

リソース管理ソフトウェア“リソースマネージャー”による車載マルチメディアシステムの性能改善

三橋 禎*

Improvement of Performance for Car Multimedia System by Resource Management Software "Resource Manager"
Tadashi Mihashi

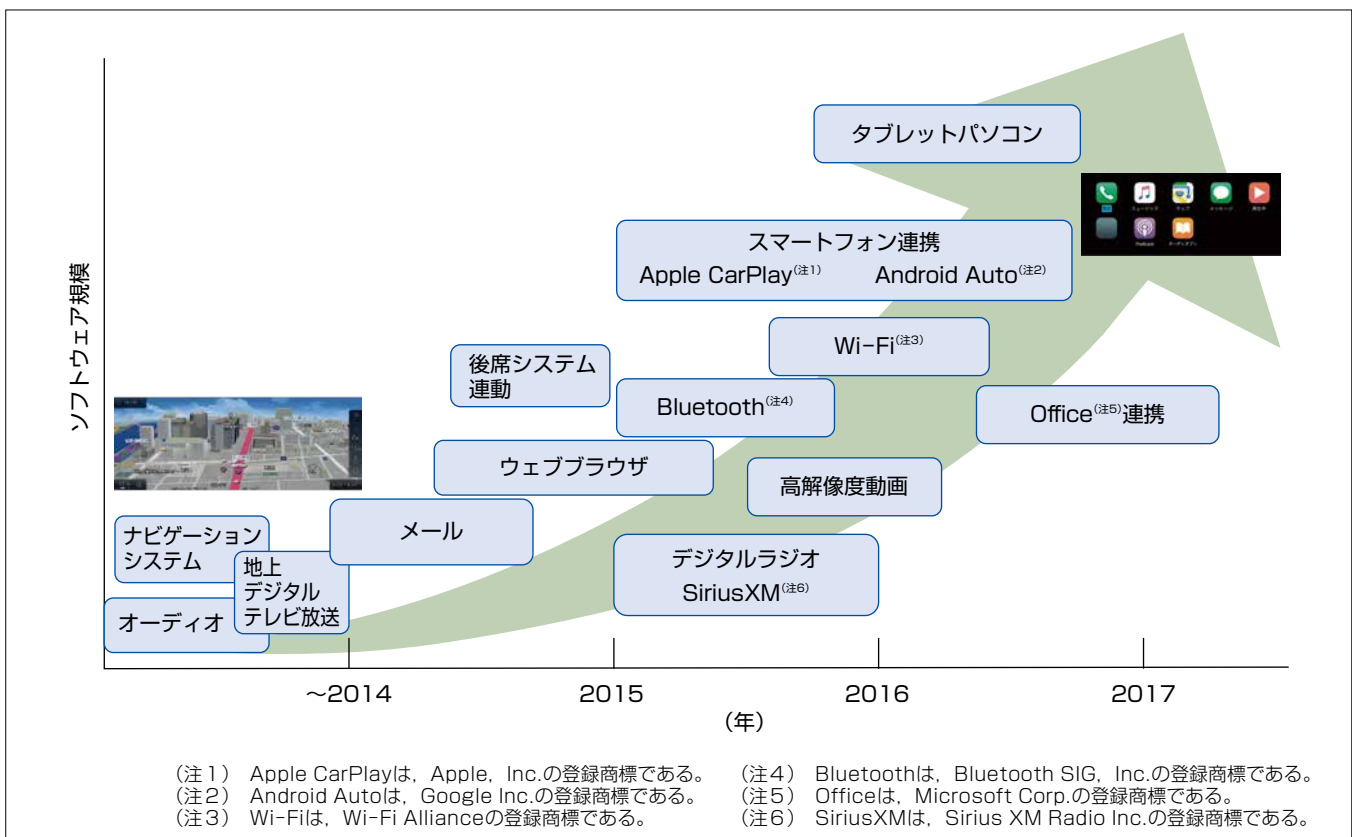
要 旨

車両に搭載されるマルチメディアシステム向けに、システムのリソースを最適化するリソース管理ソフトウェア“リソースマネージャー”を開発した。近年の車載マルチメディアシステムは、従来のナビゲーションよりも多機能・大規模化が進み、システム全体のリソースを最適化するために煩雑な調整が発生する。さらに、開発中や量産開始後に仕様変更要求が生じた場合、そのたびに同様のリソースの最適化を行う必要があった。

