

仮想化制御コントローラの系統切替手法の検討 Research on Application of Virtualization Technology to Controllers

川嶋大樹[†] 舟橋聡[†]
Hiroki Kawashima Satoru Funahashi

1. はじめに

仮想化技術は、ハードウェアを抽象化し、論理的に扱う技術である。この技術により、ハードウェアとソフトウェアを切り離すことができる。プラント、交通システム、物流運搬、船舶海洋等に関連する製品の制御コントローラに仮想化技術を適用することで、一つのハードウェアに複数の制御コントローラを集約することができるため、ハードウェアコストの削減も可能となり、大きなコストダウンが可能となる。

制御コントローラは、高い可用性が必要であるため、冗長化構成を採っている。仮想化技術を適用した制御コントローラにおいても、異常発生時には稼働側から待機側に速やかに切り替わり、リアルタイムに正常な制御信号を制御対象へ出力することが求められる。

本稿では、仮想化制御コントローラの冗長化構成におけるリアルタイム性を獲得するために、①仮想化制御コントローラ単体のリアルタイム性の向上、②仮想化制御コントローラにおける系統切替手法の検討を行った。

2. 仮想化制御コントローラの系統切替

2.1 基本構成

本検討では、KVM(Kernel-based Virtual Machine) [1] を用いて、図1、表1に示す仮想化制御コントローラの冗長化構成を構築した。仮想化制御コントローラにて定周期制御信号送信プログラム(50msec周期)を実行し、制御対象に制御信号を送信する。

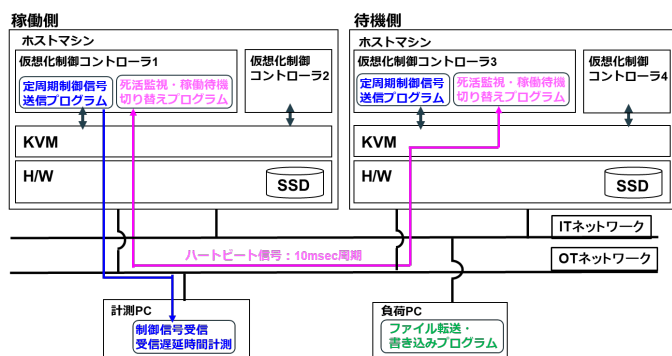


図1 基本構成

2.2 仮想化制御コントローラ単体のリアルタイム性能向上

制御コントローラは、制御対象へリアルタイムに信号を送信する必要があるが、仮想化技術を適用した場合、仮想

[†]三菱重工業株式会社 ICTソリューション本部
ICT Solution Headquarters,
Mitsubishi Heavy Industries, Ltd.

表1 評価環境

ホストマシン	カーネル	Linux 5.4.54
	PREEMPT-RT patch	5.4.54-rt33
	OS	Ubuntu 18.04 64bit
	CPU	Core i5-9500vPro 3GHz
	メモリ	8GByte
ゲストマシン (仮想化制御 コントローラ)	カーネル	Linux 3.12.74
	PREEMPT-RT patch	3.12.74-rt99
	OS	Ubuntu 14.04 32bit
	メモリ	2GByte

化特有の処理遅延が発生し、リアルタイム性の低下が考えられる。リアルタイム性を向上させる手法として以下(1)~(3)の手法を適用し、仮想化制御コントローラ単体のリアルタイム性の向上を図った。

(1) 排他制御の変更によるタスク切替オーバーヘッド低減

リアルタイム性を確保するためには、実行されているタスクよりも優先度が高いタスクが起床する際、遅延なく速やかに実行できることが必要である。通常のカーネルでは排他制御を担うスピンロックによりプリエンプション禁止時間(オーバーヘッド)が発生する。そこで、カーネルにPREEMPT_RT patch [2,3]を適用することでプリエンティブな排他制御を行い、タスク切替時のオーバーヘッド低減を行った。

(2) ハードウェアエミュレータ(QEMU)のスケジューリング優先度変更による処理遅延低減

仮想化制御コントローラはCPUや通信インターフェースなどのハードウェアリソースにアクセスする際、ホストマシン上で動作するハードウェアエミュレータ(QEMU)を介して処理する。ハードウェアエミュレータ実行に待ち時間が発生すると仮想化制御コントローラの処理も遅延する。そこで、ハードウェアエミュレータの優先度をリアルタイム性が要求されない他タスクよりも高優先度に設定し、待ち時間低減を図った。

(3) CPUの静的割付によるリソース競合回避

仮想化制御コントローラがCPUにアクセスする際に同じCPUコアを他が使用した場合(競合)、待機による処理遅延が発生する。そこで、各仮想化制御コントローラが使用するCPUコアを割り付けることで競合を回避し、処理遅延を解消した。

2.3 系統切替手法

二重化した仮想化制御コントローラが、互いに死活をチェックするためのハートビート信号をやり取りし、相手が停止したことを検知した際に自分が稼働状態となる方法を採用した。死活監視のための専用線を用いずに実現するた

め、OT (Operational Technology) ネットワークを介して、10msec 間隔でハートビート信号を通信する。待機側がハートビート信号を3回送信する間に、稼働側のハートビートを受信しない場合、待機から稼働へ切り替える。

