

DAQ-Middleware を用いた動的制御機構の提案

前田 寛幸[†] 長坂 康史[†] 千代 浩司[‡] 味村 周平^{†‡} 神徳 徹雄^{‡‡}

広島工業大学大学院工学系研究科[†] 高エネルギー加速器研究機構[‡]

大阪大学^{†‡} 産業技術総合研究所^{‡‡}

1. はじめに

近年、ネットワーク技術の発達に伴い、ネットワークを介して様々なデータがやり取りされるようになった。物理学実験に対するデータ収集 (DAQ) システムにおいても、これらのネットワーク技術を利用することで、大容量の測定データを高速に収集し、保存することが可能となってきた。しかし、同規模の実験であっても、それぞれの実験用にデータ収集システムを開発することが多く、開発者の負担も大きくなってきている。

本研究室では、これらの負担を軽減する目的で、データ収集システムの汎用フレームワークとして、オブジェクトベースの DAQ-Middleware を他研究機関と共同で開発している。このフレームワークは、システムを制御する Operator 側とデータ処理を行う Component 側に分かれており、設定ファイルを用いることで各 Component に測定装置の初期値などを設定できる。しかし、現在の DAQ-Middleware では、データ収集実行時に Component へパラメータを送信する機構が備わっていない。

そこで本研究では、Component のパラメータを動的に変更する機構の開発を目的とする。

2. DAQ-Middleware

DAQ-Middleware とは、ネットワーク分散環境でデータ収集ソフトウェアを容易に構築するためのソフトウェア・フレームワークである [1]-[3]。

この DAQ-Middleware は、Robot-Technology ミドルウェアをベースにしており、データ通信を行うそれぞれの Component 間では、オブジェクト通信を行っている。また、このフレームワークでは様々な Component を組み合わせることができ、ネットワーク分散 Component 技術によりユーザが求めるデータ収集システムを容易に開

発することが可能となる。

図1に DAQ-Middleware の基本構成を示す。図1の PC はそれぞれ、ユーザインターフェイス用計算機 (UI) と DAQ 用計算機 (DAQ#1) を示している。ユーザは DAQ-Operator を用いて、システムの制御を行う。その際に XML で記述されたファイルを読み込み、パラメータや Component などの情報を設定する。そして、各 DAQ-Component に制御コマンドを送信することで全体のシステムを動作させることが可能となる。図1の PC (DAQ#1) では、基本 DAQ-Component である Gatherer, Dispatcher, Logger, そして、Monitor が DAQ-Operator からのコマンドにより動作する。例えば、Logger は Dispatcher から送信された情報をデータベースに格納する役割を担っている。これら DAQ-Component 間の通信はネットワーク透過であるため、一つひとつの Component を別々の計算機上で動作させることも可能である。

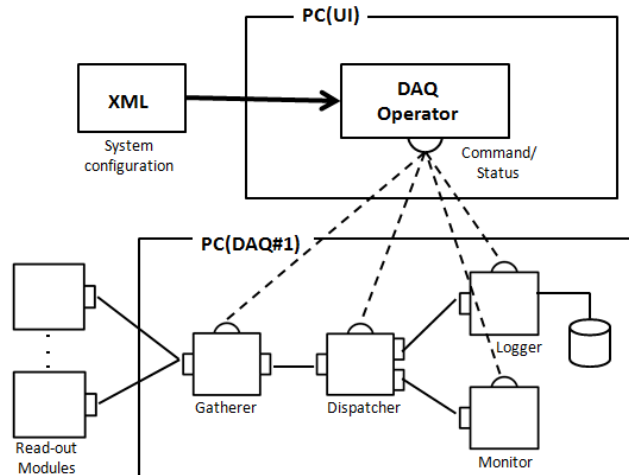


図1 DAQ-Middleware の基本構成

3. 提案方式

3.1 概要

DAQ-Middleware は、状態遷移によってシステムを制御するフレームワークである。図2に実線で従来の DAQ-Middleware が持つ4つのステー

Research on a Dynamic Control Mechanism for
Daq-Middleware

[†]HiroYuki Maeda [†]Yasushi Nagasaka [‡]Hiroshi Sendai

^{†‡}Shuhei Ajimura ^{‡‡}Tetsuo Kotoku

[†]Hiroshima Institute of Technology

[‡]High Energy Accelerator Research Organization

トと点線で新たに追加したステートを示す. 各 Component は, Operator からの制御コマンドを受信することで状態遷移を行う. 例えば, システムが Running 状態の場合, Operator 側から Pause コマンドを各 Component に送信するとシステムは Paused 状態に移行する.

