

Performance of the test control system using Java/CORBA

Isao Ito^{A)}, Norio Nakamura^{A)}, Hiroyuki Takaki^{A)}, Hiroshi Sakai^{A)}, Takashi Shibuya^{A)}, Kenji Shinoe^{A)},
Hirofumi Kudo^{A)}, Atushi Ishii^{A)}, Toshikatu Masuoka^{B)}

^{A)} Synchrotron Radiation Laboratory, Institute for Solid State Physics, the University of Tokyo

5-1-5 Kashiwanoha, Kashiwa, Chiba, 277-8581

^{B)} NICHIZO Electronic & Control Corporation

5-3-28 Nishikujo, Konohana, Osaka, 554-0012

Abstract

In order to research the control system for “Super SOR”, we have developed the test bench of the control system and evaluated its performance. We use CORBA (Common Object Request Broker Architecture) as the framework for the control system, and Java as the programming language for the graphical user interface. This paper describes the test bench and its performance.

Java/CORBAを用いた制御系テストベンチのパフォーマンス

1. はじめに

現在、東京大学では極紫外・軟X線高輝度光源施設 “Super SOR” を東京大学柏キャンパスに建設する計画を推進している。Super SORはリニアック(20m, 200MeV)、ブースターシンクロトロン(周長90m, 1.8GeV)、光源リング(周長280m, 1.8GeV)から成り、制御対象は光源リングの電磁石だけで284点にのぼる^[1]。

加速器運転を管理する制御系には、広範囲に分散設置された多種多様な加速器構成機器郡を統一かつ効率よく制御・監視し、長期間連続運転を可能とすることが求められる。さらに制御系の構成技術が一般の加速器構成機器に比べて進歩の激しい計算機やネットワーク技術に依存している部分が多く、一時的な流行に惑わされない十分な機能を持ち、寿命の長いシステムを構成することが求められる。

現在、世界各国の加速器施設でさまざまな通信・情報技術が加速器制御系のフレームワークに利用されているが^[2,3]、我々はCORBAというミドルウェアをネットワーク層にラッピングする方法を検討している。CORBAとはCommon Object Request Broker Architectureの略であり、分散環境におけるオブジェクト間のやり取りを実現するメカニズムの標準化設計思想である^[4]。CORBAに実装されているミドルウェアORB(Object Request Broker, オープ)が仲介となり、分散設置されたオブジェクト同士がセッションを確立する。図1はCORBAで加速器構成機器を制御する場合の模式図である。加速器構成機器の制御プログラム(オブジェクト)が分散設置された操作作用計算機(PCなど)、ローカルコントローラ(VMEなど)のメモリに常駐し、オブジェクト同士がCORBAを介した制御コマンドのやり取りを行うことで加速器制御を実現する。

CORBAの主な利点としては、オブジェクトに動作環境、開発環境の依存性がなくなること、制御機

器の分散オブジェクト環境を提供することでタスクを分散して制御機器1台に対する負荷・リスクを軽減し処理の高速化を実現することが挙げられる。CORBAで構築された制御系は多種多様な加速器構成機器を一括して管理・制御し、機器の更新に柔軟に対応する寿命の長いシステムに成りうる。

一方で、抽象化層(CORBA層)が挿入されているためsocket通信やRPCなどのCORBAよりも低レベルの通信に比べて通信が重く遅いことや、今までの通信方法に比べてより多くのCPUパワーとメモリが必要になることなどの欠点がある。そこで、通信層にCORBA層を挿入したことの影響がどの程度あるのか評価する必要がある。今回、VMEとPLCをCORBAで制御するテストベンチを構築した。このテストベンチでステータス収集に要する時間を測定し、通信速度に対するCORBA層の影響を調べる。

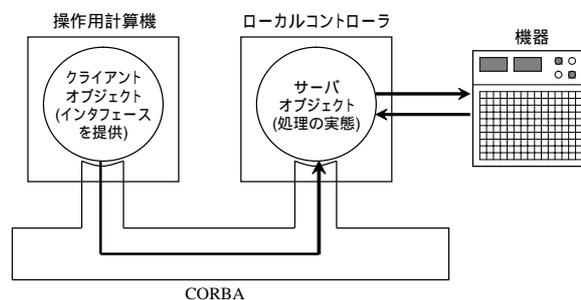


図1. CORBAによる加速器構成機器の制御

2. テストベンチの構成

2.1 ハードウェア構成

図2はテストベンチのハードウェア構成である。Super SORで使われるローカルコントローラはVMEとPLCであり、テストベンチは我々が目指すシステムの最小単位である。VMEは電磁石電源を制御し、

