

述語論理に基づくサービスベースシステムの

1U-6

並列実行の実現

萩野 正 深沢 友雄 田中 英彦 元岡 達

(東京大学工学部)

1. はじめに

計算機網中に分散して存在するサービス(各種のプログラム、データ等)をユーザに提供する際に、
 ・ユーザは、サービスの分散性に煩わされることなく、網中のサービスを使用することができる。
 ・計算機網中の各計算機上では、独立にサービスの拡張が行なえる。

等为目标として開発しているシステムがサービスベースシステム(SBS)である[1]。現在、サービスの要求言語として論理型言語を用いたSBSについて検討を行なっている[2]。

今回は、サービスを並列に実行する方法について検討を行なったので、その結果について報告する。

2. サービスの並列実行

現在までに検討してきたSBSでは、サービスの要求を出したノードは、その応答が返ってきてから、そのあとの処理を続けることになっている。その為、ネットワーク中で実際にサービスの処理を行なっているのは、常に1つの計算機のみであり、ネットワークの使用効率が低い。また、ノード間のファイル転送等、本来はそれぞれのノードでファイル読み出しのサービスとファイル書き込みのサービス等を並列に起動するべき場合には、その為の特別な処理が

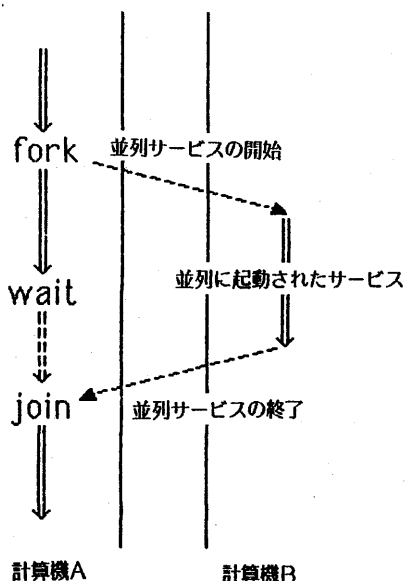


図1 forkとjoinによるサービスの並列実行

必要であり、独立にサービスを拡張することが困難である。

以上の欠点を克服するには、サービスの並列実行を実現する必要がある。その方法と実装について、以下の章で述べる。

3. サービスの並列実行の機構

SBSで並列サービスを実現するには、

1. 並列にサービスを起動する方法
2. 並列に起動されたサービス間でのデータ通信機構

について検討する必要がある。以下、それぞれについて示す。

3.1 並列サービスの起動

並列サービスは、システム管理者がサービスの定義時に、並列起動を示す述語を用いて陽に記述しておく。

並列起動を記述する方法としては、一般的には、

1. forkとjoinによる方法
2. cobegin とcoend による方法

等があるが、並列動作のわかりやすさや、記述能力などの点からforkとjoinを用いて記述するものとする。

ネットワーク内の各ノードは、forkの要求により並列サービスを開始し、joinの要求により、並列サービスの終了を待ち合わせる(図1)。

3.2 サービス間の通信機構

並列に実行されるサービス間のデータ通信機構としては、ポートとチャンネルによる方法をとる。これは、サービスの定義時にデータの出入り口(ポート)を定義しておき、並列にサービスを起動する時に、サービスのポート間をチャンネルと呼ぶ通信路で結ぶことにより、データ通信機構を実現する(図2)。チャンネルの機構は、下位の通信機構がサポートしているものとする。

4. 並列実行の実装

4.1 実験システムの構成

現在、VAX-11/730, VAX-11/780, M280Hの3台の計算機を用いてSBSの実験システムを構築中である。Prologの処理系としては、

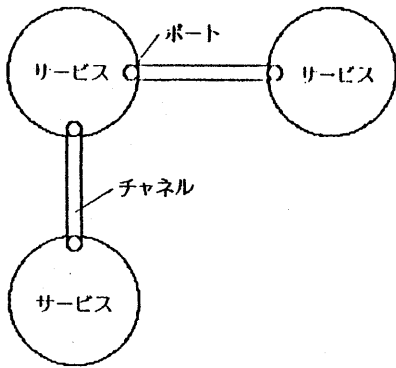


図2 ポートとチャンネル

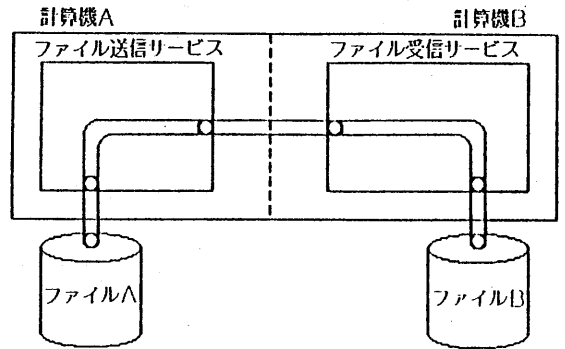


図3 ファイル転送の実現例

エジンバラ大学で開発されたC-Prolog を使用している。

4.2 forkとjoinの実装

3章で示した、

- ・forkとjoinによるサービスの並列起動
- ・ポートとチャンネルによるサービス間のデータ通信機構

を実現する為に、以下のシステム述語を実装する。

fork (s-name, s-id, c-list)

s-name: サービス名

s-id: 起動されたサービスの識別子

(UNIXのプロセスIDに相当)

c-list: ポートに接続されるチャンネルのリスト

forkを実行すると、指定されたチャンネルがポートに接続されてサービスが起動され、サービスの識別子が返される。

join (s-id)

s-id: サービスの識別子

joinは、s-idで示されたサービスの終了を待ち合わせる。

4.3 ファイル転送の実現例

4.2で導入したforkとjoinを使用して、ファイル転送を実現する例を示す。次のサービスが予め実装、定義されているものとする。

fsend (X): ファイル送信サービス

機能: Xという名前のファイルの内容をポートに出力する。

存在場所: 計算機A

frec (Y): ファイル受信サービス

機能: ポートへの入力をファイルYに書き込む。

存在場所: 計算機B

上記のサービスを次の様に並列に起動することにより、計算機Aから計算機Bへのファイル転送のサービス fcopy (X, Y) が実現できる (図3)。

fcopy (X, Y): -

```
fork (frec (Y), ID0, [C0]),
fork (fsend (X), ID1, [C0]),
join (ID1),
join (ID0).
```

5. 考察・検討

サービスの並列実行を実装することにより、ファイル転送や、ノード間でのパイプライン等、様々なサービスを実現することができた。しかし、現在の実験システムでは、ノード間では1対(往復)のチャンネルしか使用できないため、サービスの複雑な組み合わせは困難である。そこで、現在、通信回線を多重化して複数本のチャンネルを実装する方法について検討している。

6. おわりに

本稿では、論理型言語を用いたSBSに於けるサービスの並列実行について報告した。今後、ノード間の通信回線の多重化等を実現し、様々なサービスを柔軟に組み合わせることのできるシステムの開発を行なっていく予定である。

7. 参考文献

- [1] 深沢、田中、元岡、「サービスベースシステム」、情報処理学会、分散処理システム研究会シンポジウム、1984. 10
- [2] 深沢、荻野、田中、元岡、「論理型言語向きサービスベースシステムの構成」、第29回情報処理学会全国大会、6H-6、1984. 9