

データ解析

研究室の U さんの仕事を頻繁に邪魔しながら，
シミュレーション結果のデータ解析．

CRuby と JRuby

MacOSX 10.5.7 の CRuby(ruby 1.8.6) と， JRuby 1.3.1 の話

```
p "%000%000%000%001".unpack("l*")  
p "%001%000%000%000".unpack("l*")
```

を， CRuby で実行すると，

```
[16777216]  
[1]
```

JRuby で実行すると，

```
[1]  
[16777216]
```

になる．エンディアンの違いなのね．

ソース / ディスティネーションに対応したエンディアンで pack/unpack するときは

```
* n: big endian unsigned 16bit  
* N: big endian unsigned 32bit  
* v: little endian unsigned 16bit  
* V: little endian unsigned 32bit
```

を使うのね．

ちゃんと注意事項として pack テンプレート文字列 に書いてあった．

Harnessing Horizontal Parallelism and Vertical Instruction Packing of Programs to Improve System Overall Efficiency

DATE'08

「どこかで話を聞いてきた」と， K 先生が紹介してくださったので読んでみた．
IRF と VLIW を組み合わせて，性能を落とさずに消費電力を削減したという話．
ちなみに紹介された方の論文は，

```
"Orchestrating Horizontal Parallelism and Vertical Instruction  
Packing of Programs to Improve System Overall Efficiency"  
Hai Lin, Yunsu Fei  
IEEE TRANSACTIONS ON COMPUTERS, VOL.58, NO.7 JULY 2009
```

MISA - 複数の IRF 中のエン트리へのインデックスを持つ
ID ステージの前半で， IRF な実命令に対応づけられる
コードサイズ削減 / フェッチのエネルギー節約

VLIW に IRF を実装する

(1)VLIW 命令をそのままバックする．手間は一緒だけど柔軟性がない

(2) 各インストラクションスロット毎に解析 / パックする .

(2) がいいのだけど , VLIW の命令スケジューリングがこわれちゃうので ISA に

- PISA
- SISA

を導入してスケジューリング .

結果 , 約 70% 程度の消費電力削減に成功 , という論文