Faster android simulator

KangShuo(blackfin.kang@gmail.com)

Who: SkyEye Team

- The team to maintain SkyEye open source project
- The lovers of system software from Tsinghua Univ.
- The programmers use vi and emacs.
- The provider of commercial support in the simulator, binary translation etc.

Why: the importance of simulation

intel 超越未来™

ARM®

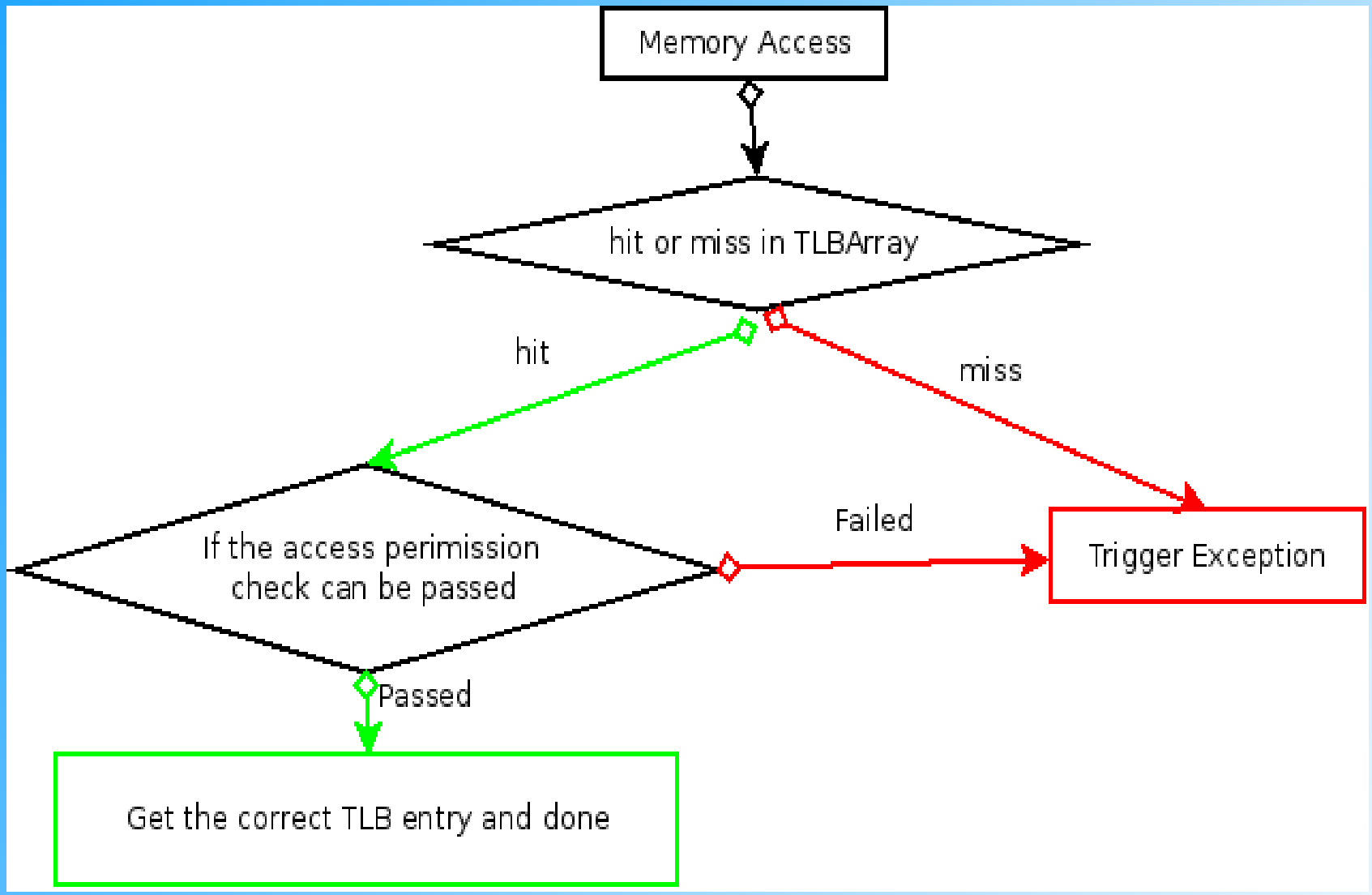
What: Key factors in fast full system simulator

