Toward Lightning RubyVM

RubyKaigi2010

Technical Lightning Talk

Koichi Sasada
ko1@rvm.jp
Passionate VM Programmer

Creative Informatics,
Graduate School of Informatics and Technology,
The University of Tokyo
THIS PRESENTATION is about

Performance of Ruby VM

Speed-up, Efficient, Optimization, …
Our current Challenges in Sasada-Lab (Students task)

- AOT Compiler
  - Ruby to C
  - Ruby to X10
- Profilers
  - Memory Profiler
  - Performance Profiler
- Regexp AOT Compiler
- Atomic-Ruby (for Embedded)
- CI (w/ performance) enviroment

Sessions at Tomorrow Morning
My Challenge

- VM re-design
  - Re-visit Method (control) frame
  - Re-visit Relationship of Ruby and C
  - Instruction level re-design

- Introducing Parallelism
  - Against Fine grain parallelism
  - Against Coarse grain parallelism
Easy Performance Analysis (1)
Micro benchmark

OS: Linux 2.6.31 32-bit
CPU: Intel Core 2 Quad 2.66GHz
Mem: 4GB
C Compiler: GCC 4.4.1, -O3
Profiled by Oprofile

Fib

Pentomino

Profiled by Mr. Shiba

ruby 1.9.3dev (2010-05-26)
VM Re-design
Re-visit Method (control) frame

Current control frame structure

Args Local
Stack Values

LFP SP
DFP
BP, PC, ...

Value Stack
Method Invocation

Control Frame Stack

Args Local
Stack Values
Args Local

LFP SP
DFP
BP
LFP DFP
BP, SP

Value Stack

Control Frame Stack

CFP
VM Re-design
Re-visit Method (control) frame

- Too complicated
- Too many values to push

→ Simplify frame structure
Redesign:
  - VM frame structure
  - Method data structure
  - InstructionSequence structure
Easy Performance Analysis (2)

Macro Benchmark

rdoc

Rails

OS: Linux 2.6.31 32-bit
CPU: IntelCore2Quad 2.66GHz
Mem: 4GB
C Compiler: GCC 4.4.1, -O3
Profiled by Oprofile

ruby 1.9.3dev (2010-05-26)
Profiled by Mr. Shiba
Introduce Parallelism for Multi/Many core era

• **Parallel**, not **concurrent**
  - **Parallel**: run simultaneously
    - Not supported by Ruby

• **Concurrent**: manage several activity
  - Supported by Ruby’s Thread
Introducing Parallelism Principle

NO Parallel Threads in CRuby

I believe parallel threads introduce unhappy result.
Introducing Parallelism

Fine grain parallelism

• Run methods in parallel
  • Time consuming methods
  • Collection of independent methods

• Rubyist should not consider
  → VM run them in parallel auto.
Introducing Parallelism
Fine grain parallelism: Solution

- Write parallel code in C level
  - C extension writer should take difficulties
- Provide a framework to write parallel C extensions
- Now considering specification
Introducing Parallelism
Coarse grain parallelism: Solution

• Easy to consider
• Separate logic into VMs using MVM (Multiple-VM)
  • MVM, a.k.a vaporware (やるやる詐欺)
  • VMs run in parallel and lightweight inter-VM communication than inter-process communication
• Similar model as other languages
  • Scala, Erlang, …
Introducing Parallelism
Parallel GC

• GC consume time
  → Speedup with Parallel

• Parallel mark
  • I’m not sure about it, however, last day nari-san talk it is considerable

• Parallel sweep
  • Seems easy to support
Request

If you have applications
(1) have performance issues
(2) too slow, give up to run on Ruby

Please ask us.

They will be important research materials.
我々の業界ではご褒美です.
Ono more
Announcement

• **Escape from Japan** (to UK)
  • Next 2 months
  • I hope **I can concentrate in VM development**
  • To: UK people, please contact me if you have interest about my work

• Stay tuned for next news of VM performance improvement
Summary

We have many Challenges

Toward Lightning RubyVM
Thank you

Toward Lightning
RubyVM
RubyKaigi2010 Lightning Talk

Presented by
Koichi Sasada
ko1@rvm.jp
This presentation is 1st time about

- English presentation in RubyKaigi
- LT presentation in RubyKaigi
- No joking presentation
- Preparing script