

並列コンピュータ Cenju-3 の実行環境について

3T-3

†加藤 智香, †鳴澤 勝, †嶋野 一人, †浅野 由裕, ‡中田 登志之

†NEC 情報システムズ, ‡NEC C&C 研究所

1 はじめに

ここ数年、コンピュータの高速化への要求はますます高まっており、並列コンピュータによる並列処理も急速に広がりつつある。その中で、NECは、並列コンピュータ Cenju-3 を発表し、既にいくつかのシステムが稼働している。並列コンピュータのユーザが増えてくるにつれ、性能向上はもちろんのこと、その利用環境の充実化もまた重要視されるようになってきた。

Cenju-3 では、以下の機能を実現するための利用環境を提供している。

- マルチユーザ、マルチジョブ
- Cenju-3 利用のためのユーザインターフェースの簡略化、および、汎用化
- 遠隔地でのプログラム開発、および、Cenju-3 の利用

今回の発表では、ユーザ利用環境、主にマルチユーザ、マルチジョブの環境構築とそれに伴うセキュリティの向上について述べる。

2 Cenju-3 システム概要

Cenju-3 は、プロセッシングエレメント（以下 PE と呼ぶ）に VR4400 を用い、最大 256 台の PE を高速な多段結合網で接続した分散メモリ型並列コンピュータである。そしてホストコンピュータであるワークステーション（以下 WS と呼ぶ）に接続され、そのバックエンドプロセッサとして機能する。

WS をネットワークに接続することにより、ユーザがそのネットワークに接続された別の WS 上でプログラムを開発し、Cenju-3 に投入することが可能である。これは WS 上で動作するユーザ利用環境が TCP/IP をベースに通信を行うことによって実現している。（図 1）

3 ユーザ利用環境概要

Cenju-3 のユーザ利用環境として、以下のものを提供している。（図 2）

- 並列処理用ライブラリ (PARALIB/CJ)
- PE 上で動作する OS (CJ3OS)

Execution environment of Cenju-3

†Chika Kato, †Masaru Narusawa, †Kazuhito Shimano,
†Yoshihiro Asano, and †Toshiyuki Nakata
†NEC Informatec Systems, Ltd. and †NEC Corporation

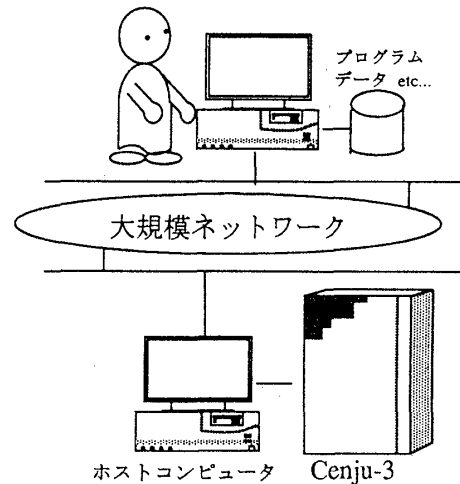


図 1: Cenju-3 構成

- マルチユーザ、マルチジョブの環境構築のためのデーモン (maser)
- Cenju-3 利用のためのインターフェースプログラム (cjsh, cjb)
- Cenju-3 利用のためのインターフェースプログラム用ライブラリ (HOS ライブラリ)

3.1 PE 上の環境

PARALIB/CJ は、Cenju-3 上で動作する並列プログラムを記述するために不可欠なライブラリである。プロセッサ間通信、同期、排他制御、リモートプロシージャコール (rpc) などの機能を、C および FORTRAN の関数として提供している。

また、Cenju-3 上から、ホストコンピュータ上のファイル操作等を行なうためのいくつかのシステムコールは PE 上で動作する CJ3OS がホストコンピュータへ rpc を行うことによって実現している。

3.2 WS 上の環境

ユーザが Cenju-3 を利用するための手段として、会話方式 (cjsh) とバッチ方式 (cjb) の 2 種類のインターフェースプログラムを提供している。これらは Cenju-3 を利用するための最低限の機能を持っており、HOS ライブラリを使用して作成している。そのため Cenju-3 の複数台の PE のうち、数台を会話方式で、別の数台をバッチ方式で同時に使用することも可能である。

