RUBYVM DEVELOPMENT, NOW AND THE FUTURE

現在の、そしてこれからの RUBYVM開発

SASADA Koichi

Creative Informatics, University of Tokyo

Agenda

- Current Virtual Machine
  - Ruby 1.9.0-2
  - Roadmap to Ruby 1.9.1 (Stable?)
- Running Projects
  - Parallel Thread Execution
  - Multi-Virtual Machine
  - Compilation
  - Making Ruby for HPC
  - Atomic Ruby
    - Real-time Garbage Collection
  - Revisit Memory Allocation
- Summary

Current Ruby Interpreters

- Many Ruby Interpreters
  - Jruby
  - Rubinius
  - IronRuby
  - MagLev
  - And CRuby

Ruby 1.9.0-2 (Snapshot)

- Released! @ 2008-06-20
- Only Security Update (?)
- Snapshot Release
  - Some issues remains
  - **Need more testers!**
- Changes
  - BUG fixes
  - M17N fixes
  - Memory Allocation Options

Roadmap to Ruby 1.9.1 (Stable)

- 1.9.0-3 @ 2008-07-25
- 1.9.0-4 @ 2008-08-25
- 1.9.0-5 @ 2008-09-25 (feature freeze)
- 1.9.0-6 @ 2008-10-25 (1.9.1 RC1)
- 1.9.0-7 @ 2008-11-25 (1.9.1 RC2)
- 1.9.1 @ 2008-12-20 (and Happy X'mas)
- Release Managed by Yugui with Redmine
- Matz Admitted It!

BTW

- Yesterday, Matz SAID …

  “My Tech. Talk bores Audiences”,

- My Presentation will Show you …
  - only Tech. Talk about Ruby Interpreter
  - Tech. is too Exciting, at least for Me.
  - Not about Ruby, but about Lower-level Programming
  - “Small” Characters. Please use Opera Grasses