特に、車載マルチメディアシステムでは、開発開始前に決定されたハードウェアを変更することなく機能追加の仕

様変更に対応することが求められる。したがって、最初に決定した、限られたCPU処理能力と、メモリ容量の中で、即時応答性能を保ったまま、機能追加を行うことが必要になり、全てのユースケースでこれらのリソースが許容量を超えないように注意して開発を行う必要がある。

これらを鑑み、車載マルチメディアシステムで動的にCPUやメモリの有限リソースを管理できるリソースマネージャーを組み込むことで、性能を維持しつつ新機能の継続的な追加を実現できた。



車載マルチメディアシステムの高機能化とソフトウェア規模の増大

車載マルチメディアシステムの対応機能は増加の一方であり、それに伴ってソフトウェア規模も増大していく。販売後の車両でも、ソフトウェア更新だけで新しい機能に対応できるよう、市場の要求も高まっている。

1. ま え が き

近年の車載マルチメディアシステムのソフトウェアは大規模化が進み、システム全体のリソースを最適化するために煩雑な調整が発生する。さらに、既に販売された車両向けにも新機能追加ソフトウェアの開発要求が生じるため、ハードウェア変更による対応ができない。したがって、CPUやメモリのリソース不足と性能劣化への対応が不可欠となる。

これらを鑑み、システムで動的にCPUやメモリの有限リソースを管理できるソフトウェア“リソースマネージャー”を開発した。これによって、システムの基本性能を維持しつつ新機能の継続的な追加を実現できた。

本稿では、リソース管理の対象となる車載マルチメディアシステムの構成と仕様変更への対応について述べ、今回開発したリソースマネージャーによるヒープメモリやCPUのリソース管理の方法とその効果について述べる。

2. システムの仕様変更への対応

2.1 車載マルチメディアシステムの構成

2.1.1 車載マルチメディアシステムのソフトウェア構成

近年の車載マルチメディアシステムでは、従来のカーナビゲーションや光学メディアなどによるコンテンツ再生だけでなく、衛星ラジオ再生やBluetoothを利用したスマートフォンからのコンテンツ再生、ウェブ閲覧、ハンズフリー電話、車両エアコン制御や照明制御など多種多様な機能に対応している。

実使用では、カーナビゲーションによる経路案内を実施しながら、Bluetooth接続されたスマートフォンから音楽を再生し、バックグラウンドでのFM放送の受信、インターネット経由でのリアルタイム交通情報の受信等、複数機能が同時実行される。

さらに、車載マルチメディアシステムは、例えばユーザーの操作によるメディアソース変更、エアコン操作やハンズフリー通話の呼発信といったエンドユーザーの要求に即時に回答できるようにする必要がある。したがって、それぞれのアプリケーションは全て即時にその機能を動かせるように、システム上で各機能が起動した状態で待機する常駐型のシステム構成が必要とされる。

また、スマートフォン等の車内に持ち込まれる機器の接続に対しても、ユーザー操作に対して素早く回答できるように、接続機器の音楽情報データや電話帳データなどをメモリに保持するようにしている。

2.1.2 車載マルチメディアシステムのハードウェア構成

ハードウェア設計に際しては、自動車メーカーからの価格低減要求に応えつつ、多種多様な機能要求と性能要求に対応することはもちろん、車両販売後のソフトウェア更新

による機能追加を考慮して構成を決定する必要がある。また、車載機器では振動や埃(ほこり)に対する要求も厳しく、車両販売後にメモリ追加やCPUアップグレードをすることを考慮した構成が困難である。機能追加に備えたCPU性能やメモリの選定については、自動車メーカーと議論・合意の上で開発を始めることになり、開発開始後の変更要求は初期に選定したCPU性能やメモリ容量で対応することが前提となる。

リソースマネージャーを組み込んだ車載マルチメディアシステムは、2016年量産開始になる高級車向けシステムとして、2013年に表1に示すハードウェア構成で開発を開始し、最新(2018年秋)のソフトウェアに対しても全く同じハードウェア構成を取っている。