HOS ライブラリはユーザにも提供されており、ユーザが独自に Cenju-3 利用のためのインターフェースプログラムを作成することも可能にしている。実際に Cenju-3 を利用するためには、インターフェースプログラムが maser との通信、Cenju-3 の資源確保、アプリケーションプログラムのロード、CJ3OS からの rpc の処理などを行なう必要がある。このような Cenju-3 でジョブを実行するために必要な処理を HOS ライブラリとしてまとめている。これにより、インターフェースプログラムのカスタマイズを容易にしている。

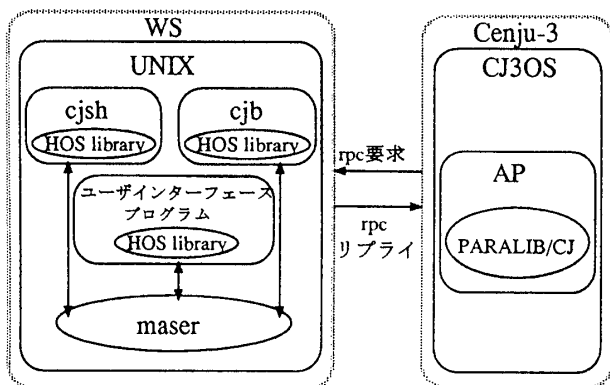


図 2: ユーザ利用環境のソフトウェア構成

4 マルチユーザ、マルチジョブの環境

Cenju-3 利用において、1人のユーザが常に全 PE を必要とするプログラムを実行するわけではない。そのため、一時に1ユーザしか使用出来ないと、Cenju-3 の利用効率が悪くなる。そこで未使用の PE が存在する場合、その PE で別のプログラムが実行出来るような機構(スペースシェアリング)を提供する必要がある。本システムではホストコンピュータ上の管理プログラムでこの機構を実現している。以下で、その詳細を説明する。

4.1 実装方法

ホストコンピュータ上で、管理プログラム(maser)が Cenju-3 の資源管理を一括して行なうことによりマルチユーザ、マルチジョブの環境を構築している。

maser は複数のインターフェースプログラムと通信し、PE の割り当て、解放等を行なっている。

maser には複数のプロセスがアクセスするので、それぞれの通信路、および、使用している PE をひとつのセッションとして、管理している。(図 3)

また、通信は TCP/IP をベースにしており、ネットワーク上の別の WS から Cenju-3 にジョブを投入することも可能である。

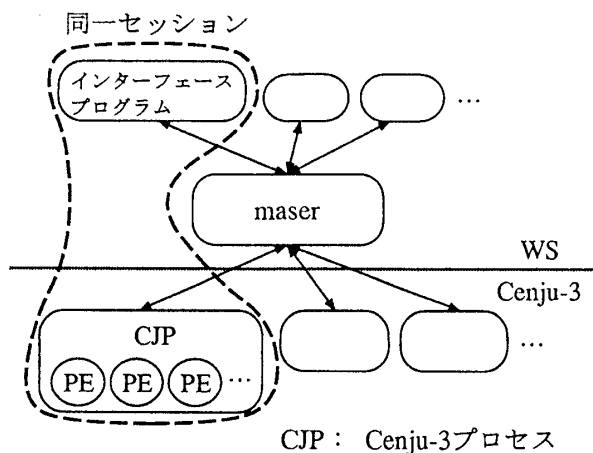


図 3: プロセスの構成

4.2 セキュリティの向上

Cenju-3 を利用するためのインターフェースプログラムはユーザが独自に作成することも可能なので、過って不正な message が送受信される可能性がある。また、複数のインターフェースプログラムが同時に maser と通信し、Cenju-3 を利用することになるので、お互いを干渉しないよう通信路の管理を行う必要がある。そのため maser ではそれぞれのセッションに、ID を割り当てて管理している。

PE とホストの間でやりとりされる message はすべて maser を経由するので、そのひとつひとつについて送信元、送信先が正当であるか確認している。これは、セッション ID、ソケットディスクリプタ、Cenju-3 プロセス ID、使用 PE の4つの組合せに基づいて行なっている。同時に、message のサイズ、フォーマットのチェックも行なっており、明らかに不正な message がやりとりされることを防いでいる。

一方、maser との通信路確立には TCP/IP をベースにした特定のプロトコルを設けており、不正なアクセスは受け付けないようにし、Cenju-3 利用に対するセキュリティの向上を計っている。

5 おわりに

本システムでは、マルチユーザ、マルチジョブ環境の構築、およびそれに伴うセキュリティの向上を達成した。それによって開発当初の目的であった Cenju-3 利用の効率化を実現することが出来た。

今後は、更にユーザ利用環境の充実化、および、処理速度の向上を計っていきたい。