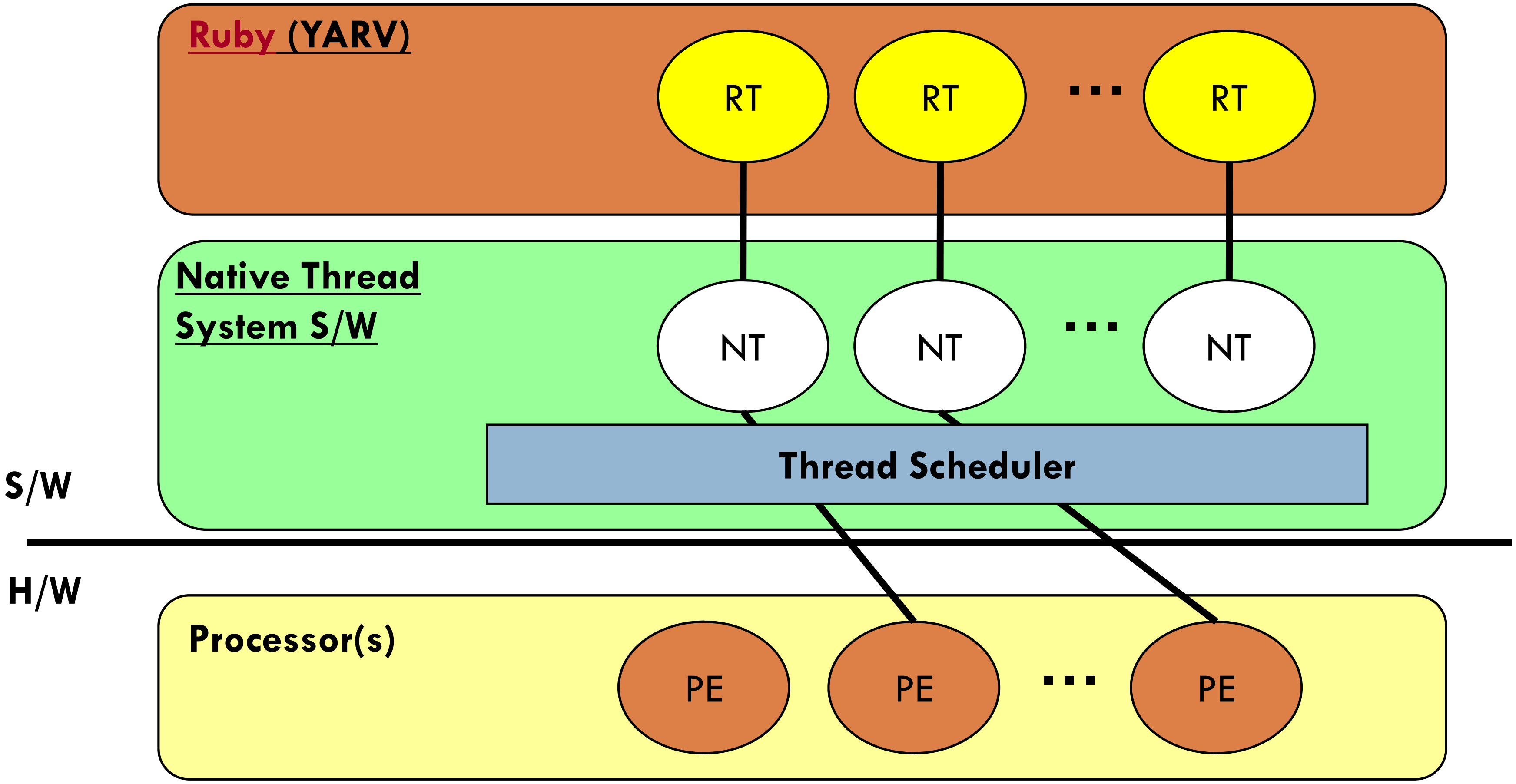
Running Projects

- Ruby Related Projects at The University of Tokyo
  - As My Research
  - As Student Research
  - As Student Exercise
  - And As My Hobby

Running Project
Parallel Thread Execution

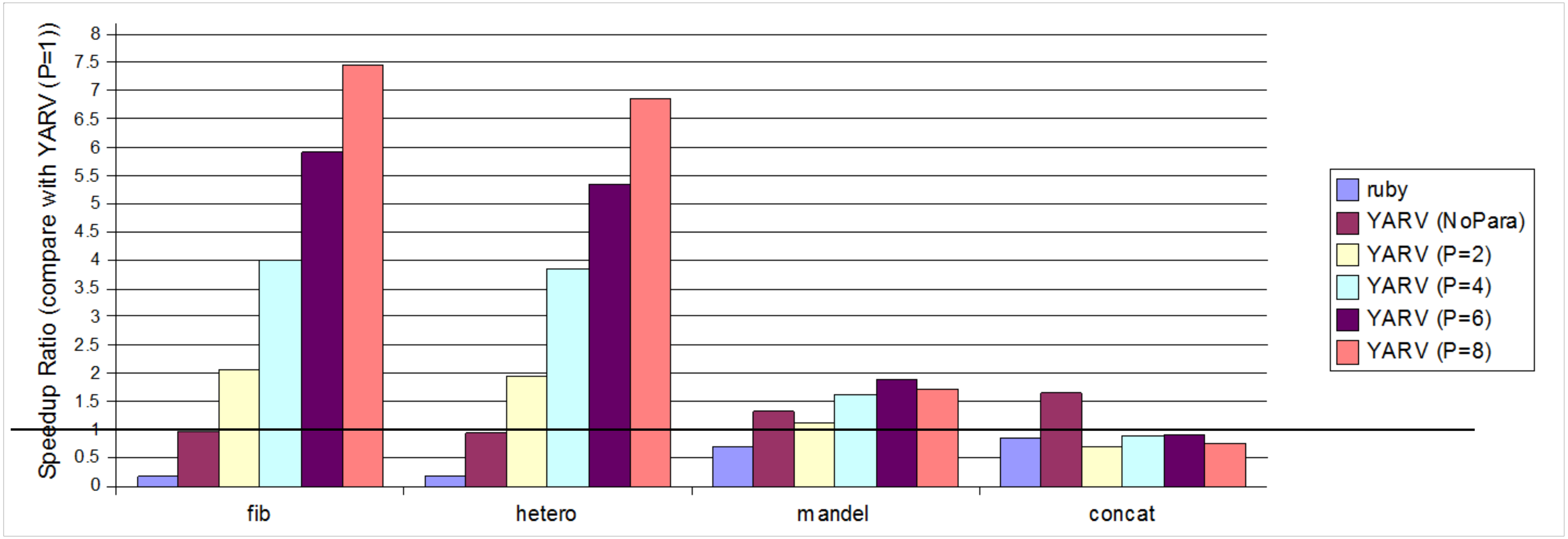
- Parallel Thread on Native Thread
- Current Ruby VM using Global VM Lock (GVL) to avoid Synchronization Issue
- Release GVL and Make Code Thread Safe
- Parallel Execution on Multi-Core Processors