- Memory simulation (Mainly MMU translation cost)
- Instruction execution (Binary Translation)
  - Qemu project
  - SkyEye project
- Peripheral simulation (Network, Display, etc...)
  - Libvirt project
  - GPU acceleration in google android simulator

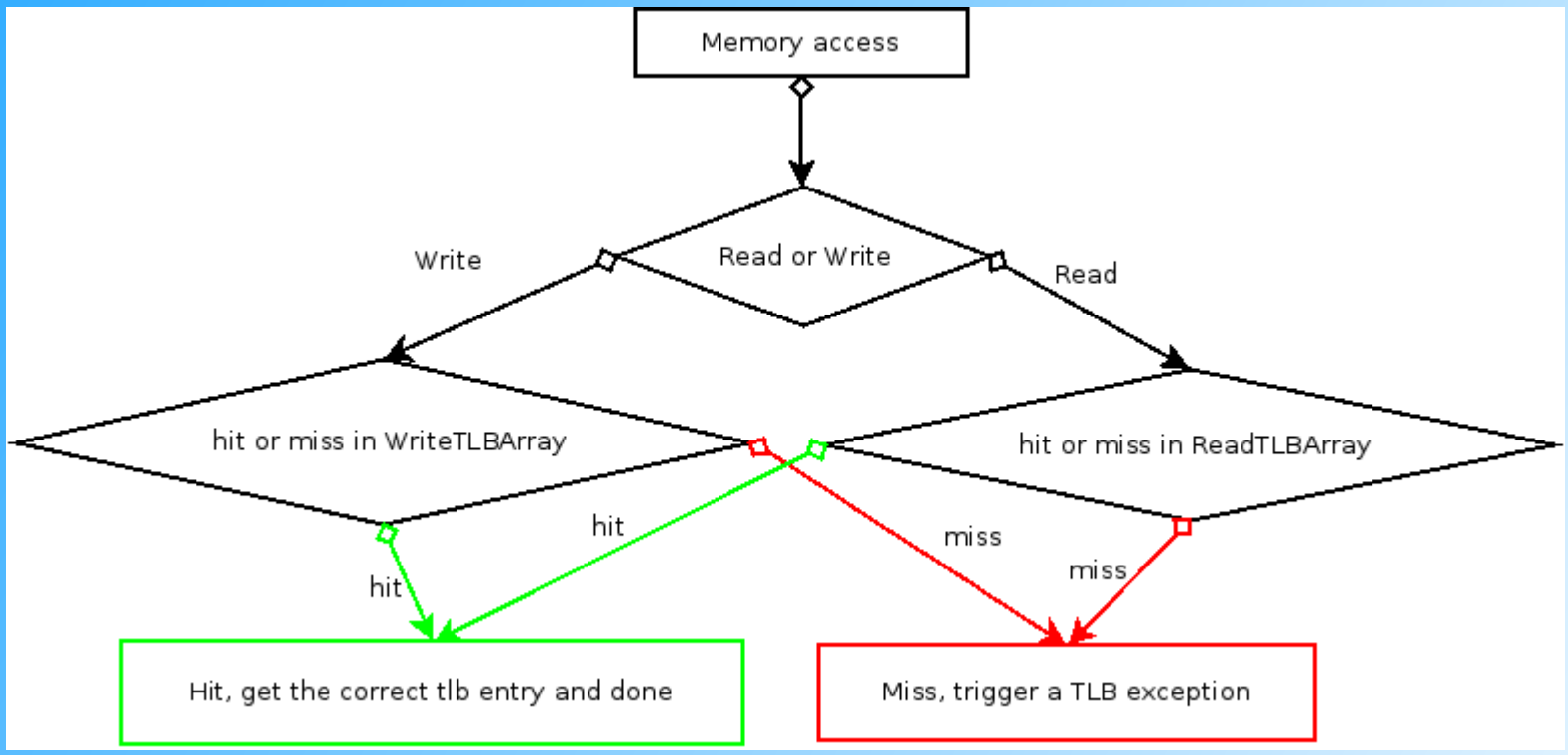
Memory Simulation

- Address translation in Full system simulator
  - GVA(Guest Virtual Address)
  - GPA(Guest Physical Address)
  - HVA(Host Virtual Address)
  - HPA(Host Physical Address)
- Big cost in address translation between GVA->HPA
  - TLB simulation is key factor for performance

