

センサ・アクチュエータノードにおけるプロセスの目的を考慮した資源割当て手法

金丸 達雄[†] 横田 裕介^{††} 大久保 英嗣^{††}

[†]立命館大学大学院理工学研究科 ^{††}立命館大学情報理工学部

1 はじめに

現在、我々は、同一地点に配備された無線センサ・アクチュエータネットワーク上のノードにおいて、複数のサービスを独立に実行するための仮想ネットワーク(以下、WSAVNと記す)について研究している[1]。WSAVNにおけるサービスは、2種類の仮想ネットワークに割り当てられる。すなわち、高優先度・広範囲のサービスのためのグローバルネットワークと、低優先度・局所的なサービスのためのローカルネットワークである。これらのサービスは、ノード上において複数のプロセスが資源(センサ、無線通信デバイス、アクチュエータ)を利用することで実現される。

WSAVNのノードでは、資源利用不能、応答性の低下、誤作動といったプロセス間での資源利用を原因とした問題が発生する(図1)。この問題の原因として、ノード上での資源利用の競合(センサ、無線通信デバイスで発生)や方針の矛盾(アクチュエータで発生)が挙げられる。これは、WSAVNでは、従来のセンサネットワークと異なり、複数のアプリケーションが資源を利用するために発生する。また、プロセスがノードに無線通信で配布されるため、プロセス間の資源利用のタイミングを静的に解析できないことも理由の1つとして挙げられる。

本稿では、ノード上の競合による問題(資源利用不能、応答性の低下)を回避するためのプロセスの目的を考慮した資源割当て手法を提案する。具体的には、資源利用不能を回避するため、プロセス毎に資源利用の期間を予約する。また、応答性の低下を回避するため、プロセス間で資源利用の期間が競合した場合、資源利用を前倒しする。本手法により、競合を原因とした問題に対処し、プロセスの資源利用の要求を達成する。

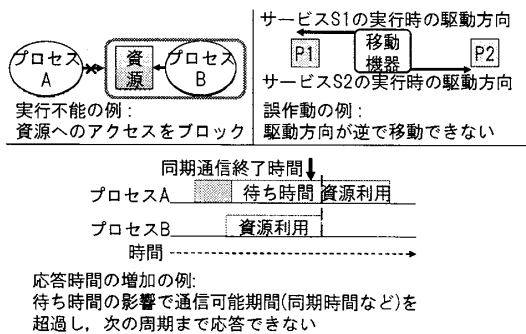


図1 資源利用を原因とした問題

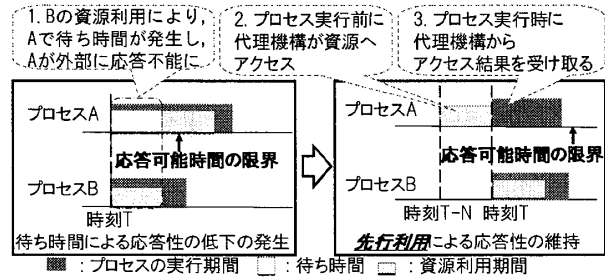


図2 先行利用による応答性の維持

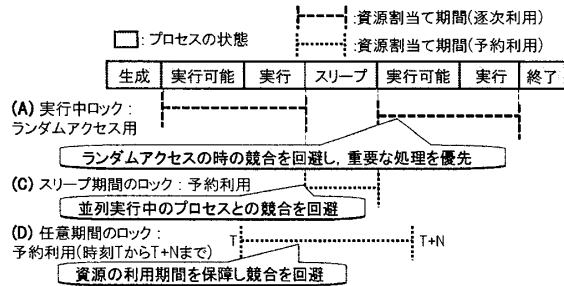


図3 資源の占有期間のタイプ

2 想定環境

本章では、本手法が想定するノード上の資源管理機構とその他の環境について述べる。

資源管理機構は、資源利用要求(後述)による資源の占有期間の予約、および2種類の資源利用方法をプロセスに対して提供する。資源利用は、占有期間を予約した後に行われる。まず、資源占有期間の予約を図3の期間を利用して行う(詳細は、文献[1])。次に、逐次利用と先行利用(図2)により資源を利用する。逐次利用とは、プロセス実行中に任意のタイミングで資源を利用する方法である。また、先行利用は、以下の動作により資源を利用する方法である。

1. プロセスが代理機構に資源へのアクセスを依頼する。
2. プロセスは、次回実行時に代理機構から資源利用の結果を受け取る。

逐次利用は、資源の利用結果に応じて、資源利用方法を変更するプロセスを対象にしている。また、先行利用は、周期的な資源利用を行うプロセスを対象にしている。周期的なプロセスは、周期を守ればタイミングを変更可能であり、かつ変更可能な期間が多い(スリープ期間が長い)ためこの方法が有効である。

その他の想定環境について述べる。まず、資源を利用して結果を得るまでの最大応答時間が既知であるとする。また、スケジューラは、プロセスの起動時刻を外部モジュールに対して通知できるとする。さらに、プロセ

A Resource Allocation Method based on Purposes of Processes on Sensor and Actuator Nodes

Tatsuo Kanamaru[†], Yusuke Yokota^{††} and Eiji Okubo^{††}

[†]Graduate School of Science and Engineering, Ritsumeikan Univ.

^{††}College of Information Science and Engineering, Ritsumeikan Univ.

