

分散環境におけるリジューム機能

上坂 靖 高橋 祐亮

松下電器産業株式会社 情報システム研究所

近年コンピュータは、小型軽量化し携帯化が進んでいるが、一方では、ソフトウェアの大型化／複雑化により、OSやアプリケーションソフトウェアの立ち上げ／立ち下げに要する時間が増加している。その時間を短縮するためのソフトウェア的なアプローチとして、リジューム機能がある。リジューム機能によって、OSおよびアプリケーションソフトウェアの停止や再起動の操作が必要なくなる。

本論では、現状のリジューム機能の問題点を明かにするとともに、分散環境に応用したリジューム機能の提案を行う。この機能は、自コンピュータの状態を保持することだけでなく、通信相手の状態、および、通信コネクションの状態を保持することも考慮しているものである。

Resume Function on Distributed Computing Technology

Yasushi Uesaka Yusuke Takahashi

Information Systems Laboratory, Matsushita Electric Industrial Co.,Ltd.
1006 Kadoma,Kadoma-shi,Osaka 571 JAPAN

Recent Computers have been getting smaller and more portable. On the other hand, Software on them are becoming more complex and larger. As a result, the time to boot up / shutdown OS and application software increases. A resume function is software that shortens the time. With the resume function, users will not be aware of shutting down / restarting both OS and application software.

In this paper, we discuss the problems of currently available resume functions. Then, we propose our resume function to which the distributed computing technology are applied. The proposed resume function holds not only the status of the host computer but the status of peer computers and connections established among the host and the peers.

1 はじめに

近年コンピュータは、そのハードウェアの性能向上に伴い、システムソフトウェア、アプリケーションソフトウェアも高機能化してきており、ソフトウェアのサイズは大型化の傾向がある。ソフトウェアの大型化／複雑化により、システムソフトウェアの立ち上げ／立ち下げや、アプリケーションソフトウェアを実際に使用する時に必要な初期化／モード遷移のために要する時間と操作は増加してきており、運用途中での仕事の中断と再開のためのコストも無視できなくなっている。このため、パーソナルな目的で使用されるコンピュータには、アプリケーションソフトウェアの実行中に電源を切断しても、電源再投入後、実行時の状態から仕事を再開できるリジューム機能が必要となってきている。携帯型コンピュータにおいては運用途中の電源断／再投入の回数が増えるため、リジューム機能はさらに重要であり、携帯型のパーソナルコンピュータやワードプロセッサでは、メモリをバッテリーバックアップすることで実現されている。

ハードウェアの小型化に伴い、UNIX においても携帯を行うような運用や、パーソナルな使われ方をされるようになってきており、リジューム機能が必要とされるようになってきている。UNIX では、信頼性が求められているため、バッテリーの容量不足のような事態を考えると、メモリをバッテリーバックアップする方式だけのリジューム機能の実現では不十分である。

また、UNIX ではスタンドアロン環境での利用は勿論、ネットワーク環境、分散環境での利用形態が必須であるが、パーソナルコンピュータなどで実現されているリジューム機能は、スタンドアロン環境で使用しているコンピュータの状態を保持するもので、通信を行っている場合、通信相手のコンピュータの状態を保持するということは考慮されていない。

本論では、まず、ネットワークを利用しないスタンドアロン環境でのリジューム機能の実現について述べ、次に、ネットワーク利用時の問題点を解決するための、分散環境でのリジューム機能について述べる。

さらに、リジューム機能の応用として、コンピュータ本体を移動させるのではなく、必要な機能、情報のみを携帯できるよう、分散環境下において中断した仕事を別の場所や別のコンピュータで再開させるとい

う、情報を携帯するという機能の提案を行う。

2 リジューム機能

2.1 スタンドアロン環境でのリジューム機能

2.1.1 リジューム機能とは

リジューム機能は、仕事の中断／再開に要するコストを削減する機能であり、UNIX においては、通常の複雑な立ち上げ／立ち下げ手順をスイッチの ON/OFF にすることで、ユーザにパーソナルコンピュータ並の使いやすさを提供する機能とも言える。