PLCは真空計のコントローラを制御する。操作ログや機器のステータスはデータベースサーバに記録される。各機器とのインタフェースは操作用PCで提供される。各機器はLANに接続されていて、TCP/IPで通信を行う。イーサネットとTCP/IPを基本としたネットワークは、通信が十分高速であるだけでなく、現在もっとも標準的であるため、メンテナンス費が安価に済み、次世代の標準ネットワークでも継続性が期待できる。

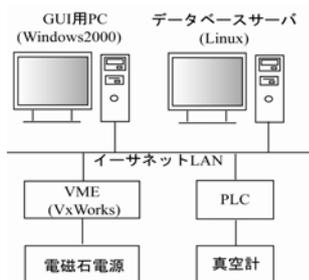


図2.ハードウェア構成

2.2 ソフトウェア構成

表1はテストベンチのソフトウェア環境である。GUIなどの操作画面の開発には、将来の計算機の更新を考慮して制御プログラムが様々な処理系で動作することを重視し、開発言語としてJava言語を選択した。Javaはオブジェクト指向言語の一つで、機種依存性の非常に少ない言語であり、多くの処理系に対応したプログラムを開発できる。電源制御プログラム、真空度収集プログラム、ロガー、ポーラには開発言語としてC/C++を採用した。PCのOSにはGUI作成のアプリケーションが豊富なWindows系OS(Windows2000)を採用した。データベースのOSには信頼性と安定性を必要とすることからUNIX系OS (Red Hat Linux)を採用した。データベース管理システムにはソースコードが公開されていて、いろいろな処理系で動作するMySQLを採用した。

機器名称	項目	名称
操作用PC	OS	Window2000
	Java	Java 1.4
	JDBC	mysql-connector-java
	ODBC	MySQL ODBC
	CORBA	ORBacus4.1.0
	開発言語	真空収集：VC++ 操作画面：Java
VME	OS	VxWorks5.4
	CORBA	ACE5.2 TAO1.2.1
	開発言語	C/C++
データベースサーバ	OS	Red Hat Linux (Kernel 2.4.20-8)
	データベース	MySQL

表1.ソフトウェア環境

図3はテストベンチのソフトウェア構成である。制御コマンドの通信にはCORBAを利用している。電磁石電源を操作する場合、PCから送られてきた制御コマンドをVMEの電源オブジェクトが受け取る。電源オブジェクトは操作コマンドに従って電流値設定及びステータスの入出力を行い、取得したステータスをPCに返す。操作ログはPCのロガーを介してデータベースに保存される。電源のステータスはポーラから定期的に参照されデータベースに保存される。ロガーからデータベースへのアクセスにはODBC(Open DataBase Connectivity、データベースにアクセスするためのソフトウェアの標準仕様)を使用する。操作用PCからデータベースを検索する時はJDBC(Java DataBase Connectivity、JavaプログラムからリレーショナルデータベースにアクセスするためのAPI)を使用する。

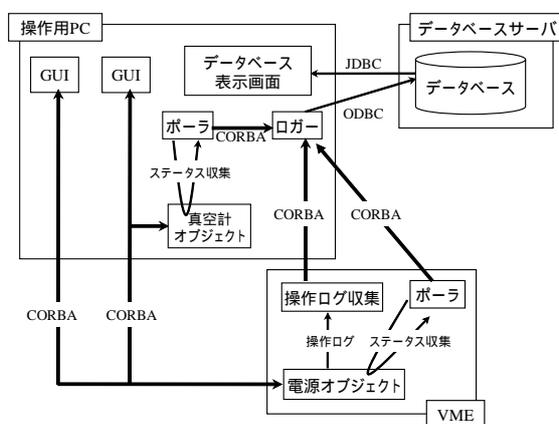


図3.ソフトウェア構成

3. パフォーマンス測定

図4は測定のセットアップである。PCから電源オブジェクトへステータス収集のコマンドを出したときの時間 T_i と、電源オブジェクトからPCにスタート情報が送られてきたときの時間 T_f をPC上で測定する。 T_f と T_i の差 T_{res} をPCが電源オブジェクトを呼び出すのに要した時間と定義する。以下では応答時間と呼ぶ。VME上では実際の電源を制御する電源オブジェクト以外にシミュレート用電源オブジェクトを

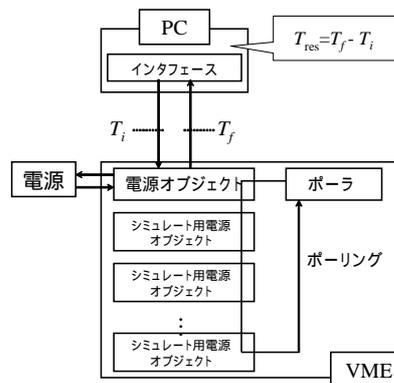


図4.セットアップ

走らせている。このシミュレート用電源オブジェクトは適当な電源ステータスを作るタスクで、電源オブジェクトをよく再現する。複数個のシミュレート用電源オブジェクトをVME上で走らせることでVMEが複数の電源を制御する状況を作り出す。ポーラは電源オブジェクトのステータスを一括収集しロガーに情報を送信した後、一定期間（ポーラ周期）休止する。ポーラ周期は1/60秒まで小さくすることができる。