About TLB simulation



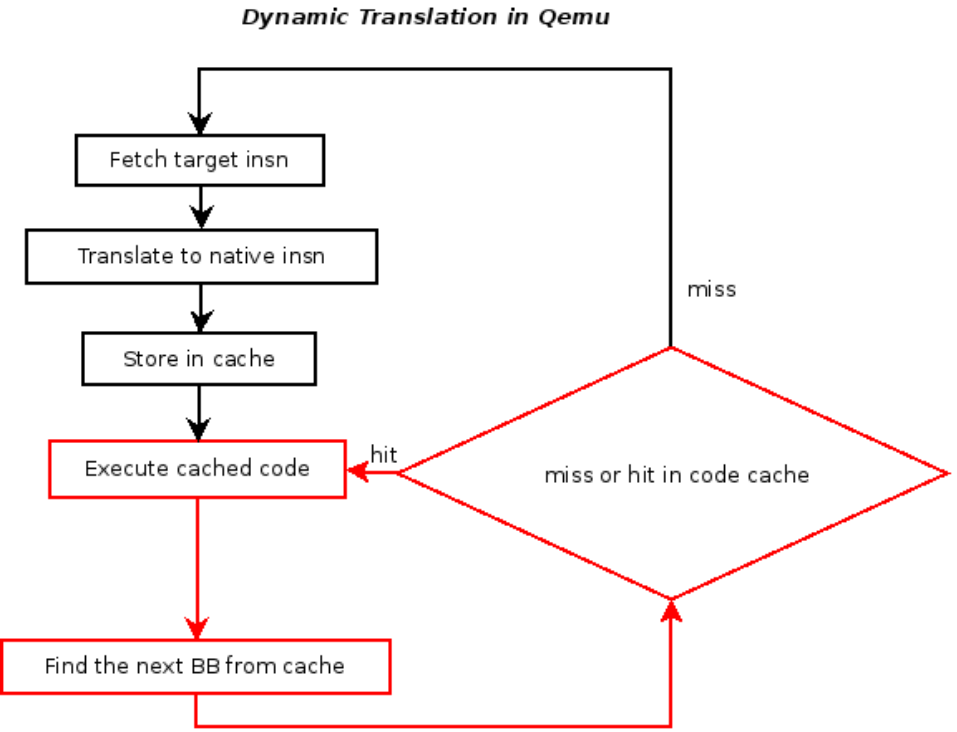
Qemu TLB methodology (SkyEye mimics Qemu)

