

4Q-4

ボード化LISPマシンにおける
RPC機能の検討

菅原昌平

NTT ヒューマンインタフェース研究所

1. 概要

専用プロセサを持つシステムを、汎用のプロセサを利用したFront End Processor(FEP)と専用プロセサを組み合わせて実現することがある。例えばLISPワークステーションELIS¹⁾はLISPプロセサボードと入出力のサーバとして動作するFEPからなる。この場合LISPプロセサボードに対してFEPから処理依頼が来ることはなかった。しかし、LISPプロセサボードをFAや各種の制御システムに組み込んで用いる場合や、FEP上のソフトウェアを使いながら特定の処理のみLISPプロセサボード上で行わない場合、FEPあるいは外部からLISPプロセサボードに対して処理依頼を行なう必要がある。さらに、組み合わせるFEPの機能に対応して効率のよい機能分散が実現できる、柔軟性のある双方向の機能分散の機構を検討した。

またネットワークを介した機能分散と、FEP・LISPプロセサボードのローカルな機能分散の機構の両立、マルチLISPプロセサボードシステムでの適応性も考慮した。

2. FEP・LISPプロセサの結合システム

LISPプロセサボードを実行機として用いる場合、その特徴を生かした使い方を考える。開発環境としてLISP環境やLISPマシンを用い、実行機として汎用プロセサを用い、開発したソフトウェアを実行機用にコンパイルしなおすアプローチと比較すると、

- 1) OS相当の機能もLISPで開発でき、特殊な入出力装置を用いたりする場合の開発効率が上がる
- 2) 実行機上のソフトウェアを動的に変更するのが容易
- 3) OS相当の機能に対しても動的な変更が容易等の利点がある。

一方、従来指摘されている問題点として、GCによる遅延がある。この点に関してはGCフリーコーディ

ングを可能とする手段を提供することで対処できる²⁾。特にシステムの再コンパイル、再立ち上げ等を行なわなくても動的に入出力制御機能や、システムサービスの機能を変更できることが長所となる。もちろんAPについては従来のLISPシステムの長所がそのまま生きており、動的に戦略を変更したり知識を変形していく処理に適している。

動的な処理をLISPプロセサボードに担当させる場合、処理要求の発生は、単なる入力収集等と違って非周期的、非同期に起こる。すなわちFEP・LISPプロセサボードの間での機能分担は事前にスケジューリングできず、また各々で行なう処理内容はまったく異なる。このような状況で要求発生時に相手側を呼び出すメカニズムとしてRPCを用いる。

3. FEP・LISPボードRPCの特徴

基本的にLISPボードからFEPへのRPCは入出力要請、FEPからLISPボードへのRPCは処理の依頼が行なわれる。

• FEPからLISPボードへのRPC

通常のRPCではサーバ側のサービスプログラムの識別子を登録し、クライアントはその識別子でサーバのプロシジャを呼び出し、処理結果を受け取る。LISPボード上のサーバはそのプロシジャ自体が動的変更を許す点の特徴となる。すなわち、1)クライアント側がサーバとなるべきプロシジャを任意の時点で定義する、2)必要な時点でサーバのプロシジャを書き換える、等の使い方ができる。また、

- リクエスト/アンサーの一往復で終了するconnectionlessの呼び出し
- リクエストでconnectionが張られ、何回かのやりとりを行なって終了するconnection-orientedな呼び出し

が利用できる。

さらに、プロシジャの実行環境の設定により、一連の処理が終了してから、次に同じ処理を起動したとき、前回の呼び出しの副作用を残すか/残さないかを使い分けることができる

• LISPボードからFEPへのRPC

組み込みシステムなどでFEPがあまり高度の機能を持たない場合を考慮する必要がある。基本的に提供できるサービスはFEP主導で決まり、動的変更は考えない。

通常のRPCと同様に、FEP側のサービスプロシジャが(FEP上)登録済みのときその識別子を用いて呼び出す、リクエスト/アンサー型のconnectionless呼び出しである。FEPは入出力のリクエストに対するサーバ機能以外持たないことも有り得る。

4. ネットワークRPC

他のシステムとのネットワーク経由のRPCについては標準プロトコルを用いる必要がある。現時点ではNFSのサポートに必要なことから、SUNのRPCをサポートする。ローカルなRPCは

処理プログラムを動的に変更しない

リクエスト/アンサー型のconnectionless呼び出しに限定する

サーバ、クライアントの並列動作を用いない

メッセージプロトコルとしてXDRを用いる

等の制限をつけることにより、ネットワークRPCと同じように動作する。すなわち、ローカルRPCのサービスプロシジャのうちこの条件を満たすものはネットワークRPCのサーバとして提供可能である。表1にこれらのRPCの関係を示す。

機能	FEP->LISP7' ヲセサ	LISP7' ヲセサ->FEP	ネットワーク
動的変更	可	不可	不可
副作用有	可	不可	不可
副作用無	可	可	可
connectionless	可	可	可
connection-oriented	可	不可	不可
並列動作	可	可	不可

表1. RPC機能の分類

5. マルチプロセサ環境

ELISのLISPプロセサボードはネットワーク経由でマルチプロセサ化する以外に、FEPに対し複数枚接続し、FEP経由でマルチプロセサとすることが可能である(図1)。この場合それぞれのプロセサが処理をやりとりする方法として、ネットワークRPCを縮退させる方法とローカルRPCを用いる方法とがある。特にLISPシステムのマルチプロセサ構成であるから、自由度の大きなローカルRPCを利用し、FEPとの通信で用いるチャンネル番号にソースプロセサのidを追加した簡易通信プロトコルを用いる。ネットワークで接続したマルチプロセサとしたい場合は4.で述べた形に制限した呼び出しを用いれば良い。図2.にこれらの実現イメージを示す。

参考文献

- 1) 渡邊他「知能処理用ワークステーションELIS」研実報第37巻第2号、129-136頁(1988)
- 2) 鈴木他「新ELISのシステム概念」電子情報通信学会秋季全国大会(1989)

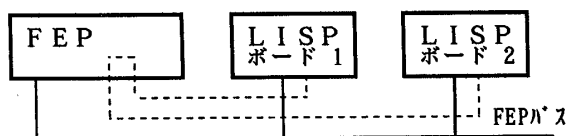


図1. FEP経由のプロセサ間通信

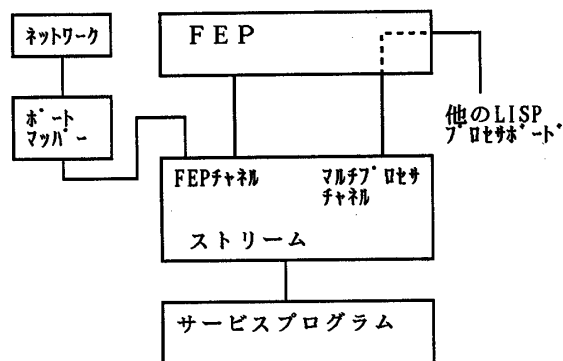


図2. 各種RPCの実現イメージ