2.2 仕様変更への対応

2013年の開発着手以降、スマートフォンの本格普及による連携機能の強化、車外との通信を使ったサービスの充実など、車載システムとしての技術トレンド追従、さらには開発システムの搭載予定車種の増加によって、膨大な仕様変更が発生した。開発中及び量産後の半年ごとの更新プログラムの仕様変更累計数、及びソフトウェア総ステップ数の推移を図1に示す。最も仕様変更要求が多く発生した2017年春実施の量産ソフトウェア切替えでは、448か所の仕様変更要求が発生し、その規模増加量はプログラムで80万行以上にも及んだ。変更の大部分は新たな機能追加に伴うものであるが、これは開発初期の想定を大幅に超えるものである。

先に述べたとおり、このシステムでは、全てのアプリケーションが常駐し、複数のアプリケーションが同時に実行されるユースケースが通常の使用で発生する。したがっ

表1. 車載マルチメディアシステムのハードウェア構成

CPU	Cortex ^(注7) -A15 4コア(1.5GHz)
RAM	DDR3 4GB
HDD	300GB
eSD	2GB
NOR FLASH	4MB

(注7) Cortexは、Arm Ltd.の登録商標である。
RAM : Random Access Memory
eSD : embedded SD

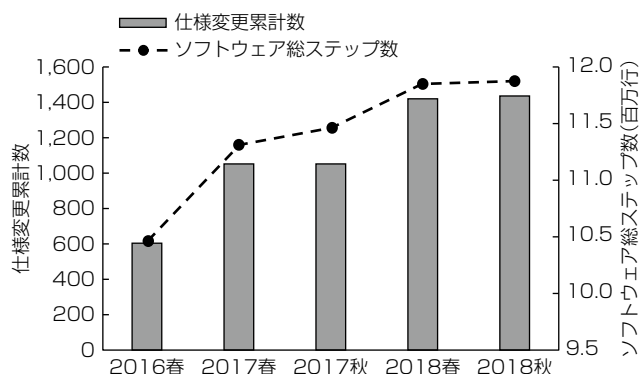


図1. 仕様変更累計数とソフトウェア総ステップ数の推移

て、このような大規模な機能追加は複数の機能が同時に実行された場合のCPUリソースの枯渇を招く。また、アプリケーション常駐による消費メモリ容量が増加するため、メモリリソースの枯渇を引き起こすおそれもある。これらのリソース枯渇によって、性能劣化やシステム安定性低下といった問題を引き起こす。

こうした課題を解決するためには、有限であるリソースの枯渇が発生する前に、本来そのタイミングでリソースを必要とするモジュールに対してリソースを割り当てる必要がある。その手段として、有限のリソースを動的に管理できるリソースマネージャーを開発することにした。

3. リソースマネージャー

3.1 ヒープメモリリソースマネージャー

対象機器では2,200MBのヒープメモリ容量があり、アプリケーションごとに静的にメモリ設計している。2016年量産開始版ソフトウェアに対し、2017年春の機能追加ソフトウェアでは、大規模な機能追加要求によってヒープメモリ使用量が当初想定値より最大450MB増える見込みとなり、ヒープメモリの容量を超える可能性があった。しかし、各アプリケーションのヒープメモリ使用量削減による最適化には限界があるため、2017年春以降の更なる機能追加にも対応できるように、ヒープメモリリソースを次のように動的に管理するヒープメモリリソースマネージャーを開発・実装した。

- (1) ヒープメモリの使用量モニタリングとしきい値判定
- (2) しきい値を超えた場合、ユーザーへの影響が少ない低優先機能から順にヒープメモリ開放実施
- (3) ヒープメモリ空き容量が十分になればヒープメモリ開放した機能へメモリ再割り当て

ヒープメモリ開放動作の優先順位は自動車メーカーとも協議し、次のように設定した。

- (1) WEBブラウザキャッシュ削除
- (2) ログ用バッファ破棄、ログ出力一時停止
- (3) リモートソフトウェアダウンロード一時中断
- (4) バックグラウンドでの描画用メモリ最適化処理中断
- (5) Bluetooth接続電話の電話帳キャッシュ量低減

低優先機能のヒープメモリ開放を行う場合のヒープメモリリソースマネージャーの動作例を図2に示す。ヒープメモリリソースマネージャーはメモリ確保要求を受け、メモリ残量と機能優先順位を加味し、必要に応じて低優先機能から順にメモリを解放していくことで、動的に高優先機能に対してメモリを確保できる。

このヒープメモリリソースマネージャー導入による効果を図3に示す。グラフはメモリ使用量が最大となるワーストケースでの各機能のヒープメモリの使用量を表す。大規模な機能追加要求によって、機能同時動作のワーストケ