例えば、コンピュータを用いて仕事をしているとき、中断した仕事を再開することを考える。

電源の投入、ハードウェアの初期化、基本ソフトウェアの起動、初期化、さらに、アプリケーションソフトウェアの起動、初期化を経て、そこから目的の仕事、例えば、文書処理を行っていた場合では、目的の文書の呼び出し、編集箇所への移動、編集モードの設定、等々を行い前回の仕事は再開される。

リジューム機能のあるコンピュータでは、電源の投入、ハードウェアの初期化を行い、電源切断時に実行していたプログラムを、中断した状態から再開できる。

また、コンピュータが実際には使用されていないことを検知し、自動的にリジューム機能による電源の切断を行うことができれば、ユーザに余分な負担をかけず、仕事の速やかな再開を保証したまま、無駄な消費電力を抑制することができる。このような機能は、携帯型のバッテリー駆動式のコンピュータにとっては特に有益である。

リジューム機能とは、実行時の状態を保持したまま電源断(サスペンド)／再投入(リジューム)を実現する機能であると定義する。

2.1.2 リジューム機能の処理概要

リジューム機能は以下のように、電源の切断を行うサスペンド処理と、電源の再投入に伴うリジューム処理とで実現されている。

- サスペンド処理

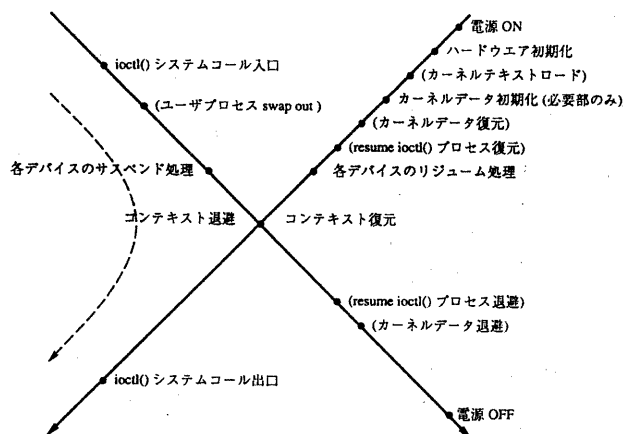


図 1: スタンドアロン環境でのリジューム

1. ユーザからの電源断要求を受け取ると、CPU, I/O の状態など、現在実行中のプログラムを再開できるだけの状態/情報を、メモリ上に退避する。
2. そのメモリをバッテリーバックアップして保持したり、メモリ内容をハードディスク上に書き込んだりすることで保持または記録した後、電源を切断する。

- リジューム処理

1. 再び電源が投入されると、保持または記録した情報をもとにプログラムの実行を再開させる。

2.1.3 リジューム機能の実現

UNIX におけるリジュームの実現においても、基本的には、パーソナルコンピュータなどで実現されているように、電源を切ることで失われるハードウェアの状態をメモリにセーブし、そのメモリをバッテリーバックアップにより保持することで実現する。

しかし、バッテリーの残り容量が少い時などでは、バッテリーバックアップによりメモリの内容を長期保持することができない。UNIX では、データの信頼性が要

求されるため、バッテリーの容量不足によるデータの消失は問題となる。また、UNIX では、一般に実装されているメモリがパーソナルコンピュータに比べ大容量であり消費電力も大きいため、バッテリーの容量不足の問題は特に顕著になる。このため、バッテリーの残り容量が少ない場合はメモリの内容を二次記憶装置(ディスク)に退避する。このように状態の保存領域を、

- バッテリーバックアップされたメモリ
- 二次記憶装置

の、二段階とすることにより、サスペンド/リジューム処理の高速化と、低消費電力化に貢献することができる。

サスペンド/リジューム処理の高速化、特にユーザの操作開始の待ち時間になる、電源再投入後の立ち上げ時間の短縮を目標とし、スタンドアロン環境でのリジューム機能を UNIX SVR4 上に実現した。