3. 評価

3.1 定周期制御信号の遅延評価

3.1.1 評価方法

想定している制御システムでは、制御対象が制御コントローラからの信号の受信間隔を計測し、受信間隔がタイムアウト時間（制御周期の2倍）を超えた場合に制御信号が途絶したと判断し、異常として検知する。そこで本評価では、制御対象で定周期制御信号の受信間隔を計測し、制御周期との差を”受信遅延”として評価した(図2)。計測時は、システム負荷をかけるためファイル転送・書き込みの処理負荷を印加した。

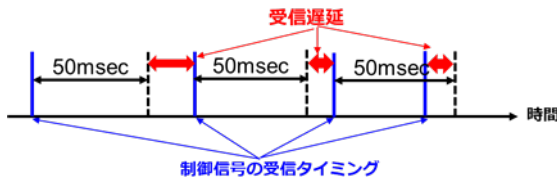


図2 受信遅延

3.1.2 評価結果

定周期制御信号を144万回（20時間）制御対象に送信し、リアルタイム性向上手法適用前後の受信遅延時間を計測した(図3)。結果、最大受信遅延時間が29%低減できた。このことから、リアルタイム性向上手法が有効であることが分かった。

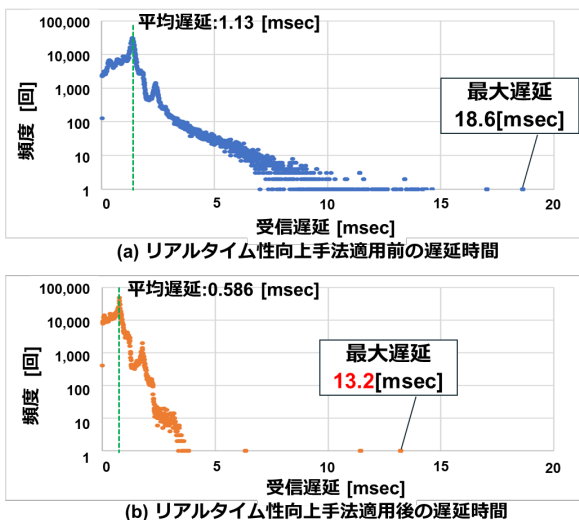


図3 受信遅延分布 144万回測定（20時間）

3.2 稼働・待機切替時間の評価

3.2.1 評価方法

二重化ホットスタンバイ環境における稼働待機切替時間を評価した。制御対象で受信する制御信号の出力先が切り替わる直前と直後の受信間隔を稼働・待機切替時間とする(図4)。

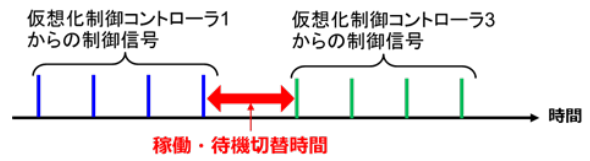


図4 稼働・待機切り替え時間

想定している制御システムでは、制御信号途絶の異常検知タイムアウト値が制御周期(本稿では50msec)の2倍であることから、稼働・待機切替時間の目標値を100msec以下とした。

負荷条件として、定周期制御信号送信プログラムが動作する仮想化制御コントローラにファイル転送・書き込みの処理負荷を印加した。

3.2.2 評価結果

仮想化制御コントローラ1が稼働状態の際に、①仮想化制御コントローラのシャットダウン、②ホストマシンのシャットダウンの2つ切替条件を発生させ、稼働・待機切替時間を計測した。その結果、両条件において目標値100msec以下であることを確認し、仮想環境化においても要求時間以内に切り替えられることが分かった。

表2 稼働・待機切替時間 計測結果

切替条件	稼働・待機切替時間 最大値 [msec]
仮想化制御コントローラ1のシャットダウン	72.5
ホストマシンのシャットダウン	81.3

4. まとめ

制御コントローラのハードウェアコスト削減を狙いとして、仮想化制御コントローラの冗長化構成に適用可能な、OTネットワークを介したハートビート信号に基づく系統切替手法を検討した。考案手法のリアルタイム性評価システムを構築し、稼働待機切替時間を計測した結果、最大が81.3msecとなり、目標値100msec以下であることを確認した。このことから、仮想化制御コントローラにおいても制御コントローラが要求する系統切替時のリアルタイム性を満足できることが分かった。

5. 今後の課題

定周期制御信号の遅延評価について、最大遅延時間は低減したが、遅延時間分布から見て、突出している。リアルタイム性を保証するためには、突出した遅延の原因究明と改善が必要である。今後、ホストマシンとのリソース競合や通信インターフェース処理部分の影響についても考慮し、さらなるリアルタイム性の向上方法を検討していく予定である。

参考文献

- [1] Kernel Based Virtual Machine
<http://www.linux-kvm.org>
- [2] Reghenzani, Federico, Giuseppe Massari, and William Fornaciari. "The real-time linux kernel: A survey on PREEMPT_RT." ACM Computing Surveys (CSUR) 52.1 (2019):
- [3] Kiszka, Jan. "Towards linux as a real-time hypervisor." Proceedings of the 11th Real-Time Linux Workshop. 2009.