

Javaがマイクロプロセッサに与える影響

HOT Chips V3より

渡辺 玲

IEEE主催の高性能チップに関するシンポジウム、HOT Chipsが開催されました。本稿ではJavaに関する話題を中心にレポートします。

発表プログラムはHOT Chipsの公式Webページ、<http://www.hot.org/hotchips>を見てください。

Javaアーキテクチャ

● 今年にJavaで盛り上がった！

「今年のHOT ChipsはJavaで盛り上がった」というと、「聞き飽きた」といわれそうですが、「盛り上がった」の意味合いが普通のコンファレンスとは全然違ったのです。このシンポジウムはチップを作る人たちの集まりなので、ソフトウェアやインターネット関係者はほとんど参加していませんでした。

では、何で盛り上がったのかというと、それはJavaのVM(バーチャル・マシン)のアーキテクチャ(すなわち命令セット)においてでした。

ご存じの方も多いと思いますが、Javaはそもそもプラットフォームを選びません。それはJava言語で書かれたソース・プログラム(.java)が一度、Javaコンパイラで中間コード(class)に落とされ、その後各プロセッサ固有のJIT(ジャスト・イン・タイム)コンパイラでネイティブ・コードに落とされ、実行されるからです。

この中間コードはJavaの機械語に相当します。JavaのVMは仮想的なプロセッサですが、本当のプロセッサと同じように、命令セットがあり、それが仮想プロセッサのアーキテクチャを定義しています。

● RISCの総本山になぐり込み

今回で8回目の開催となるHOT Chipsでは、過去にいろいろなチップを取り上げてきました。その中でも特にRISC推進者にとっては、RISCの優位性を論文で証明するまたとない場所となってきました。

スタンフォード大学、カリフォルニア

大学バークレイ校やシリコン・バレーの企業など、RISCの推進者が年に1度集まる場所に、今回異教徒Javaが姿を現したのです。

仮想プロセッサとしてのJavaは可変語長の命令体系をもつ、紛れもないCISCです。Intelのx86と同じく、1バイト目がOPコードで、それをデコードしなければ、何バイトの命令なのかがわかりません。さらに、汎用レジスタではなくスタックを使うスタック・マシンです。Javaプロセッサは今までRISC推進者が優位性を主張してきた固定語長命令、汎用レジスタ・マシンの存在意義を全面的に否定するものなのです。

しかも、それを世に送り出したのがあのSun Microsystems社です。同社は世間に先駆けてRISCを使ったワークステーションを作り、商業ベースで初めてRISCを成功させた会社です。そんなRISC連合の老舗ともいえる会社が起こした宗旨変えに波紋は広がりました。

CISCといえばIntelですが、Intelは好きでCISCを作っているのではなく、変えたくても機械語の互換性から変えられないのです。これに対して、チップ設計者にとってのJavaプロセッサはまさしく「過去から蘇りし亡霊」なのです。

● 伝説の人ビル・ジョイ

読者の皆さんの中には、たかが仮想プロセッサの話でそんなに大騒ぎすることはないのではないかと思います。中間コードでスタック・マシンを想定するコンパイラはほかにもあります。Javaコンパイラの専売特許ではありません。Javaの場合はその中間コ

ード自体が流布するので目立つだけではないかとも考えられます。

また、中間コードが可変語長なもの、通信(インターネット)で配ることや、組み込み機器で使うことが主になるJavaの用途を考えれば、コードのサイズがよりコンパクトになる点で利点があるとも言えます。

キーノート・スピーカだったビル・ジョイ^{注1}も、Javaのよいことは、各プロセッサごとにもう一度コンパイル(JITコンパイル)するところにあると、強調していました。

その言葉の趣旨は単にプロセッサを選ばない点にあるだけではありません。

ソフトウェア(コンパイラ)でやることはハードウェア(プロセッサ)ではやらない。その結果、ハードウェアが簡素になって、スピードが上がる。これはRISCが出てきたときの中心的な考えでした。しかし機械語の互換性が問題になって、実際にはインターロック回路などをプロセッサに組み込まなくてはなりませんでした。

ところが、JITコンパイラを使うことを考えれば話は違います。どんなプロセッサを作ろうが、それにあったJITコンパイラと一緒に付けければ問題はありません。それこそ、インターロックも何も無い、シンプルで高速なプロセッサを作っても大丈夫です。また独自のVLIWプロセッサでも大丈夫です。

● ビル・ジョイ vs. デイビッド・パターンソン

しかし、JavaのVMの命令セット(中間コード)が本当にJITコンパイラのことを考えて、よく練られた結果出てきたのかどうかは非常に怪しいものがあります。ビル・ジョイもJavaの中間コードに関しては「(内容ではなく)すでに出回っているということが重要なのだ」と、とても歯切れが悪く、夜のパネル・セッション^{注2}でかつての仲間であるパターンソン教授^{注2}も、「そんなことなら、いっそIntelのx86の機械語を直接インターネットで流したらどうだ。広く普及しているし、今さら