リジューム機能は、仮想デバイス /dev/resume をオープンし、ioctl() を行うことで実行される。

図 1 に示すように、resume ioctl() を行ったプロセスのコンテキストで、状態を保持するために必要なデータの退避を行い電源の切断を行う。電源再投入時は、リジューム機能による立ち上げであることを判断し、カーネル初期化のテンポラリなコンテキスト

で、退避したデータをもとに状態を復元し、resume ioctl() を行ったプロセスのコンテキストにスイッチする。ユーザプロセスから見ると ioctl() システムコールの入口と出口で、処理が継続しているように見える。

バッテリーバックアップされたメモリによる状態保持

図1において、“()” で囲まれている項目は、状態の保存領域をバッテリーバックアップされたメモリを選択した時には実行しないものである。

メモリ内容の退避/復元を行わないため、サスペンド/リジューム処理に必要な時間における、ハードディスクの電源投入時から回転が安定化するための待ち時間など、ハードウェアの初期化に要する時間が占める割合が大きく、問題となる。

この問題に対し、以下のような方法をとることで、リジューム処理の高速化をはかる。

- 時間のかかるハードウェアの初期化は、デバイスドライバのリジューム処理時に、初期化ルーチンをバックグラウンドで起動する。
- 電源再投入後、はじめてそのデバイスを使う時、まだ、初期化が終了していない場合、初期化終了を待つ。

このような手法をとることで、ハードディスクの初期化が完了しなくても、オンメモリ上のプロセスは動作することができる。この結果、リジューム処理にかかる時間は約1秒となった。

二次記憶退避による状態保持

メモリ内容の二次記憶退避による状態保持において、主記憶装置として実装されているメモリの内容を全て退避させるとすると、実装メモリ容量が大きい場合、退避に必要なディスク容量が大きくなり現実的でない。また、退避データの増加はサスペンド/リジューム処理時間の増加につながる。退避データを極力少くし、かつ、リジューム機能専用に必要なディスク容量を少なくするため、仮想記憶のメカニズムを応用し、以下のような方針でデータの退避を行うようにした。

1. カーネルテキスト部については退避しない。
リジューム処理時に、カーネルテキストは再ロードする。
2. ユーザプロセスのテキスト部については退避しない。
プロセスの実行時に、page fault を起こすことで実行ファイルよりロードする。
3. ユーザプロセスのデータ部については、サスペンド時にプロセスを強制的に swap out させる。
プロセスの実行時に、page fault を起こすことで swap 領域よりロードする。
4. ディスクキャッシュなど、ディスク上に複製のあるメモリについては、退避しない。

この結果、サスペンド/リジューム処理時に明示的に退避するデータはページテーブルなどのカーネルデータ、および、resume ioctl() を行ったプロセスのイメージのみとなる。

プロセスイメージの総量が約 24 MB となる環境で、二次記憶装置にデータ退避を行うサスペンド/リジューム処理を行った時の、退避/復元に要する処理時間を実測した。処理時間はハードディスク、システムバスなどの性能に大きく依存するため、実装メモリの内容を全て退避する場合の処理時間との比較を行った。

図2は、主記憶装置として実装されているメモリ量が16MBの時、実装メモリを全て退避させる時の処理時間を100とした、退避/復元に要する処理時間の割合を表したものである。

図2に示すように、本方式による実現においては、主記憶装置として実装されているメモリを2倍にしても、サスペンド時の退避に要する処理時間は1.04倍と、ほとんど変化しない。このため、実装メモリの内容を全て退避する場合の処理時間と比べ、主記憶装置として実装されているメモリが大きくなるほど有利となる。さらに、電源再投入後の立ち上げに必要な復元に要する処理時間は、主記憶装置として実装されているメモリが64MBの時には、実装メモリの内容を全て復元する場合の処理時間に比べ、90%削減することができた。

