5ZA-07　分散共有メモリシステム *Lemuria* の Linux上での構成と通信の高速化手法

大浦 貴士 †　近藤 照之 ††　芝 公仁 ††　斎藤 彰一 †††　大久保 英嗣 †
† 立命館大学理工学部情報学科　†† 立命館大学大学院理工学研究科
††† 和歌山大学システム工学部情報通信システム学科

1　はじめに

分散共有メモリは，ワークステーションクラスタなど，分散配置されたノード上で動作している各プロセスの，仮想アドレス空間の一部を共有するための機構である．我々は，大規模な分散並列処理環境や広域分散環境において，計算機間でデータを共有する手段として分散共有メモリが有効であると考えている．現在，我々が開発を進めている *Lemuria* は，この分散共有メモリ機能を提供することを目的とした分散並列処理のためのプラットフォームである．

これまで，*Lemuria* は，Solaris などの Unix 上で開発を行ってきた．分散共有メモリはデータ共有の手段であるため，異機種間でデータを共有することが重要である．そこで，*Lemuria* を Linux へ移植することにより，より多様な環境で動作させることが可能となる．また，*Lemuria* での通信を高速化させることで，*Lemuria* の動作を高速化することができる．本論文では，*Lemuria* の Linux への移植と通信の高速化手法について述べる．

2　*Lemuria* の構成と特徴

Lemuria は，ユーザプログラムにリンクして分散共有メモリ機能を提供する C 言語ライブラリである libLemuria と，クラスタの管理やクラスタ間の通信などを行う Lemuriad 及び Reflector の 2 種類のサーバから構成されている．*Lemuria* の全体構成を図 1 に示す．

Lemuria は，パーソナルコンピュータやワークステーションなどの一般的な計算機を対象としている．*Lemuria* では，複数の計算機からなる小規模な計算機群をクラスタと呼ぶ．*Lemuria* の最大の特徴は，階層化されたクラスタ上に，分散共有メモリを実装している点である．クラスタを階層化することにより，通信量やクライアント情報を削減し，大規模分散処理に対応している [1][2].

An Implementation of Distributed Shared Memory System Lemuria on Linux and Its High Speed Communication
Takashi OURA †, Teruyuki KONDO ††, Masahito SHIBA ††, Shoichi SAITO †††, and Eiji OKUBO †
†Department of Computer Science,
Faculty of Science and Engineering, Ritsumeikan University
††Graduate School of Science and Engineering, Ritsumeikan University
†††Department of Computer and Communication Sciences,
Faculty of Systems Engineering, Wakayama University

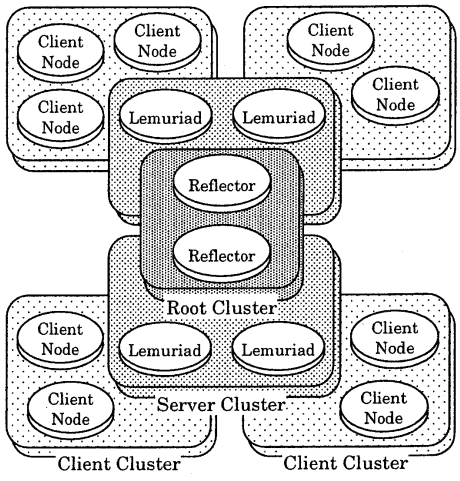

図 1: *Lemuria* の全体構成

3　Linux への移植

Lemuria の特徴として，既存の分散共有メモリシステムが実行対象としなかった大規模分散環境と広域分散環境における分散共有メモリの利用を可能にしたことがある．これらの分散環境においては，多種多様な OS とハードウェアが利用されている．我々は，どのような OS でも分散共有メモリを使用可能にするために，Lemuria を各種 OS に移植している．今回行った Lemuria の Linux への移植について述べる．

3.1　Linux の特徴

Linux は，GPL(GNU General Public License) にしたがって配布されている Unix 系 OS である．Linux は，Intel 80x86 系の CPU だけでなく，alhpa, SPARC, PowerPC などの CPU 上でも動作するものがある．今回は，より一般的な Intel 80x86 系の CPU 上で動作する Linux への移植を行った．

3.2　実装

Lemuria は，これまでに Solaris や IRIX などの Unix 系 OS や Windows 上でも動作している．しかし，Linux

は, BSD 系でも SystemV 系でもない, 独立した Unix 系 OS である. そのため, 移植を行う上で若干の問題が生じる.