Accepted Method:
Ruby Thread and Native Thread (1:1) ← Ruby 1.9/YARV



PE: Processor Element, UL: User Level, KL: Kernel Level

Evaluation
Result (Micro-benchmark)



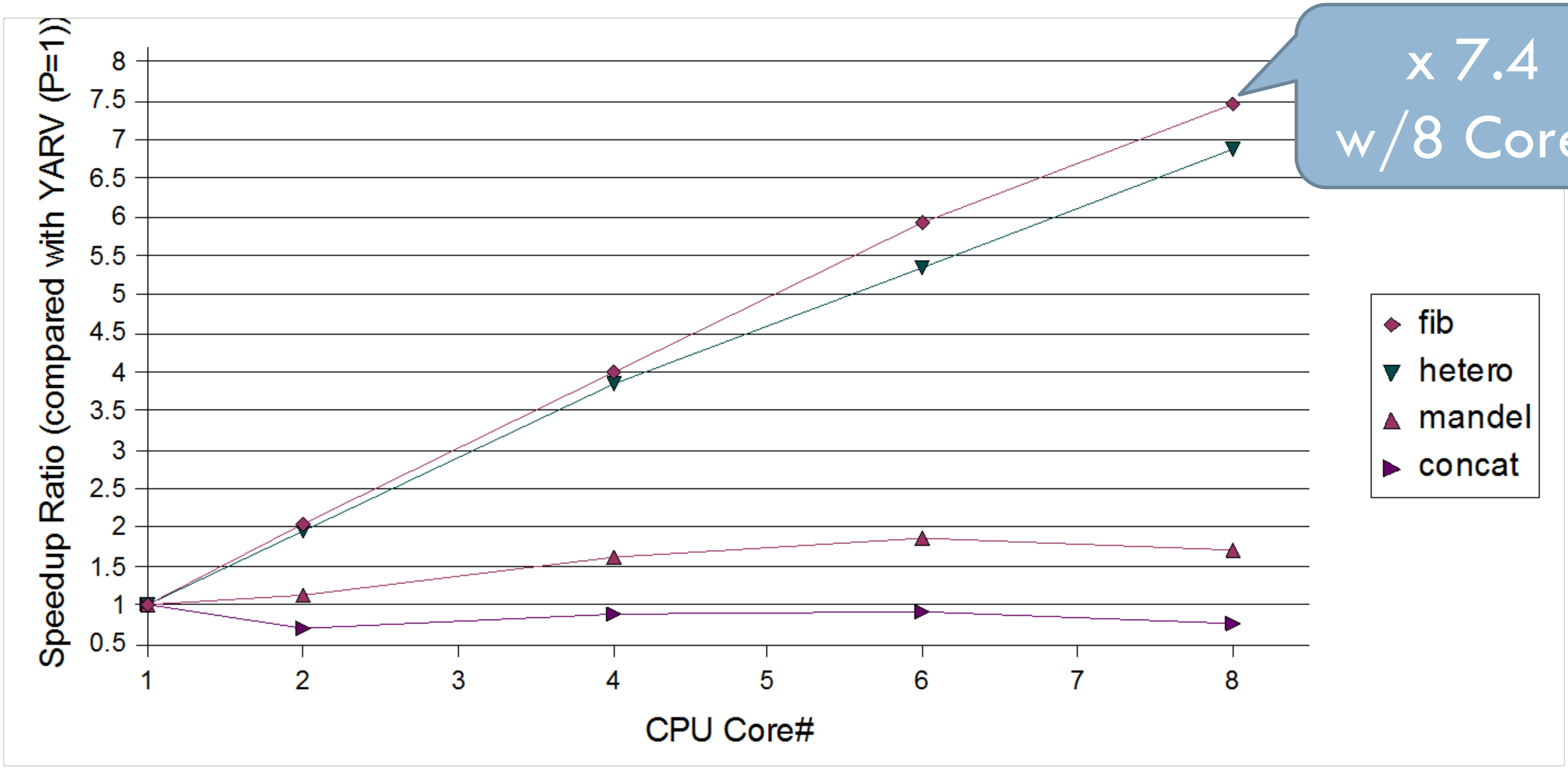
•fib: fib(N) (Make new Thread if N>30)