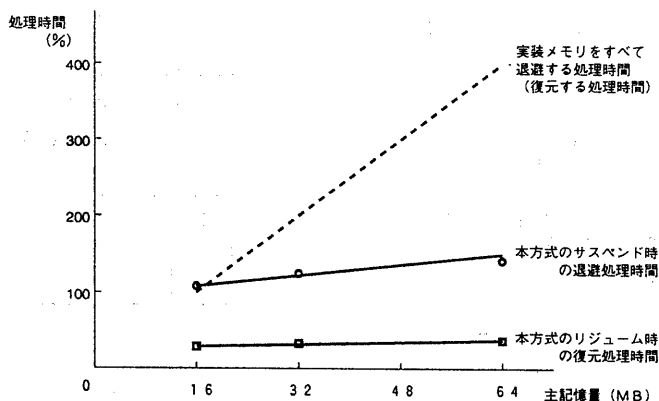


図 2: 退避/復元を行うデータ量

2.2 分散環境でのリジューム機能

2.2.1 分散環境でのリジューム機能とは

スタンドアロン環境で動作するコンピュータのリジューム機能は前節で述べてきたように、電源断時に自コンピュータ上の状態を保持することで実現されている。

分散環境で動作するコンピュータにおいて、コネクションレスの通信ではなく、TCP のようにコネクションを確立して通信を行っているときには、

- 通信相手がデータを送信した場合、通信相手は、受信側がデータを受信すると、プロトコル上の確認応答のメッセージを送り返すことを要求している、
- 通信相手がデータを送信していないときにも、お互いが生存していることを確認するためのデータのやり取りを行っている場合がある、

などの理由から、自コンピュータ上の状態を保持するサスペンド/リジューム処理を実行しても、通信の継続を行うことができない。

このため、コネクションを確立して通信を行うコンピュータに対するリジューム機能、ネットワークリジューム機能は、スタンドアロン環境でのリジューム

機能に加え、通信相手の状態、および、通信コネクションの状態を保持することで実現する必要がある。

2.2.2 ネットワークリジュームに求められる機能

ネットワークリジュームに求められる機能には、通常通信時のデータ、および、制御コードの透過性の保証の他、一方がサスペンド処理により電源が切断されている間、

- 通信相手プロセスの状態保存
通信相手プロセスの状態変化が起こらないようにする、
- 通信コネクションの状態保存
通信コネクションについて状態遷移が起こらないようにする、
- データ蓄積

通信相手からデータ送信があれば全て蓄積する。電源が切断されていたコンピュータの電源再投入後、リジューム処理によって通信が再開された時、蓄積されていたデータを転送する、

がある。

2.2.3 ネットワークリジューム機能のモデル

ネットワークリジューム機能を以下のようなモデルで考える。

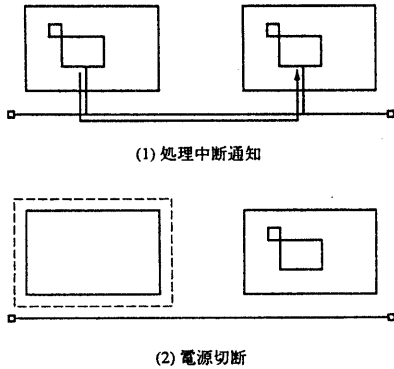


図3: ネットワークリジューム (サスペンド処理)

図3に示すように、サスペンド処理では、

- 電源を切断しようとしているコンピュータ上のプロセスが通信している全コネクションに対し、これから処理を中断することを通信相手に通知する (図中 (1))、
- スタンドアロン環境でのサスペンド/リジューム処理と同様にして電源を切断する (図中 (2))、

という動作を行う。

通信相手側は、処理を中断する旨の通知を受け取ると、コネクションを一旦解放する。そして通信を行っていたプロセスは処理再開を通知するメッセージが到着する時まで停止する。

図4に示すように、リジューム処理では、

- スタンドアロン環境でのサスペンド/リジューム処理と同様の処理を行い、通信以外の状態を復元する、
- 全コネクションに対し、通信相手に処理を再開する旨を通知し、再びコネクションを確立するよう依頼する (図中 (3))、