Lemuria は, マルチスレッドを使用して記述されている [3]. Linux では, POSIX 準拠の pthread ライブラリを利用することで, マルチスレッドを実現することができた. また, システムコールにおいては, 特に, メモリ管理に関連したシステムコールで多くの問題が発生したため, これらのシステムコールの変更を行った. さらに, シグナルが発生した際に, それを正しく処理できないなどの問題が発生したため, カーネルについても若干の変更を加えなければならなかった.

4 通信の高速化手法

ソフトウェアによって分散共有メモリを実現するためには, ノード間でメモリの内容の一貫性を保つための通信を行うことが必須である. したがって, 一貫性を保つための通信が, 分散共有メモリの性能に大きな影響を与える. 本章では, Lemuria において通信を高速化する手法について述べる.

4.1 データリンク層を使った通信

Lemuria では, 分散共有メモリを実現するために UDP 上に独自のプロトコルを実装することによって, ノード間の通信を行っている. しかし, これらの通信においては, UDP や IP のプロトコル処理と *Lemuria* 独自のプロトコル処理では, 重複している部分あり, それらのオーバヘッドが大きくなってしまう. そこで, ノード間の通信においては, libLemuria が直接データリンク層にアクセスして通信をする. これによって, UDP や IP のプロトコルオーバヘッドを削減することができる.

4.2 Linux Socket Filtering

Linux において, アプリケーションがデータリンク層にアクセスする方法として, いくつかの方法がある. 今回は, 効率などの点から Linux Socket Filtering(LSF) を用いた. LSF は, アプリケーションがデータリンク層をアクセスする機能を提供するパケットフィルタである. LSF は, BSD のパケットフィルタである Berkeley Packet Filter(BPF) [4] をもとに作られたものである. LSF を使ったパケットの流れの様子を, 図 2 に示す.

LSF のもっとも大きな特徴は, 強力なフィルタ機能である. アプリケーションは, 独自のフィルタを作成し各パケットに適用することで, 容易に独自プロトコルによる通信を行うことができる. また, フィルタリングは, カーネル内で行われるため, アプリケーションにコピーされるデータ量を抑え, フィルタリングのオーバヘッドを削減している.

Lemuria のノード間通信において, 独自プロトコルのための独自のフィルタを作成する. これを LSF に用いることで, データリンク層を使った *Lemuria* の独自プロトコルによる通信を実装する. これにより, 通信のコストを削減し, より効率のよい通信を行うことができる.

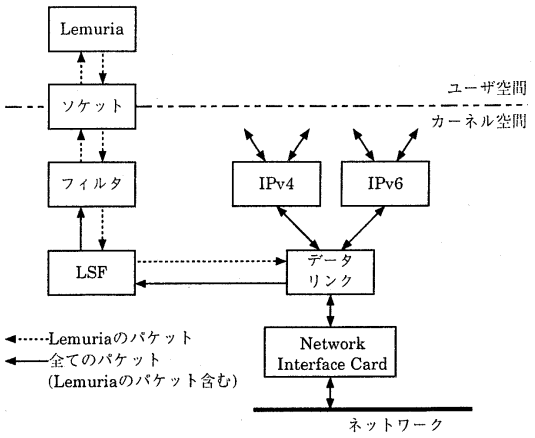

図 2: LSF でのパケットの流れ

5 おわりに

本論文では, *Lemuria* の Linux 上での構成とデータリンク層を使った通信の高速化手法について述べた. *Lemuria* を Linux へ移植することで, より多様な環境での動作が可能となった. また, データリンク層を使って通信を行うことにより, 上位層のプロトコルオーバヘッドを削減し, より高速な通信を行うことができる. これにより, *Lemuria* の動作を高速化することができる. 今後は, 現在 UDP 上で実装されている *Lemuria* 独自のプロトコルを, データリンク層レベルで実装するための独自プロトコルの拡張などについて検討して行く予定である.

参考文献

[1] 斎藤 彰一, 大久保 英嗣 : "分散配列処理のためのプラットホーム Lemura における分散共有メモリの性能評価", 電子情報通信学会論文誌, D-I Vol. J82-D-I, No. 3, pp. 457-466 (1999).

[2] 斎藤 彰一, 國枝 義敏, 大久保 英嗣 : "広域分散環境における分散共有メモリの実現とその性能評価", 情報処理学会論文誌, Vol. 40, No. 6, pp. 2563-2572 (1999).

[3] 近藤 照之 : "分散並列処理のためのプラットフォーム *Lemuria* の FreeBSD への移植とマルチキャストの実装", 立命館大学理工学部情報学科 卒業論文 (1998).

[4] W. リチャード・スティーヴンス著, 篠田 陽一 訳 : "UNIX ネットワークプログラミング 第 2 版 Vol.1 ネットワーク API: ソケットと XTI", 株式会社トッパン (1999).