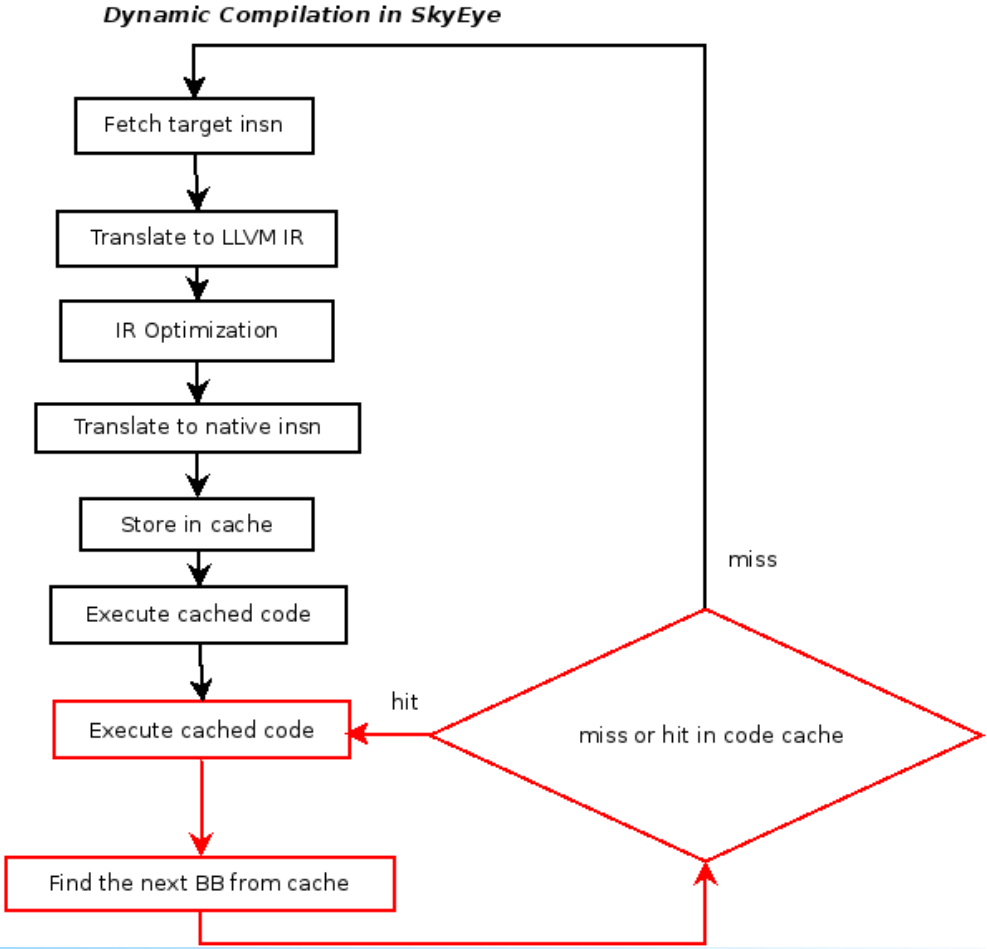


Now: how fast the technology of BT

- Some research results prove : BT can reach about 50%-60% of native ISA execution.
  - Use a lot of tricky and manual optimizations
- Arm simulation in Qemu is about 10%-15% of native ISA execution.
  - Less optimizations
- Arm simulation in SkyEye is about 20% - 30% of native ISA execution.
  - Use some compiler to do some optimizations