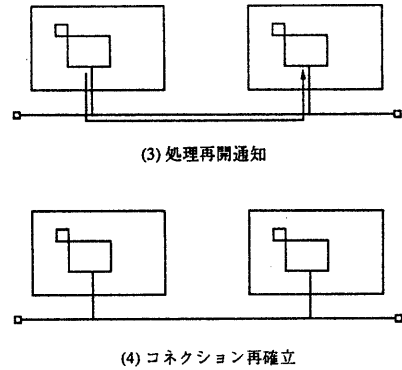


図4: ネットワークリジューム (リジューム処理)

という処理を行う。

通信相手側ではリジュームした通知を受け取ると、コネクションを再確立する (図中 (4))。

2.2.4 ネットワークリジューム機能の実現

上記のモデルを、通信プロトコルを TCP/IP を例として実現した。IP 層はコネクションレスなので、TCP 層におけるコネクションのみを考える。

モデルをそのまま実現すると、通信を行う各アプリケーション毎に、新しいライブラリの再リンク、または OS のライブラリの入れ換え、あるいは OS のカーネル内モジュールの変更などを行う必要がある。

リジューム機能を実行するコンピュータでは、リジューム機能の実装にあたり OS などの変更を行う必要があるが、既存の通信相手のコンピュータに対する変更は不可能な場合もある。そのため、ネットワークリジューム機能は、通信相手のアプリケーション、および OS の変更なしに実現することを目標とした。

自コンピュータがサスペンド処理により電源が切断されている間、通信相手にはあたかもまだ、動作しているかのごとく見せかける代理応答サーバをネットワーク上に設ける。

代理応答サーバは、サスペンド/リジューム処理により電源を切断するコンピュータからの依頼で、

1. 依頼を行ったコンピュータ上の全プロセスの全

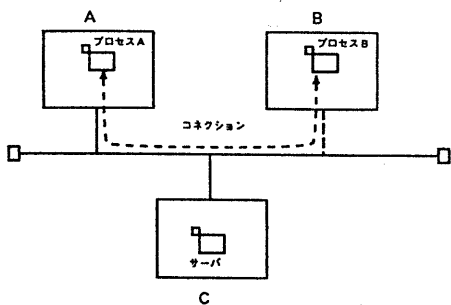


図 5: 代理応答サーバ (通常状態)

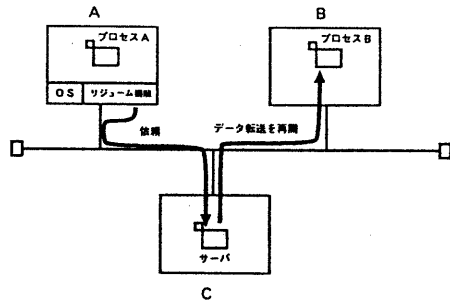


図 8: 代理応答サーバ (リジューム処理)

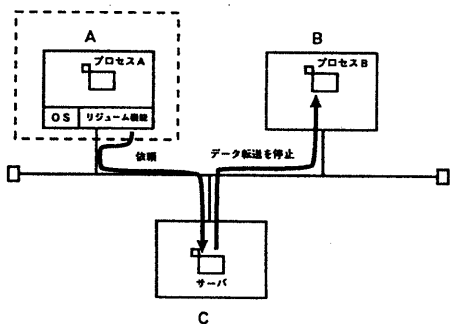


図 6: 代理応答サーバ (サスペンド処理)

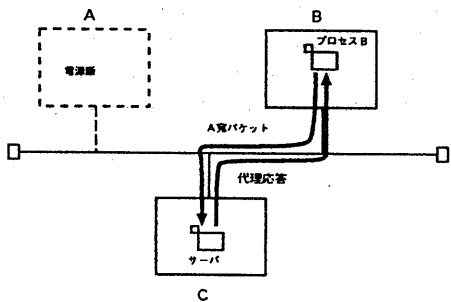


図 7: 代理応答サーバ (代理応答)