スではヒープメモリ必要量が実際に存在するメモリ容量最大値を超える見込みとなっていたが、リソースマネージャー導入によって、2018年秋の切替えソフトウェアでのワーストケースでもヒープメモリ枯渇に陥ることなく、性能も維持することが可能になった。

3.2 CPUリソースマネージャー

対象システムのCPUでは、16,900MIPSの性能を持ち、CPUリソースの割り付けは優先度スケジューリング方式である。したがって、図4に示すようにCPU負荷が低い場合は優先度が低いスレッドも適宜実行されるが、優先度が高いスレッドが高負荷となった場合、低優先スレッドにはCPU処理リソースがなかなか割り当てられないという

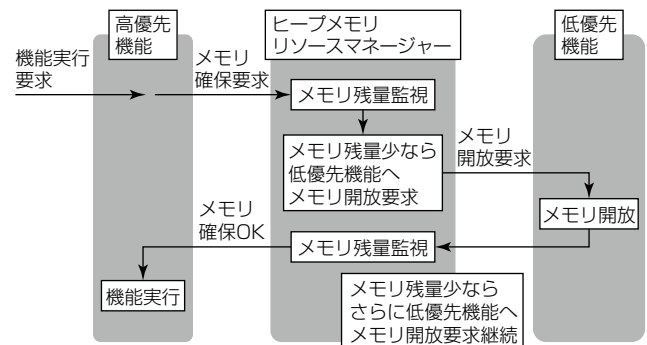


図2. ヒープメモリリソースマネージャーの動作例

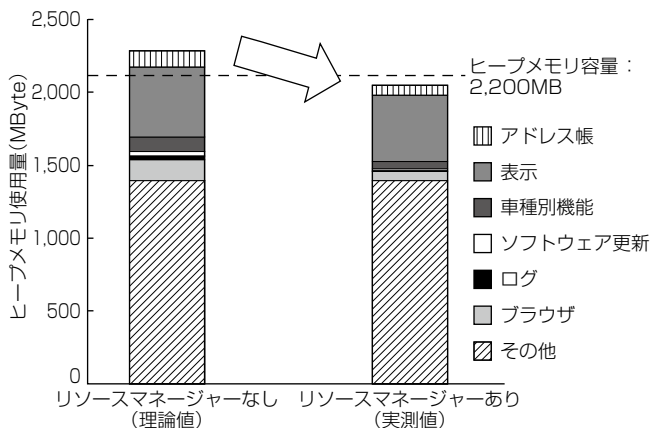


図3. ワーストケースでのヒープメモリ使用量

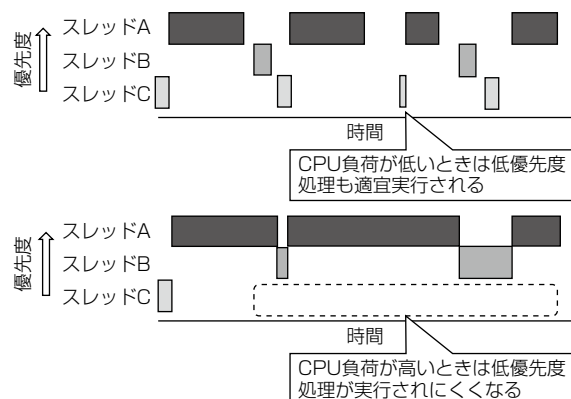


図4. スレッド優先度とスレッド実行の例

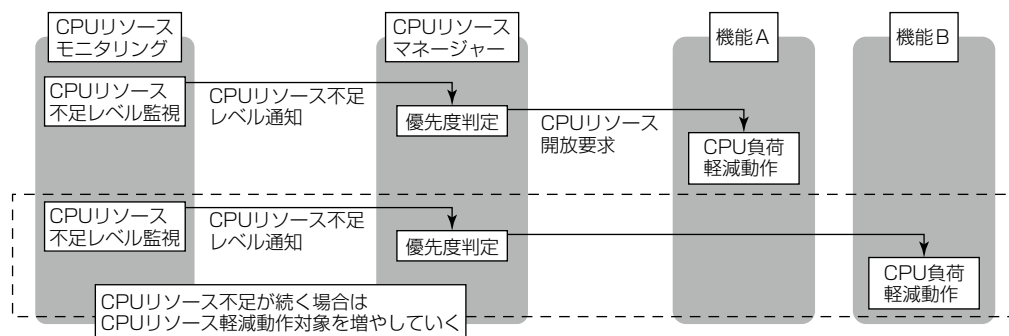


図6. CPUリソースマネージャーの動作例

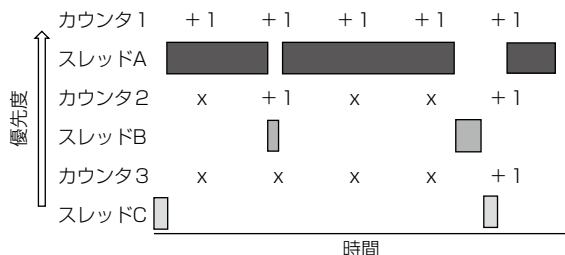


図5. カウンタの動作例

ことが起こる。このようにスレッド実行がなかなか行われない場合、一部機能の極端な性能劣化や動作停止という問題を引き起こす場合がある。

先に述べたとおり、対象機器では16,900MIPSの性能を持つCPUを採用していたが、2017年春の機能追加ソフトウェアでは大規模な機能追加要求によって、CPUリソースが最大44,100MIPS必要になることが見込まれた。しかしながら、各アプリケーションの処理負荷低減やスレッド優先度調整による最適化手法では限界があり、2017年春以降の更なる機能追加にも対応できるように、CPUリソースを次のように動的に管理するCPUリソースマネージャーを開発・実装した。

- (1) CPU使用状況モニタリングとリソース不足検出
- (2) ユーザーへの影響が少ない機能から順にCPU負荷軽減動作実施

各部の内容を次に述べる。

3.2.1 CPU使用状況モニタリングとリソース不足検出

