

カーネル用アスペクト指向システム: KLAS

東京工業大学大学院 情報理工学研究科 数理計算科学専攻

柳澤 佳里* 光来 健一 千葉 滋

概要

大容量のファイルのダウンロード中にリモートログインを行うと時おり著しい通信遅延が起きることがある。インターネット上での通信でこれが起きるのは往々にして考えられることであるが、LAN 内で通信している場合でもこのような症状が起きることがある。この原因としてネットワーク内の輻輳が考えられるのはもちろんだがカーネル内部でのリソース競合についても考えることが出来る。

これを確認するにはカーネル内部の挙動を知る必要があるためカーネル用のプロファイラが必要である。しかしながら、既存のカーネルプロファイリングのためのツールではプロファイリングできる箇所がかなり限定されておりリソース競合を調べるのに十分なプロファイリングができるとは言い難い。また、手動でソースコードを変更してプロファイリングを行う方法ではソースコードの変更のたびに再起動が必要であり、計算結果を返り値として指定している場合に返り値を見るにはそれを一時的に作成した変数に入れて表示するなど本来とは関係ない箇所の変更が必要となることもあってかえって管理が大変である。さらにカーネルのソースコードでは関数ポインタや構造体を用いてオブジェクト指向を C 言語で無理に実現しているコードが多用され実行フローの追跡が困難である。

この問題を解決するためにカーネル用アスペクト指向システム KLAS を提案する。KLAS はプロファイリングに必要な時間測定のコードを実行時に挿入、削除をすることを可能にする。このためコード挿入時にコンパイル、再起動することによるメモリ、ネットワーク情報の消失を防ぐことが出来る。これは再現が難しい事象が起きた瞬間にプロファイリングを開始できることを意味するためカーネルの開発効率の向上に繋がる。また、KLAS はアスペクト指向を利用したシステムであり、挿入するコードを OS と分離して書くことが出来る。これによりカーネル内のコードの OSI 階層を跨るような箇所に対してコードを挿入することが出来、階層を越えたプロファイリングが容易に行える。さらに、KLAS はコンパイルによって失われる型やローカル変数の情報を保持し、コード挿入時に利用する。これはコンパイルで失われる型情報やローカル変数の情報を用いてコード埋め込み位置の指定が出来ることを意味しており、関数ポインタや構造体が多用される OS のコードのプロファイリングをより容易に行うことが出来る。

実行時にコード埋め込むのにはコード埋め込み位置にブレークポイントトラップ命令を埋め込み、ブレークポイント発生時に指定した時間測定のコードを実行することで行う。ブレークポイントトラップ命令は 1 バイト命令なので 5 バイト命令である `jmp` 命令や `call` 命令と違い実行フローを破壊しない。また、コード埋め込みのためにコンパイル時にコードを挿入するための領域を作る方法と違い実行時のオーバーヘッドもかからない。

*E-mail: yanagisawa@csg.is.titech.ac.jp

KLAS は型やローカル変数名を利用してコード挿入位置を指定するが、これらの情報の取得は改造した gcc を用いてコンパイル時に行う。