

# DKLIC 処理系における分散論理変数の資源管理

## Resource Management for Distributed Logic Variables of DKLIC

600P031-4 高木祐介 上田研究室  
TAKAGI Yusuke, Ueda Laboratory

### 概要

普及している手続き型言語では、プロセス群の分散、同期、通信プロトコルの実装といった分散プログラミングに不可欠な処理の記述が非常に面倒である。

本研究では、動的プロセス生成、自動同期、非同期メッセージ通信を宣言的に記述する KL1 言語に基づき、既存の KLIC 処理系を拡張して宣言型ソケット通信の機能を提供する DKLIC 処理系を実装した。このことによって、直観的でなかった分散処理の記述を宣言的に行えるようにした。分散論理変数の直結や廃棄を検出することによって、既存の DKLIC 処理系では不可能であった二重送信や通信の中継を排除し、分散論理変数の ID を早期に再利用することを可能とした。

また、KL1 言語に蓄積された高度な静的解析の技術を分散プログラミングに適用することを可能とした。それだけでなく、既存の静的解析の技術である型・モード・線形性体系に明確には現れない通信路の確立（分散論理変数の open）という概念を明確に示した。分散プログラミングにおいては、素性の知れない通信データやモバイルコードに対して静的に検証を行えることは非常に重要である。

また、静的意味論を変えずに様々な通信手法を試みることは今後のネットワークの負荷を軽くする研究に役立つであろう。通信手法によるネットワークの負荷の変化を定量比較することは、本研究においても今後の課題である。

### keywords

KL1 言語, KLIC 処理系, 分散処理系, 宣言型分散プログラミング.

KL1, KLIC, Distributed Implementation, Declarative Distributed Programming.

### 1 既存の分散機構

1. プロセス. 並行処理に必要. ex) OS, GUI, シミュレーション.
2. ソケット. 共通の先祖を持たないプロセス間の通信に必要. ex) Wnn, X Window System.
3. RPC. ソケット通信の詳細をある程度隠蔽する. ex) NIS, NFS, Java RMI.
4. ORB. サーバの位置を検索する. ex) CORBA, Jini.
5. エージェント. プロセス移送によって通信の詳細をある程度隠蔽する. ex) Web Robots.

## 2 分散環境の問題領域

1. 並行性. ネットワークに接続された各ホストは本質的に並行動作する.
2. 通信遅延. 分散環境の通信遅延は無視できない程度に大きい.
3. オープンコンピューティング. オープンな通信プロトコルに従う限り, 通信相手を選ばない.
4. 例外処理. ネットワークの一部が故障することはしばしばである.
5. 保安性. ネットワークを通じて個人情報を漏らしてはいけない.

## 3 KL1 言語

述語呼び出しによって, プロセスの動的生成を宣言的に記述することができる.

プロセス間の共有変数を通信路として, プロセス間通信プロトコルを宣言的に記述することができる. 通信メッセージとして未定義変数を輸出入することによって, 通信路の動的確立を宣言的に記述することができる.

蓄積された静的解析の技術によって, 通信データ/モバイルコードの検証/最適化が可能である.

## 4 DKLIC 処理系

既存の KLIC-KL1 ソケットは未定義変数を輸出できない, 手続き型である.

未定義変数を輸出入できる, 宣言型ソケットを実装する必要がある.

単純な実装を目指す. KLIC 処理系を拡張する. プッシュ型の非同期メッセージ通信を行う. 次の階層に分けて実装する.

1. 接続サーバ dklicio.
2. 検索サーバ.
3. ユーザ定義サーバ.

4. アプリケーション.

本研究では dklicio を実装した.

## 5 DKLICP

dklicio が使う通信プロトコルである. 通信路の輸出入・書き込み・直結・廃棄を実現する. 通信路のライフサイクルは次の通りである.

1. open. ID を割り当てて輸出入する.
2. use. ID を利用して送受信する.
3. close. 書き込み・直結・廃棄して ID を回収する.

DKLICP は書き込み・直結・廃棄によって変数 ID を早期再利用する.

## 6 dklicio

次の階層に分けて実装した.

1. 分散論理変数 fusesusp. 書き込み・直結・廃棄されたことを検出して dklicio に報告する. 250/C
2. 変数表. fusesusp を管理する. ID の割り当て・再利用を行う. 400/C, 150/KL1
3. dklicio. 変数を輸出入する. 変数の書き込み・直結・廃棄を送受信する. 350/KL1

## 7 まとめ

既存の試作 DKLIC に比較して, 通信路の直結・廃棄を検出できる. 二重送信・中継を排除した. ID を早期に再利用する.

## 8 今後の課題

下位処理系を検討する. 例外処理を実装する. ID 割り当て方式を検討する. 検索サーバ・エージェントを実装する. もっと計算量の小さな通信プロトコルを設計する.