Dynamic Translation VS Dynamic Compilation

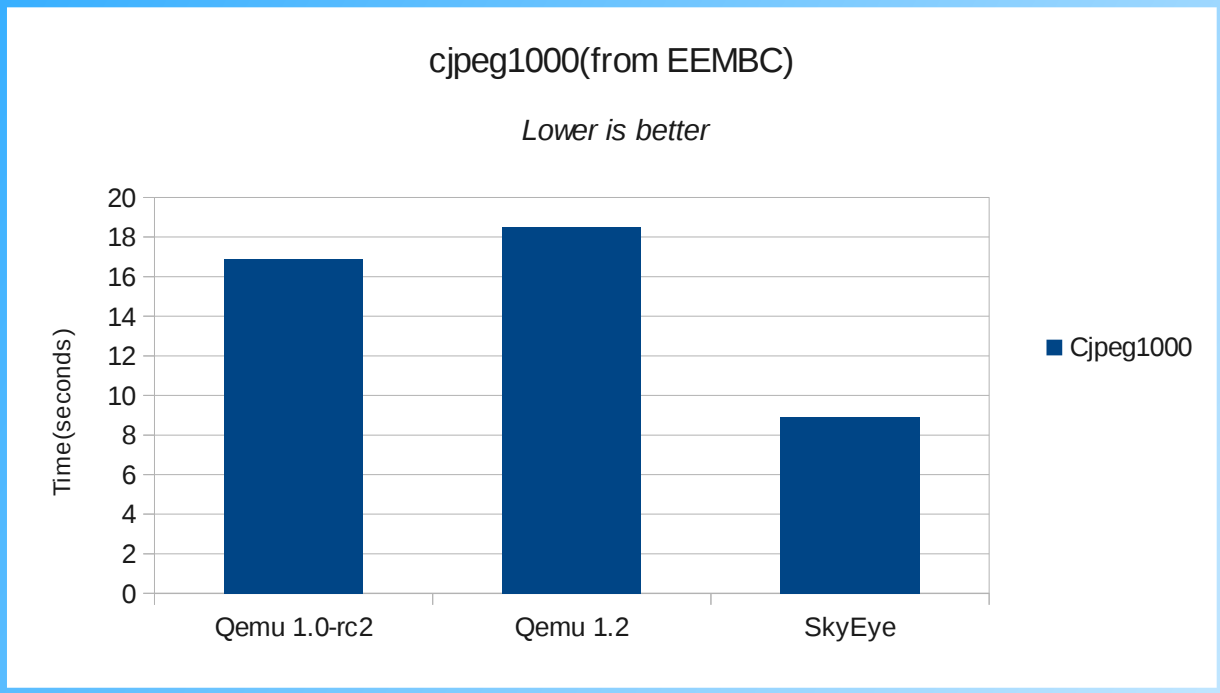




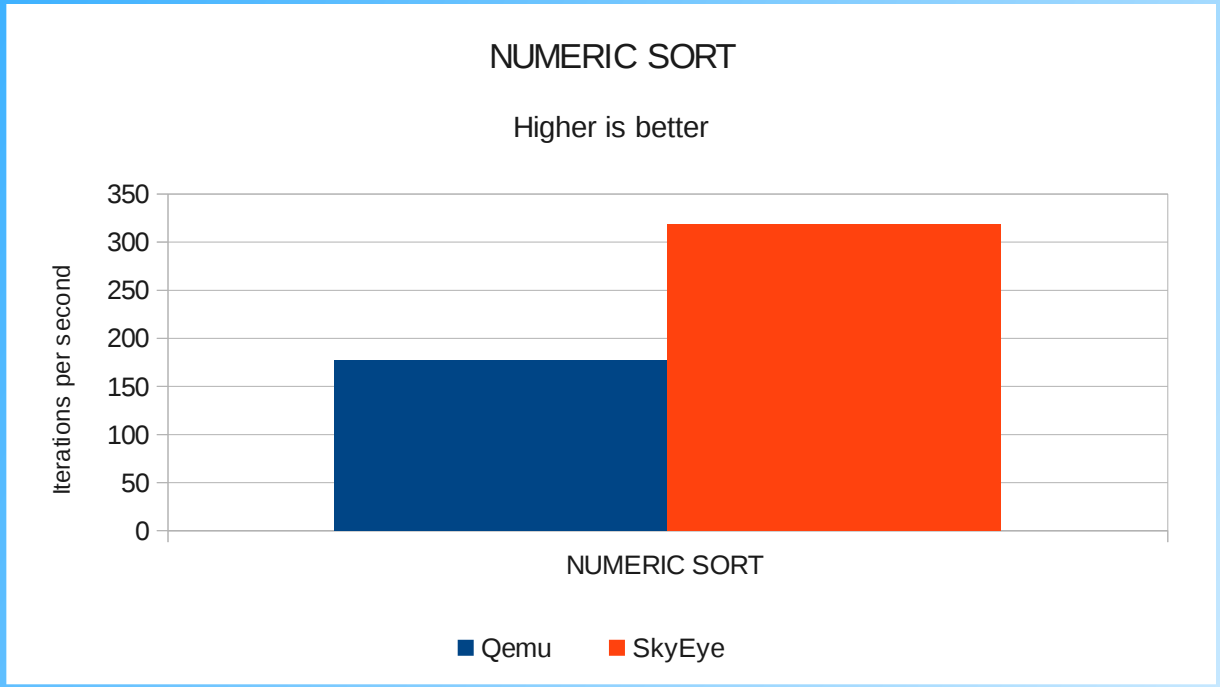
Benchmark Platform

- Benchmark platform:
  - Linux 3.1.0, OpenSuse 12.1
  - C compiler: arm-linux-gcc
  - libc: static
  - Processor:Intel(R) Core(TM) i7-2600K CPU @ 3.40GHz
  - Memory:16G