•hetero: fib + concat (1 thread)

•mandel: Mandelbrot (Big GC overhead)

•concat: String Concatenate (No Parallelism)

Evaluation
Result (Micro-benchmark)

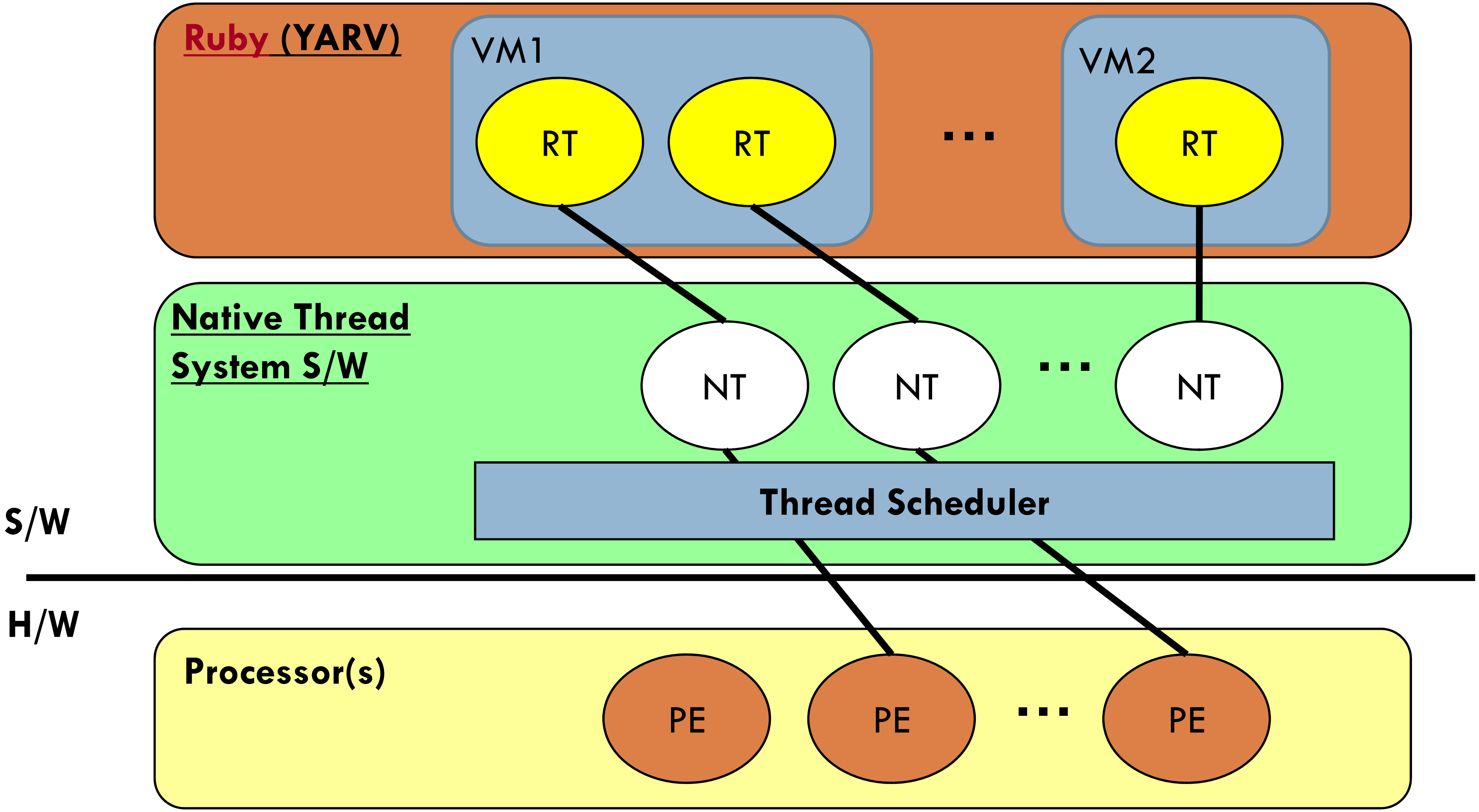


x 7.4
w/8 Cores

Running Project
Multi-VM

- Multiple Virtual Machine in same Process
- This helps …
  - Lightweight Interpreter (VM) Creation
  - Embedded Ruby Interpreter in Application
  - Parallel VM Execution
- Collaborative Research with Sun Microsystems
  - JRuby Team
  - Working with Nobuyoshi Nakada (PatchMon)

Multi-VM Overview



Ruby (YARV)

VM1

RT

RT

…

VM2

RT

Native Thread System S/W

NT

NT

…

NT

Thread Scheduler

S/W

H/W

Processor(s)

PE

PE

…

PE

PE: Processor Element, UL: User Level, KL: Kernel Level

Multi-VM Current Status

- API Design (Done)
- MVM Bootstrap (Done)
- MVM Creation (Doing)
- Documentation (Writing)