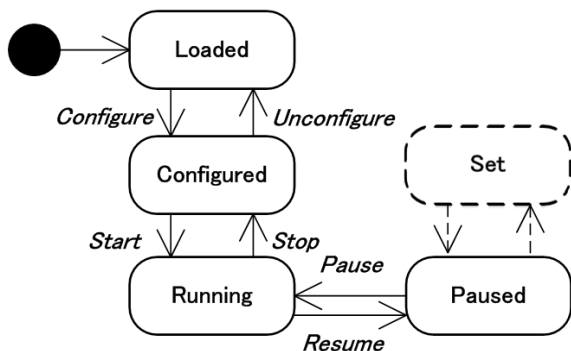


図2 DAQ-Middleware の状態遷移図

3.2 実装

本研究では, パラメータを動的に変更するために新たに Set ステートを追加した. パラメータの動的変更には, 複数の XML ファイルを使用する. これらのファイルには異なるパラメータが設定されており, ユーザは Paused 状態から追加ステートに遷移する際に任意のパラメータを動的に選択することが可能となっている. パラメータは名前と値という組のリストにして各 Component へ送信する. 各 Component では送信されたリストから名前をキーにして値を取得するため, パラメータの名前は XML ファイル内でユニークにする必要がある.

4. 性能評価

4.1 セットアップ

提案システムの性能を評価するため, 設定ファイルを選択してから, Component にパラメータが反映されるまでの実行時間を測定した. 測定には, 2台の同型 PC を使用した. 表1にその PC のスペックを示す. 測定の際に使用したパラメータのデータサイズは, 2 bytes~1024 bytes のバイナリファイルである. 各データサイズの実行時間は, 50 回の測定結果を平均したものである.

表1 PC スペック

OS	Scientific Linux 6.2
CPU	Intel(R) Xeon(R) CPU E5606 2.13 GHz
Memory	2048 MBytes
NIC	Intel Corporation 82598EB 10-Gigabit AT2 Server Adapter

4.2 結果と考察

測定結果を図3に示す. この図は, データサイズにおける実行時間の変化を表している. 例えば, データサイズが 1024 bytes の場合, 実行時間は約 6 ms となった. 従来の DAQ-Middleware では, パラメータを変更するためにシステムを停止させ, 再起動する必要があった. 例えば図1のような基本的なシステム構成の場合, 再起動には約 5 sec 必要となる. よって, 本システムはパラメータ変更に必要な時間を大幅に削減できたと考える.

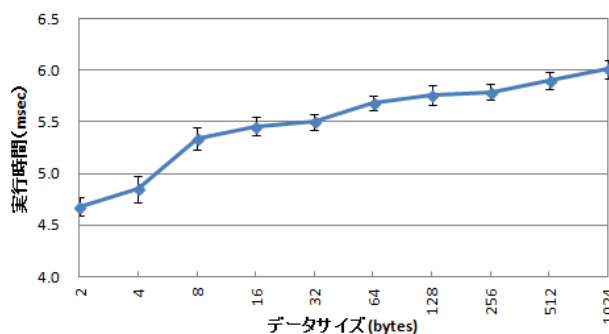


図3 データサイズにおける実行時間の変化

5. おわりに

本研究では, パラメータを動的に変更する機構を付加することでより効率の良いシステムにすることを目的とした. そこで, 新たに追加したステート上で設定ファイルを選択し, 再度読み込むことでシステムを停止させずにパラメータを動的に変更することができる機構を開発した. また, 設定ファイルを選択してから, Component にパラメータが反映されるまでの実行時間を測定した. データサイズが 1024 bytes の時, 実行時間が約 6 ms となった. これらの結果から, 本システムはパラメータ変更に必要な時間を大幅に削減できたと考える.

今後, コマンドラインからの入力など様々なニーズに合うパラメータ動的制御機構を実装することで, DAQ-Middleware を柔軟なフレームワークにしていく.

参考文献

- [1]Y. Yasu, K. Nakayoshi, E. Inoue, H. Sendai, H. Fujii, N. Ando, T. Kotoku, *et al.*, "A Data Acquisition Middleware," in *Proc. IEEE/NPSS Real Time Conference*, pp. 1-3, May 2007.
- [2]Y. Yasu, K. Nakayoshi, H. Sendai, and E. Inoue, "Functionality of DAQ-Middleware," *IEEE Trans. Nucl. Sci.*, vol. 57, no. 2, pp. 487-490, Apr. 2010.
- [3]K. Nakayoshi, H. Sendai, Y. Yasu, E. Inoue, T. Kotoku, N. Ando, Y. Nagasaka, S. Ajimura and M. Wada, "DAQ-Middleware: Progress and status," *J. Phys.: Conf., Ser.* 331 02202318, 2011