別のCISCを作るまでもないだろう。」と皮肉りました。

恒例の夜のパネル・セッションでは、主催者はいつも意図的にキーノートを叩く話題をテーマに選び、キーノート・スピーカーと参加者とのホットな論争を演出しています。

今年は、ビル・ジョイがキーノートのスピーカーに選ばれたので、夜のパネルのテーマは、「ソフトウェアかシリコンか-Javaにつながる最善の道」でした。当然、シリコンなんてだめだ・・・という懐疑的な意味が含まれていました。

夜のパネルは皆がワインやビールを飲みながら、わいわいといった感じで議論が進みました。屋外でやる開放感も加わり、楽しくスタンフォードの夜は更けていったのでした。

Java チップ, MMX, Touchstone, VLIW

● picoJava

このJavaチップの存在が、ますますビル・ジョイの発言、ひいてはJava全体の立場を不鮮明なものにしています。機械語独立はどこに行ったのでしょうか。

発表者は、中間コードを直接実行するpicoJAVAが一番Javaプログラムの実行速度が速いと力説していましたが、それではJITコンパイラは役立たずなのでしょう。JITコンパイラにより機械語の互換性の束縛から解かれて、その結果もっと速いプロセッサを作るのも夢ではない、といったビル・ジョイの話180度方向が違います。

Javaチップに絡むと、ビル・ジョイの歯切れもますます悪くなりました。筆者はこのpicoJAVAに関しては、今回の発表を聞いた限りではまだ、存在意義が納得できませんでした。

RISCの弱点である命令密度の低さからくる命令フェッチのボトルネックは今回も話題になりました。「アップルは68KからPowerPCに移り、それでコンパイラをやり直したが、2~3倍になるはずの性能が何割増しにしかならず、それはこ



のボトルネックのせいだ」と発表した人もいました。しかし、一方でDRAMの集積度が上がれば、一つのチップ(またはパッケージ)にすべての必要な主記憶を入れられるので、ボトルネックは解消されると予言した人もいます。パターンソン教授です。

● MMX (マルチメディア・エクステンション)-Intel

マルチメディア用に命令を追加するのは、SunのUltraSPARCや、NECのVシリーズなどですでにやられていますので、珍しいことではないのですが、x86が拡張されたというのは、市場占有規模からみて、インパクトは大きいと思います。Cyrixも同じ話題で発表があり気合いが入っていました。

● Touchstone

Microsoftの提唱する3Dチップ・セットのプロジェクト Touchstoneに関する発表がありました。Microsoftが音頭をとってチップを作ったということに新しい時代を感じます。日本からは富士通の米国人がこのプロジェクトに参加しています。

● VLIW

Intelは例によって何も発表しませんでした。できるまで沈黙を守つつもりでしょうか。今回発表されたのは、フィリップスと三菱の組み込み系のVLIWプロセッサです。RISCのときもそうでしたが、組み込み系は機械語の互換性の束縛がほとんどなく、常に必要に応じて最適なプロセッサを使う傾向があります。

5年くらい前からすでに、高い演算性能が要求される分野ではRISCが使われ



てきました。今もっとも演算性能が要求される分野はマルチメディアです。まず、このあたりからVLIWが使われていくことも十分考えられます。RISCの牽引車であるワークステーションは、以前のPCがそうであったように機械語の互換性の束縛から、あまりVLIWに簡単に乗り移れないのではないのでしょうか。HP以外でどこか計画しているところはあるのでしょうか。

まとめ:カオスの時代が到来か

10年ほど昔、CISCしかなかったところにRISCが登場して、RISC旋風を巻き起こし、CISC対RISCの一大論争が始まりました。昨年のHOT Chipsなどでは、それももう落ち着いてきたかに見えましたが、今年はCISCでスタック・マシンのJavaチップが出てきて、また一騒ぎです。

さらに、新顔のVLIWのチップが商業ベースで参戦してきて、プロセッサの世界はカオスの時代に突入したのではないかと感じました。

わたなべ あきら

マイクロプロセッサ設計エンジニア

注1 パークレー校の学生のときにパークレー版UNIX作りに加わり、卒業後にSunを創業したことはあまりにも有名。コンピュータの世界の伝説の人の一人。

注2 パークレー校で1980年からRISCを作り始めた。そのとき作ったRISC-IIがSunのSPARCの原型になった。MIPSを作ったスタンフォード大学のヘネシー教授と並ぶRISCの草分け的存在。両教授の共著の2冊の本は、我々プロセッサ設計者のバイブルとなっている。