定理証明システムのための統合サーバの作成*

6T－9

石曽根 信† 澤田 寿実‡
(株)SRA§
e-mail: ishisone@sra.co.jp, sawada@sra.co.jp

1 はじめに

現在、我々は代数仕様記述言語 CafeOBJ [1] をベースにしたネットワーク分散型の代数仕様作成支援統合環境を構築中である [2]。このシステムの概要を図１に示す。システムは仕様の編集や他の仕様の検索を支援する編集・検索系、記述された仕様をプログラムとみなして実行する言語処理系、および記述された仕様の検証を支援する検証支援系から構成される。

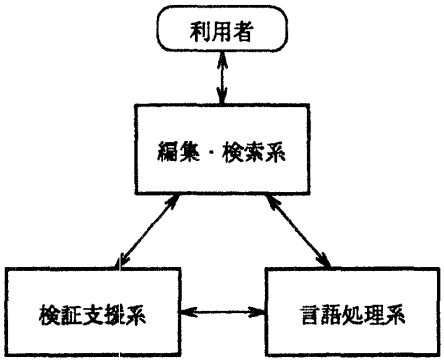

図 1: システム概要図

検証支援系は、記述された仕様の持つさまざまな性質について利用者が証明を行なうための支援を行なうため、定理証明エンジンをバックエンドとして用いる。エンジンには特定機能に特化したものも多く、これらを複数組み合わせることでより柔軟な支援を目指す。

分散環境下でこれらのエンジンに統一的な方法でアクセスできるようにするため、サーバを作成した。

2 統合サーバ

図２に作成したサーバの構成を示す。サーバは各エンジンに対応した TCP/IP のポートを作成する。エンジンにアクセスしたいクライアントが目的のポートに接続すると、サーバは自動的にエンジンプロセスを起動し、クライアントと接続する。クライアントとの接続が切れればサーバはプロセスを終了させる。つまりプロセスの起動および終了は全てサーバが管理する。

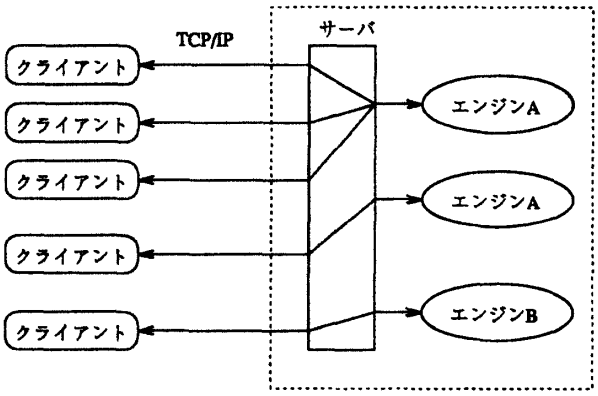

図 2: サーバ構成図

エンジンとの入出力は全て標準入出力を使用する。元々対話的動作を行なうエンジンならば問題ないが、データをコマンドの引数やファイルで渡さなければならないものに関しては、適当なフロントエンドプロセスを用意する必要がある。

クライアントはネットワークで接続されていれば、エンジンがどこにあっても接続でき、また好きな時に終了できる。接続方法も統一されており、インストールや環境設定や起動オプションの指定に悩む必要がない。

これだけでは Unix のネットワークサービスデーモン *inetd* と基本的機能に変わりはないが、本サーバはこれに加えて次のような機能を有する。

- 証明エンジンの複数クライアント間共有
- 証明エンジン実行処理への割り込み
- セキュリティ

3 証明エンジンの共有

構築中の環境では、利用者の要求に応じて複数のツールが協調動作を行なうため、複数のクライアントが１つの証明エンジンを同時に利用できることが望ましい。例えばあるクライアントがエンジンにデータを送り込み、別のクライアントがそのデータを用いて処理を実行することが考えられる。このために証明エンジンの共有機構を実装した。

クライアントがサーバとの接続時にエンジン共有を指定するとサーバはある ID を返す。以降、同じエン

---

* An integrated server for theorem proving systems
† Makoto Ishisone
‡ Toshimi Sawada
§ Software Research Associates, Inc.

