PC へのリアルタイム JavaVM 実装検討

二村 祐地
3D−5
三菱電機 (株) 情報技術総合研究所

1. はじめに

制御システムでは標準化やオープン化，低価格化への強い要求から，パソコン (PC) 及びインターネット技術の適用が近年急速に進んでいる．Java 言語は，オブジェクト指向，スレッド /GUI/ ネットワークの容易な操作，高いプログラム開発効率など，制御システム構築に適した特徴を持つ．しかし現状 Java 実行環境の多くは処理速度や定応答性が十分でなく，その適用は対人領域に限られている．

本稿では，汎用 PC 上で Windows アプリケーションと共存しながら，制御システムの基幹処理に向けたリアルタイム性の高い Java 実行環境を実現するリアルタイム Java VM(RT-Java VM) の実装について，その検討結果を報告する．

2. 前提

2.1 システムモデル

本検討の対象システムを図 1 に示す．制御対象と直接ないし制御 LAN で接続した汎用 PC 上で，PLC(Programmable Logic Controller) 相当の制御処理を行う高リアルタイムの Java プログラムと，その実行環境である RT-Java VM，並びに GUI や帳票処理などを行う緩リアルタイムの Java あるいは Windows アプリケーションと，それらの実行環境である Java VM 及び Windows NT が動作する．

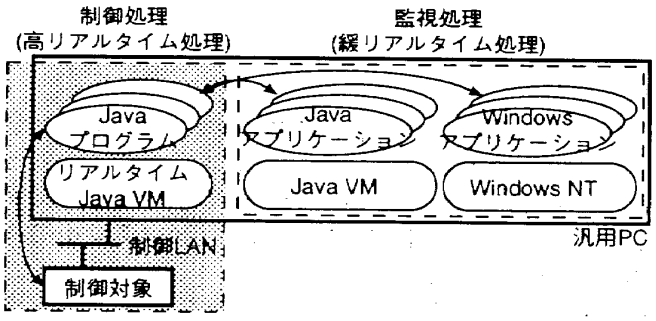

図 1：システムモデル

On an Implementation of Real-Time Java VM for PCs

Yuji Nimura
Mitsubishi Electric Corporation
Infomation Technology R&D Center
5-1-1 Oofuna, Kamakura, Kanagawa 247-8501, Japan

2.2 目標

PLC で行う制御システムの基幹処理は，一般に数ミリ秒から十数ミリ秒の定周期処理である．また Java 実行環境としては一般的機能は全てサポートすることが望ましい．よって RT-Java VM の基本目標を以下の通り定める．

1. 制御処理の 10 ミリ秒定周期動作実現
2. Java プラットフォーム core API の提供

3. 検討

3.1 実装方法

Windows NT 上のアプリケーションとしてプログラムを動作させた場合，10 ミリ秒の定周期処理は十分な定応答性が得られず実現困難である [1]．一方本検討で実現を図る制御処理は，その内容に以下の特徴がある．

1. 制御データの入出力と演算には高い定周期性 / 定応答性が要求される．一方ファイルやネットワーク，GUI に対するリアルタイム性の要求は緩く，WindowsNT 上でも達成可能である．
2. Windows アプリケーションとの円滑な協調のために，Windows NT 管理下のファイルや GUI を容易に扱える必要がある．逆に Windows NT とは独立したファイルや GUI はほとんど必要ない．

これより RT-Java VM は，以下のように定周期性 / 定応答性が重要な高リアルタイム部とそうでない緩リアルタイム部に分割して，その実装を図る．

- 高リアルタイム部
  バイトコード実行部と制御データの入出力及び演算処理が該当．これらは Windows NT からの動作干渉を避けるように実装し，さらに必要に応じ定応答性向上の対策を取る．
- 緩リアルタイム部
  ファイルやネットワーク，GUI などが該当．これらは Windows NT の機能を用いて実装する．その際，高リアルタイム部の定応答性を阻害しないよう配慮する．

本検討によるRT-Java VMの構造を図2に示す．

1. バイトコード実行部はWindows NTのドライバ層で実装する．なお実装には，別途開発を進めている汎用PCへのリアルタイム制御機構[1][2]を活用する．
2. ファイル/ネットワーク/GUIの各処理は，Windows NT上のサポートプログラム(サポータ)を経由してWindows NTに依頼する．

