mバイト、推奨16Mバイト)
JavaOS for Consumers	50MHz以上(32ビット)?	2Mバイト	1Mバイト

一つは、組み込み機器向けにJavaの環境を一番早く提供できる方法だったからである。つまり、組み込み機器でJavaを評価できる環境をとにかく早く構築するには、汎用CPUを使うしかなかった。Javaは組み込み機器の世界では、まだ未成熟な分野である。Javaの利点を理解してもらうことも重要なことだった。

Javaはネットワーク指向の技術であるから、断片的にJavaを採用しても、そのメリットは小さい(図1)。

たとえば、Jiniと呼ばれるサービスがある。Jiniは、ネットワーク上に存在するプログラムおよびデバイスを一つのシステムと見なす考え方である。たとえば、自分のパソコンにデバイスを追加する場合、従来は、デバイスを自分のマシンに接続して、デバイス・ドライバをインストールして、場合によってはシステムを再起動して、やっと使えた。これに対して、Jiniシステムだとネットワーク内のデバイスのリソースにアクセスするだけで使用できる。もちろん、トータル

で見るとメモリ資源の節約にもなる。また、ネットワーク内へのデバイスの追加もホット・プラグインが可能である。ネットワーク機器に応用した場合のメリットが大きいことは容易に想像できる。

しかしながらこれは、Javaでシステムが構築されてこそ効果があるのであって、LANに接続される機器の一部にだけJavaが採用されても意味がないのである。

Jiniと同様なサービスに、CORBA(Common Object Request Broker Architecture)などもある。

また、組み込み分野の中でも資源、電力などの制約がそれほどシビアでない分野がある。たとえば、ネットワーク・コンピュータやPOS、カーナビなどがあげられる。この場合は、汎用CPUのJava環境でも要求を満たせる。しかも、汎用CPU用のOSですでに作成されたシステムを再利用することも可能である。

システム構築の考え方として、Java中心の処理系にするならば、Javaチップ

(picoJava)が最適である。しかしリアルタイムOSを使った過去の資産があり、その拡張部分だけをJavaで構築するような処理系では、汎用プロセッサに頼らざるを得ない場合もあるのが実状である。

Javaのプラットフォーム

Java実行環境(JavaOS)の実現において、ターゲットに要求されるスペックをまとめたのが表1である。

○JavaOS for NCs

JavaOS for NCsは、Java APIのほとんどの機能をもつOSで、ネットワーク・コンピュータやPOS、カーナビなどの高いスペックをもつ機器をターゲットにしている(図2)。

ROMについては、実際のOSのサイズが8Mバイト近くであるため、OSのダイエットをしない限り、この程度のサイズは必要になる(ネットワークからのOSブートはこの限りではない)。

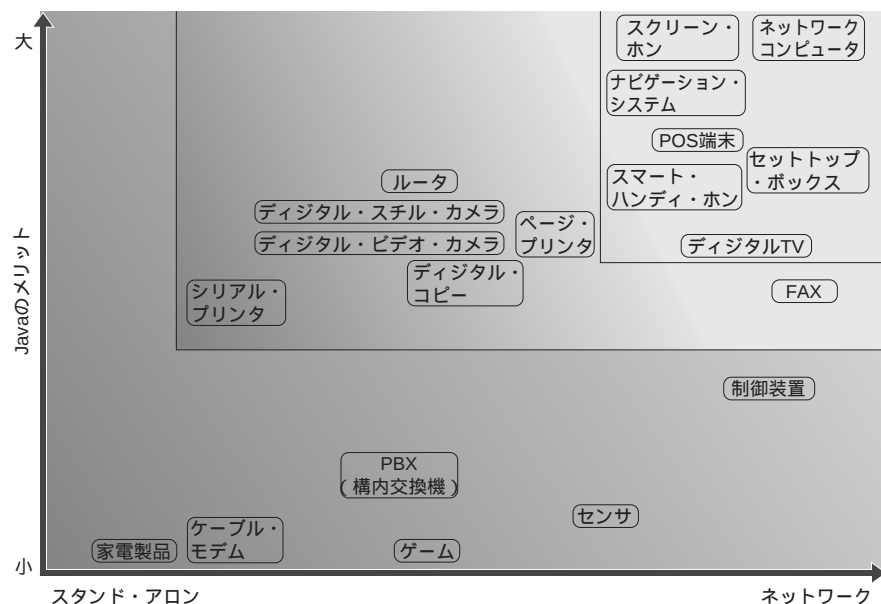
また、JavaOS for NCsは、ハード・ディスクなどのローカル・ストレージ・デバイスを前提にしていないので、ファイル・システムとしてNFSを使うことになる。ネットワークを使用しない環境では、JavaプログラムをROM(またはフラッシュなどの不揮発性メモリ)に格納する必要がある。

RAMについては、16Mバイトで問題ない。しかし、ダウンロード・データのメモリ・キャッシュやJITの作業メモリとしても使用されるため、運用面を考慮に入れてサイズを決める必要があるだろう。

JavaOS for NCsを搭載したネットワーク・コンピュータの例としては、Sun Microsystems社の「JavaStation」がある。

○JavaOS for Business

JavaOS for Businessは、Java言語仕様のフルスペックをもつOSである。JavaOS for NCsとは、マイクロカーネル部分などが異なる。プラットフォームとして、JavaOS for NCsと同程度のスぺ



【図2】Javaのターゲット