スが実行中に資源を占有する場合、最大占有期間が既知であるとする。センサの場合、現実世界の事象がプロセスの挙動と関係なく一定期間で収束するため、この期間の把握は可能である。また、通信デバイスの場合、通信の成否を判断するためにこの期間が設けられると考えられる。

3 WSAVN ノード上の資源割当て手法

3.1 資源利用要求

資源占有の期間を予約するために、プロセスは資源利用要求を資源管理機構に発行する。資源利用要求は、以下の項目から構成される。

利用する資源 どの資源を利用するか指定する。

利用方法 逐次利用か先行利用か指定する。

利用回数 この予約が何回有効か指定する (1 から無限)。

利用期間 資源を利用する期間を指定する。指定しない場合、1 回の資源利用の最大応答時間が割り当てられる。

先行利用を行う場合、実行タイミングや競合した場合の対処を指定する必要がある。そのため、以下の項目を追加で指定する。

基点時刻 資源を利用する基点時刻を指定する。または、資源利用要求を発行しているプロセスのスリープ中に資源を利用することを指定する。

利用周期 周期的な利用であれば、資源を利用する周期を指定する。

競合時のタイミング 競合相手の資源利用の終了を待つか、競合前に前倒して行うか、資源利用のタイミングを変更しない (競合した場合、プロセスの優先度に基づき、資源を割り当てる) か指定する。

状態の共有 競合した場合、資源の利用方法が同一 (パラメータなど) であれば、資源利用の結果を共有して複数資源を利用しないか否か指定する。

復帰処理 割当て不能になった場合、空き時間を待って実行するか指定する。

パラメータ 資源を利用する際に、ドライバに提供する値を指定する。

3.2 資源割当て手順

資源利用要求の発行、先行利用・逐次利用を行うプロセスがスリープ状態になるごとに資源割当てを行う。図 4 に検査の概要を示し、以下に詳細な手順を示す。

1. 資源割当ての検査期間を決定する。予定されている資源利用 (プロセス、先行利用) の中で、最長のものを検査終了時刻とする。
2. プロセス実行中に逐次利用する場合、次のプロセス起床時を基点時刻として占有期間の割当てを行う。
3. 先行利用の場合、検査終了時刻まで基点時刻と周期を用いて割当てを行う。これは、 $\text{基点時刻} \leq \text{基点時刻} + \text{周期} \times n + \text{利用期間} \leq \text{検査終了時刻}$ となる n の範囲で行う。
4. 競合した場合、最大優先度のプロセスの資源利用を優先する。
5. 競合した場合、逐次利用するプロセスの優先度が低ければ、高優先度の資源利用の終了を待った後の資源利用を設定する。
6. 先行利用のプロセスが競合した場合、以下の 7 から 9 の処理を行う。

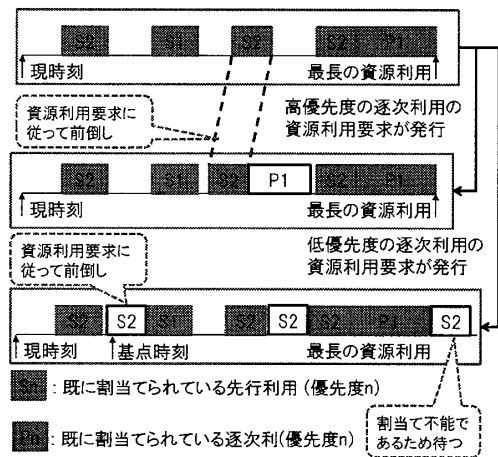


図 4 資源割当ての例

7. 低優先度の先行利用は、資源利用要求の“状態の共有”の項目を検査する。高優先度の先行利用のパラメータと自身のパラメータが一致すれば、その時間帯に 1 度だけ資源を利用して結果を共有する。
8. 低優先度の先行利用が状態を共有できない場合、資源利用要求の“競合時のタイミングの項目”に従って資源利用期間を変更する。
9. 資源割当てが失敗した場合で資源利用要求の“復帰処理”が有効である場合について示す。このとき、資源利用は、割当てがされていない時間 (空き時間) に配置される。なお、空き時間がない場合は、その先行利用を破棄する。

4 考察

本手法では、資源利用を前倒しすることで競合による問題を回避している。これは、競合の回避が困難であるためこの手法を利用している。なお、複数の資源利用要求が存在した場合、すべての資源利用の周期が倍数かつ位相が異なる場合以外では、競合は発生する。また、本手法は、資源利用期間を変更する際、プロセス間の調停を必要としない。そのため、ノード上のソフトウェア開発者の負担を軽減できる。これは、プログラム作成者がプロセス毎の交渉プロトコルを作成する必要がないためである。

本手法を実装する場合、競合が連鎖して計算量が増加する可能性がある。そのため、時刻の管理をスロットで行うことにより競合の判定の計算量を削減する必要があると考えられる。

5 おわりに

本稿では、センサ・アクチュエータノードにおけるプロセスの目的を考慮した資源割当て手法について述べた。今後、シミュレーションによる評価、およびクロスボー社の MicaZ を用いて LiteOS [2] 上で実装を行う。

参考文献

- [1] 金丸 達雄, 横田 裕介, 大久保 英嗣, センサ・アクチュエータネットワークにおける仮想ネットワークの構成と資源管理機構, 情報処理学会研究報告, 2008-EMB-10, pp. 67-74, 2008.
- [2] Cao, Q., Abdelzaher, T., Stankovic, J. and He, T.: The LiteOS Operating System: Towards Unix-Like Abstractions for Wireless Sensor Networks, Information Processing in Sensor Networks, pp. 233-244, 2008.