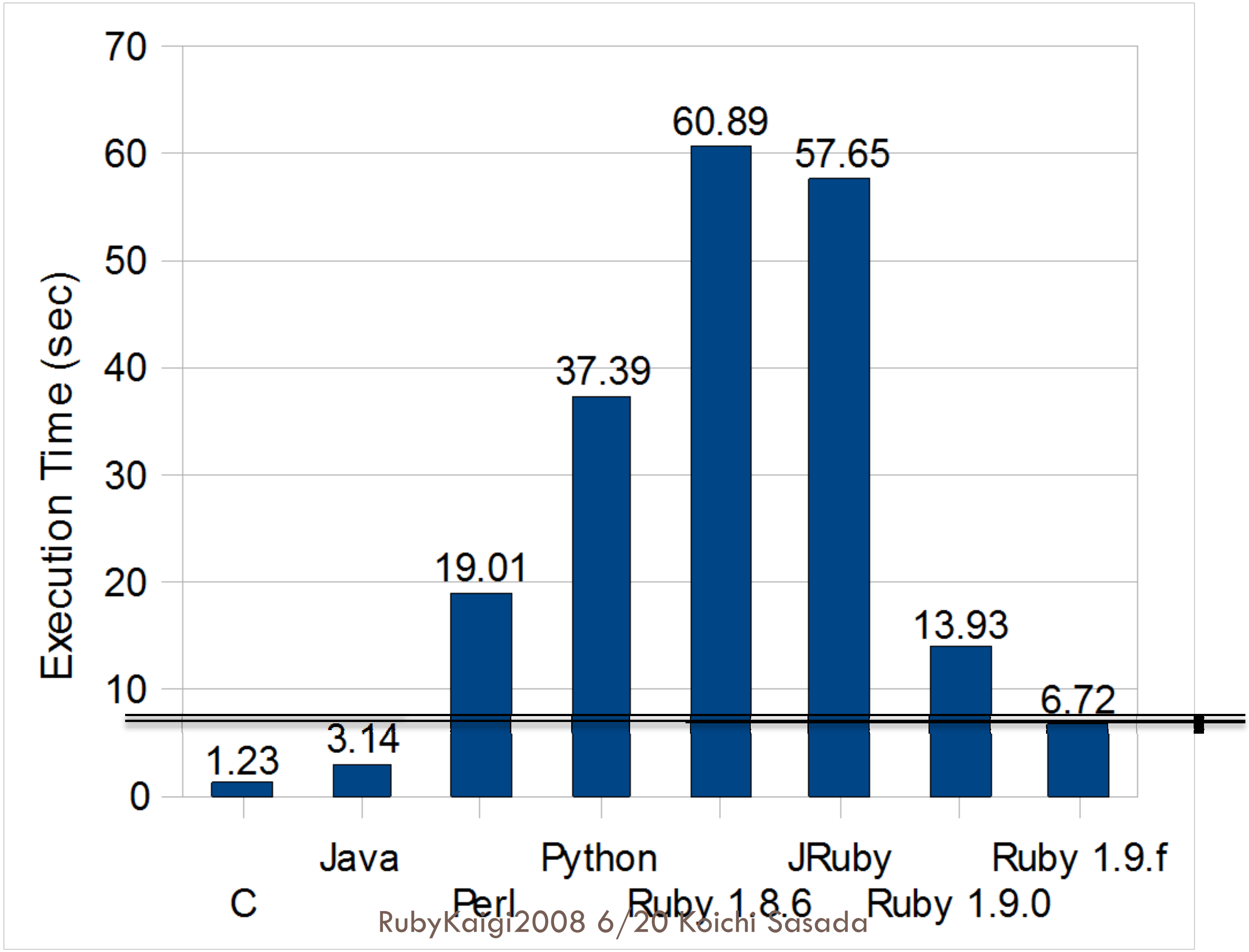
Running Project Compilation

- Ruby to XXX
  - Ruby/YARV BC to C (AOT) Compiler
  - Ruby/YARV BC to Native Code (JIT) Compiler
- Compilers improve Interpreter Performance
  - With Static Analysis
  - With Reduce Additional Instructions

Running Project
High Performance Computing on Ruby

- Ruby is Dynamic Language,
  - So It Hits Performance Improvement
  - But It Has High Productivity
- Parallel Computation Model on Ruby
- Improve VM Performance on Calculation
  - Floating Point Optimization for HPC

Evaluation of FP optimization Compared with Other Languages



Running Project
Atomic Ruby

- 「Only One Ruby in the World」
  「世界でたったひとつだけのRuby」
  - Previous Project Title: Lightweight-Ruby
- Exercise Program on IST Hands-on Program
  （情報理工実践工房）

Atomic Ruby (cont.)

- To Make a “Optimal Ruby Interpreter” for YOU!
- This helps …
  - Ruby on Embedded Systems
  - Porting Ruby Interpreter for rare Environment
  - Avoid Needless Security Issues
- 3 Students are Working
  - Plug-in/out Core Features
  - Bytecode serialization and Embedding
  - Customizable Garbage Collection

Running Project
Memory Revisit

- Revisit Memory Allocation
  - GC Algorithms
  - Malloc/Free Strategy
- Committed Exact Memory Usage Calculator
- Planning Sophisticated Memory Allocator

Running Project
New Garbage Collector

- Real-time Garbage Collector