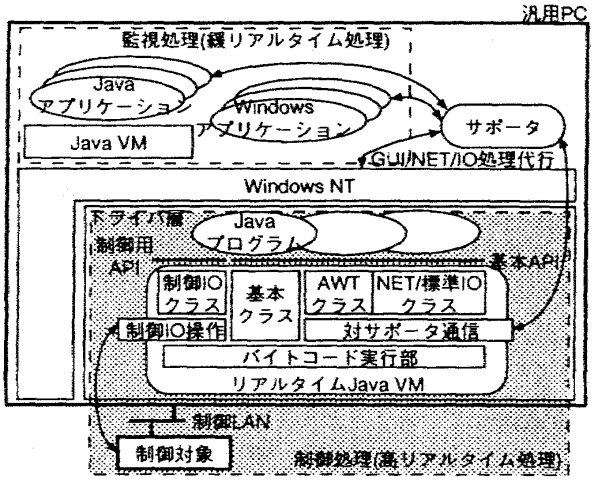

図2：汎用PCでのリアルタイムJava VM構造

3.2 定応答性対策

既存のJava VM(JDK1.0.2, HP-UX 10.20, PA-7200, 100MHz)にて，各種動作がJavaスレッド間切替え時間に及ぼす影響を調査したところ，以下の要因により10ミリ秒を超える乱れを確認した．

1. ファイル書き込み等OS内部での長時間動作
2. メモリのガベージコレクション処理(GC)

OS内部の長時間動作への対策

第1の状況はJavaスレッドがOS内で逐次的に動作するためであり，一般にはJavaスレッドをOS内でも別々のスレッドとして動作させること(ネイティブスレッド化)で対処する．しかし本RT-Java VMは，バイトコード実行部はWindows NTのドライバ層で動作させ，OS(Windows NT)のサービス呼出しはアプリケーションレベル(サポータ)経由で行う，と特殊な実装をとっている．このため本件は次の形で対処を図る．

- サポータへの処理依頼はどのJavaプログラムより低優先とし，バイトコード実行部は他に動作可能なJavaスレッドを優先して実行する．
- 制御データの入出力はドライバ層内で処理し，Javaスレッド間の切替え時間への影響は極力抑えるよう実装する．

GC処理への対策

第2の状況はGC処理中にスレッド切替えができないためであり，一般にはincremental GCの適用で対処する[3]．ただし本RT-Java VMに要求される定応答性は制御プログラムの定周期性確保であるため，GCにも周期的な動作が許容でき，例えばgenerational GCなどより粒度の粗い技法も適用の余地がある．他方，制御システムとしては，可用性確保の面からメモリ資源の枯渇を回避すべくメモリリークやフラグメンテーション対策が必要であり，また制御処理とGCの負荷バランスも重要である．よって本件は，定応答性，可用性，処理負荷の各視点を踏まえた試作評価により対処を図る．

4. おわりに

本稿では汎用PC上でWindowsアプリケーションと共存可能なリアルタイムJava VMの実装に関する検討結果を述べた．今後はさらに以下の点も踏まえ，有用性の実証を進める予定である．

- リアルタイム処理API
  Javaプラットフォーム core APIにはリアルタイム処理の記述に不備/不足がある[4][5]．システム構築の容易さは実システム展開上重要な要素であるので，標準化動向を踏まえつつ適切なモデル及びAPIを検討する．
- 制御用入出力装置API
  制御データの入出力操作や制御対象の事象変化を契機とする処理の起動を，容易かつ統一的に扱えるようにするため，制御用入出力装置のモデル及びAPIを検討する．

参考文献

[1] 川上，片山，黒澤:“汎用PCに置けるリアルタイム制御機構の実現(その1)”，情報処理学会第51回全国大会，1997

[2] 片山，川上，黒澤:“汎用PCに置けるリアルタイム制御機構の実現(その2)”，情報処理学会第51回全国大会，1997

[3] Paul R. Wilson:“Uniprocessor Garbage Collection Techniques”, ftp://ftp.cs.utexas.edu/pub/garbage/bigsurv.ps,1994

[4] Kelvin Nilsen:“Development of Partable Real-Time Software in Java”, http://www.newmonics.com/webroot/technologies/java/intranet.slides.ps,1996

[5] 三好，喜多山，徳田:“Javaの Real-Time 拡張”，第75回システムソフトウェアとオペレーティングシステム研究会，June 1997