てのコネクションに対し、通信相手にウィンドウサイズ 0 を通知するという TCP レベルでのフロー制御により通信相手からのデータ転送を停止させる、

2. ネットワーク上のパケットを監視し、代理応答を依頼されているものに対しての KEEP ALIVE によるパケットに対しては代理応答する、

を行い、サスペンド／リジューム処理により電源を切断するコンピュータ上の通信コネクションを保持する。

図 5 に示すよう、コンピュータ A 上のプロセス A と、コンピュータ B 上のプロセス B がコネクションを確立し通信を行っている時、サスペンド処理を行う場合、コンピュータ A は、コンピュータ C 上の代理応答サーバに対し、通信コネクションの保持を依頼する (図 6)。

コンピュータ A の電源が切断されている状態では、図 7 に示すように、代理応答サーバは、プロセス B からプロセス A に宛てたパケットを横取りし、代理応答を行う。

リジューム処理時には、コンピュータ A は、代理応答サーバに対し、リジューム処理の依頼を行い、通信コネクションはもとの状態に復帰する (図 8)。

ネットワークリジュームについては現在実装中である。

2.3 コンピュータの移動

スタンドアロン環境、分散環境でのリジューム機能について述べたが、サスペンド／リジューム処理を行って電源が切断されたコンピュータの移動についてはないものと仮定している。

これは、移動を行う場合、コンピュータを異なるネットワークに移動させても、移動前と変わらない通信を保証する必要があるためである。

このような、コンピュータが異なるネットワークに移動した時にも、移動前と変わらない通信を行うことを保証する移動管理方式の研究が活発になってきている。

移動管理をパケット転送により行う方式 [1] では、パケット転送サーバと代理応答サーバを結合させることで、コンピュータを異なるネットワークに移動させても、リジューム機能を実現できる。

2.4 リジューム機能の応用としての情報の携帯化

携帯型のコンピュータは今後いつそう小型化、軽量化が進むであろうが、用途によってはキーボード、ディスプレイに適正サイズがあり、ある大きさより小さくならないものもあると考えられる。このようなコンピュータに対しては、リジューム機能の応用として、コンピュータ本体を移動させるのではなく、保存した状態に関する情報だけを移動させ、移動した先の別のコンピュータ上に、保存した状態を復元し仕事を再開させることが考えられる。

- サスペンド／リジューム処理により保持する状態を、ネットワークを介して、あるコンピュータ上に保存し、別のコンピュータで、その保存してある状態をもとに仕事を再開させる。
- あるいは、フラッシュメモリの IC カードのような、不揮発性で、携帯可能な記憶装置上にサスペンド／リジューム処理により保持する状態を記録し、別のコンピュータ上で、その保存してある状態をもとに仕事を再開させる。

以上のような処理を行うことで、情報の携帯化を実現する。

スタンドアロン環境でのリジューム機能、ネットワークリジューム機能をベースに実現していくが、以下のような問題点を解決する必要がある。

- 各コンピュータの、
 - － 実装メモリ容量の違い
 - － 接続されている周辺 I/O 装置の違いなど、ハードウェアコンフィグレーションの違いを吸収できるか。
- 各コンピュータからの参照できる環境の等価性をいかに実現するか。

3 おわりに

実行時の状態を保持したまま電源断／再投入を実現するリジューム機能の必要性と、スタンドアロン環境、分散環境下でのリジューム機能の実現について述べた。

現在スタンドアロン環境のリジューム機能は評価中であり、ネットワークリジューム機能については実装中である。

今後、ネットワークリジューム機能の評価を行い、分散環境でのリジューム機能、情報の携帯化の有効性を実証していく予定である。

参考文献

- [1] H.Wada, T.Yozawa, T.Ohnishi and Y.Tanaka, "Mobile Computing Environment Based on Internet Packet Forwarding", USENIX Winter Conference January 25-29, 1993 San Diego, CA