図5は電源オブジェクトとの通信を1000回行ったときの平均応答時間である。PCの内部時間測定精度は1ミリ秒であり、これにより通信を1000回行った時の時間測定精度は1マイクロ秒となる。Super SORで想定される環境（電源オブジェクト数が100）の場合、10秒のポーラ周期に対する平均応答時間は約1.6ミリ秒であった。この状況でポーラ周期を小さくすると平均応答時間は若干増加するが、20ミリ秒のポーラ周期でも平均応答時間は2ミリ秒以下であり、問題はない。電源オブジェクトの数を10倍にしても平均応答時間に大きな変化はなかった。

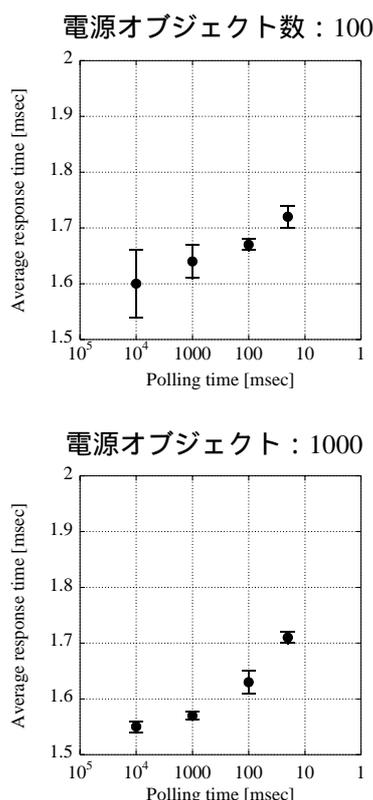


図5. 電源オブジェクトとの通信を1000回行ったときの平均応答時間

図6は電源オブジェクトの呼び出しイベントの応答時間をヒストグラムで表した図である。1イベントごとの応答時間をプロットしている。電源オブジェクト数は100である。測定精度は1ミリ秒である。呼び出しイベントの応答時間はたかだか20ミリ秒以下で、そのほとんどが1ミリ秒以下だった。

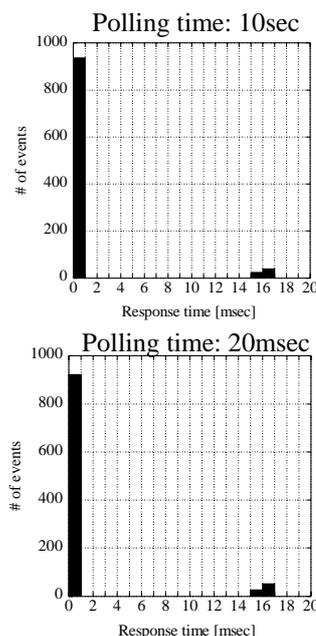


図6：電源オブジェクトの呼び出しイベントの応答時間分布

4. まとめ

CORBAを用いた制御系の応答時間を評価した。電源オブジェクトの呼び出しイベントに要した平均時間（平均応答時間）を電源オブジェクト数とポーラ周期を変えて測定した。Super SORで想定される環境（電源オブジェクト数100）について、10秒のポーラ周期に対する平均応答時間は約1.6ミリ秒だった。さらにポーラ周期を20ミリ秒（1/60秒）にしても、平均応答時間は2ミリ秒以下だった。電源オブジェクトの数を10倍に増やした環境（1000）でも、平均応答時間は100の時と比べて大きく変化しなかった。

Super SORで想定される環境（電源オブジェクト数100）で呼び出しイベントの応答時間分布を測定した。応答時間はたかだか20ミリ秒以下で、そのほとんどが1ミリ秒以下であり、問題になる応答時間の遅れはなかった。

5. 今後の課題

PLCおよびデータベースとの通信時間を測定し、テストベンチを総合的に評価する。

参考文献

- [1] N. Nakamura, "Design of the Super-SOR Light Source", Proc. of the European Particle Accelerator Conference 04, Lucerne, 2004, 2347.
- [2] L. Dalesui et al., "The Experimental Physics and Industrial Control System Architecture: Past, Present, Future", Nucl. Instrum. Meth. A 352(1994)179.
- [3] <http://www.esrf.fr/tango>
- [4] <http://www.omg.org>