  - Snapshot GC
  - Needs Write Barrier
- Partial Compaction Garbage Collector
  - Compaction with Conservative GC

CM: SASADA-Lab

- Student Recruitment!
  - **SASADA-Lab**, Department of Creative Informatics, Graduate School of Information Sci. & Tech, The University of Tokyo
  - **東京大学大学院情報理工学系研究科創造情報学専攻笹田研究室**
  - Search with “**笹田研**”
  - Launched at this Spring with 2 students
  - Important Date
    - **Application Period: 1st to 7th, July**

CM (cont.)

- Many Research Topics on SASADA-Lab are …
  - Ruby, Ruby, Ruby, PHP, Ruby, Ruby Ruby, Ruby, Ruby, Ruby, Ruby, Python, Ruby, Ruby, Perl, Ruby, Ruby, Ruby, Ruby, Ruby, Ruby, Ruby, Ruby, Lua, Ruby, Ruby, Ruby, Java, Ruby, Ruby, Ruby, Ruby, Ruby, Ruby, Ruby, Ruby, Ruby, Ruby, Implementation of Programming Language, Operating System / Processor Architecture, Ruby, Ruby, Software development, Ruby, Ruby, Ruby, Ruby, Ruby, Ruby, Ruby, Ruby, Ruby, Ruby, Ruby, Ruby, Ruby, Ruby
  - And such as Stimulable Ruby Projects

Summary

- Ruby 1.9.0-2 (snapshot) was released!
- Ruby 1.9.1 (stable?) will be released next X'mas!
- Many Exciting Ruby Projects

  Improve Performance and Functionality,
  - Parallelization, MVM, Compilation,
  - Revisit CRuby Architecture, Memory Strategy
- **SASADA-lab Application Period: 7/1～7/7**

Finished

ご清聴ありがとうございました

Thank you for your attention!

ささだこういち

SASADA Koichi

ko1@atdot.net