ジンを共有したいクライアントはサーバとの接続時に同じIDを送ると共有モードでの接続となり、同一のエンジンプロセスを共有することになる。

各クライアントはそれぞれ独立にエンジンにデータやコマンドを送り、処理結果を受け取るが、実際には各データやコマンドは同一エンジンで実行される。実行中に他のクライアントから送られてきた場合にはサーバの持つ待ち行列に入れ、逐次的に実行する (図3)。

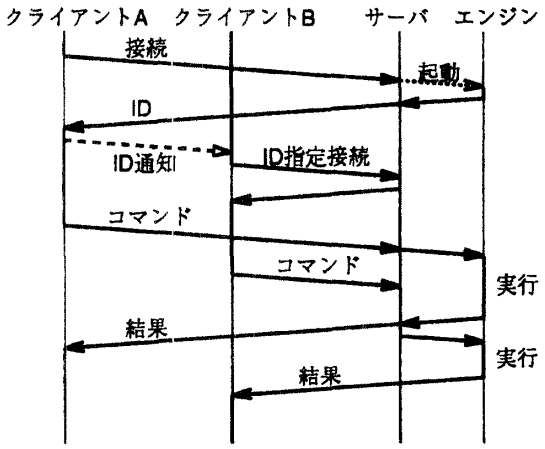

図 3: 共有モードでの処理の流れ

共有クライアント間およびエンジンとの間の同期を取るために、特殊文字を使用したごく簡単なプロトコルを実装している。共有モードのクライアント、およびエンジンはこのプロトコルを実装する必要がある。これはクライアントからのデータやコマンドの最後、またエンジンからの実行結果の最後に区切り文字として特殊な文字を入れるというもので、特にクライアント側での実装は非常に容易である。

4 割り込み

証明エンジンの処理は時として非常に長時間に渡り、また停止しないことも多い。しかも事前にそれを予測することが困難である。そこで、何らかの手段で処理を途中で中断する機能が必要である。ネットワークを通してエンジンにアクセスしている場合にはプロセスに直接シグナルを送って処理を停止させることができないので、別の手段が必要であり、次の2種類の手段を用意した。

1つは接続を切ることである。クライアントとの接続が切れたエンジンはサーバによって強制終了させられる。ただしこの場合、エンジンのプロセス終了とともにロードしたデータが失われる。

もう1つは特殊な文字を送ることである。^C を送るとサーバはプロセスにシグナルを送って現在実行中の処理を中止する。この場合プロセスは生きているのでデータが失われることはない。

5 セキュリティ

証明エンジンの中にはファイルシステムへのアクセス機能やプロセスの生成機能などを持つものも少なくない。このような機能は信頼のおける閉じたネットワーク環境下で利用する分には問題はないが、インターネットを通して広く一般に公開する場合、悪意のあるユーザによって、システムが危険に晒される可能性がある。根本的な対策はエンジン側でこれらの機能を利用不能にすることだが、不可能であったり、たとえできたとしても穴がある可能性がある。そこで「もしも悪用された場合の被害を最小限にとどめ、原因の追求を可能にする」という方針に基づき、次のような対策をとった。

- ファイルアクセス範囲の制限
  エンジンを実行する際にファイルシステムのルートディレクトリを変更することでアクセスできるファイルの範囲を制限する。
- アクセス権の制限
  エンジンのプロセスのアクセス権を低く設定することでファイルアクセスなどを制限する。
- 同時起動プロセス数の制限
  同時に多数のエンジンが動いてシステムの動作に影響を与えないよう同時起動プロセス数を制限する。
- アクセスログの収集
  万が一問題が発生した時に原因の追求および適切な対処の手助けとなるようにアクセスなどのログを収集する。

6 おわりに

この成果は情報処理振興事業協会 (IPA) が実施している「創造的ソフトウェア育成事業」の一環として行われたものである。

参考文献

[1] Nakagawa, A.T., Sawada, T., and Futatsugi, K. *CafeOBJ Manual*, SRA, 1997; available at `ftp://www.sra.co.jp/pub/lang/CafeOBJ/Manual/manual.ps`

[2] Nakagawa, A.T., "*Manipulating CafeOBJ on Networks*", in *Preprint for 12th Workshop on Algebraic Development Techniques*, Tarquinia, June 1997