モニタリング機構そのもののリソース消費を最小とすることを優先し、定時間ごとにカウンタ値を1ずつ増やす低優先度スレッドを複数追加した。CPU負荷が高い状況では、低優先スレッドのカウンタ値が増えにくくなることから、このカウンタ値を読み出すことでCPU使用状況を推測することが可能である。図5の例の場合、優先度の高いカウンタ1はカウント数が5増えるが、優先度の低いカウンタ2, 3はそれぞれ2, 1しかカウント数が増えていない。このようなカウント値の差異から、カウンタ2より優先度の高いスレッドAがCPUを占有している時間が長く、カウンタが一回しかされていないカウンタ3以下の優先度ス

レッドがほとんど実行されていないことが把握できる。

このようなカウンタによるCPU使用状況モニタリングを実施し、カウント値によってCPUリソース不足レベルを5段階に評価して通知するメカニズムを持たせた。

3.2.2 ユーザーへの影響が少ない機能から順にCPU負荷軽減動作実施

上記CPUリソース不足レベルに応じて、段階的にCPU負荷を下げるに当たり、CPU負荷軽減動作の優先順位は自動車メーカーとも協議し、次のように設定した。

- (1) ログ出力機能の停止
- (2) 音声認識用の動的辞書作成速度低減
- (3) Wi-Fiの通信速度低減
- (4) 音楽プレイヤー楽曲情報取得速度低減
- (5) 画面描画更新頻度低減(24回/秒→5回/秒)
- (6) ソフトウェア更新データダウンロード中断

CPU負荷軽減を行う際のCPUリソースマネージャーの動作例を図6に示す。CPUリソースマネージャーはCPUリソースモニタリング機能からのCPUリソース不足レベル通知を受け、負荷軽減優先度順にCPU負荷を軽減することによって、低優先スレッドにもCPUリソースを割り当てることができる。

このCPUリソースマネージャー導入によって、大規模な機能追加を実施したにも関わらず、ワーストケースでも性能劣化を最小限に抑えることが可能になった。

4. む す び

高級車向け車載マルチメディアシステムとして、最先端機能の継続追加と性能への高い要求に応えるために開発したリソースマネージャーについて述べた。今後の自動運転やカーシェアリングで要求される新しい車載マルチメディアシステムは従来のカーナビゲーションやメディアコンテンツの視聴にとどまらず、更なる機能向上や新しいサービスへの対応など進化が継続することが想定される。これらの新しい機能やサービスのタイムリーな提供ができるよう、容易にアップデート可能な進化し続ける車載マルチメディアシステムを提供していく。