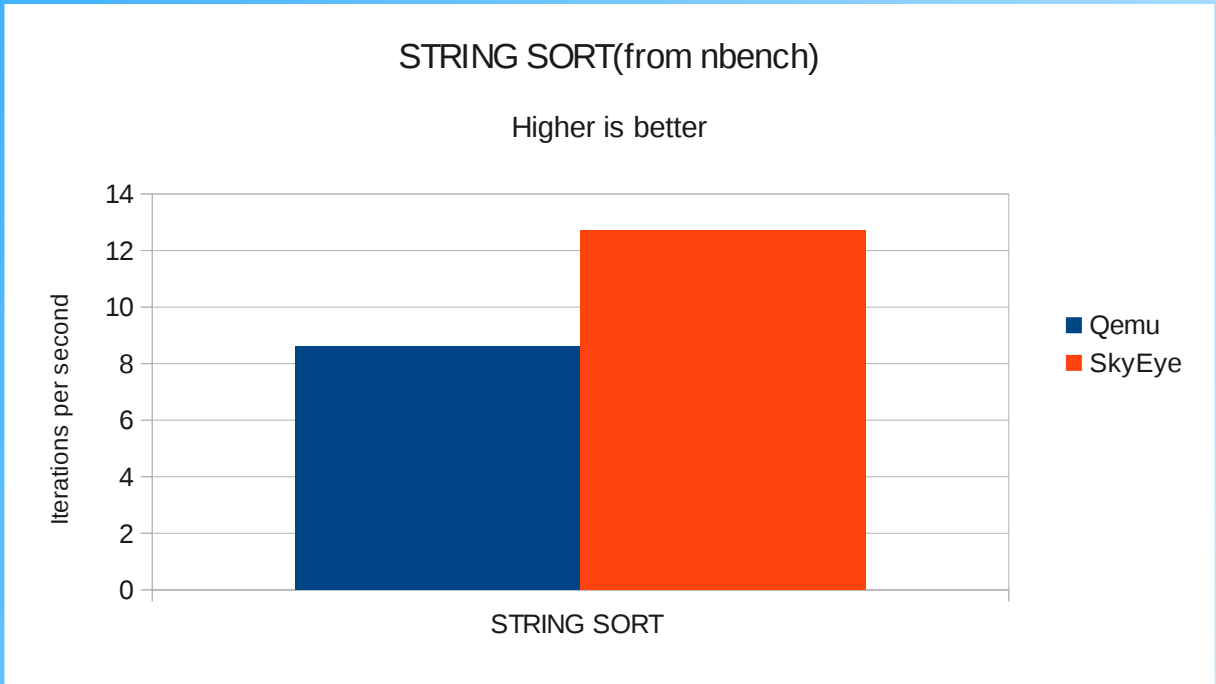
Cjpeg1000 benchmark under user mode



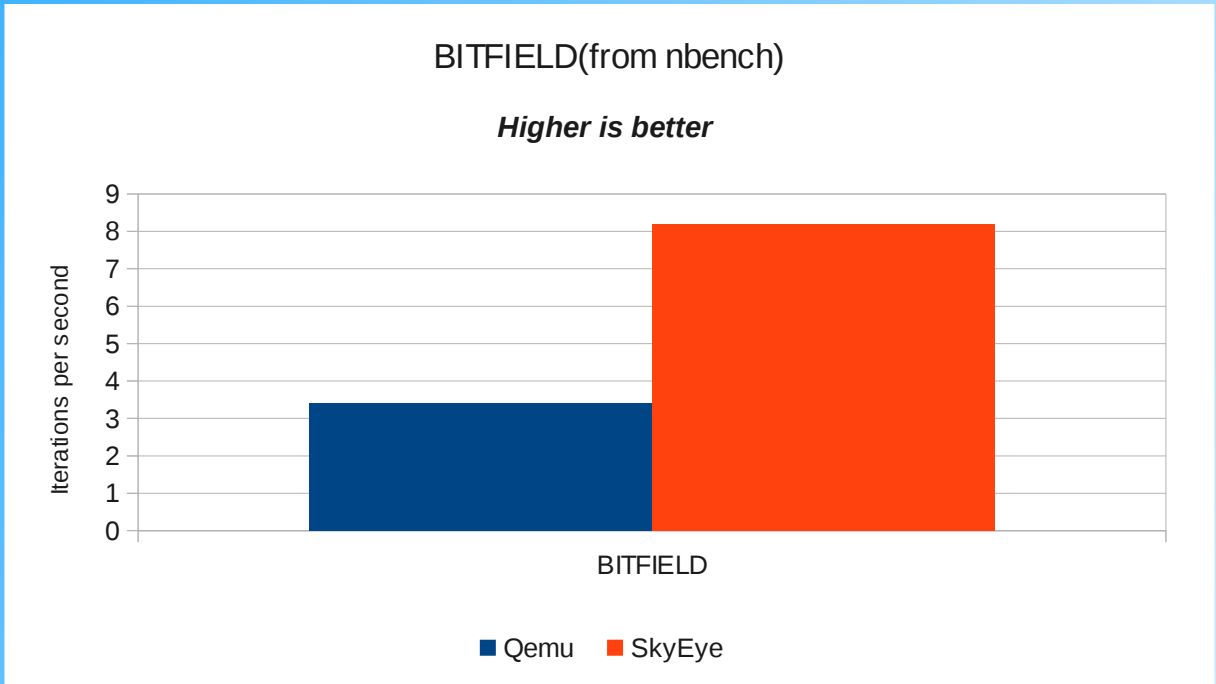
nbench benchmark under user mode



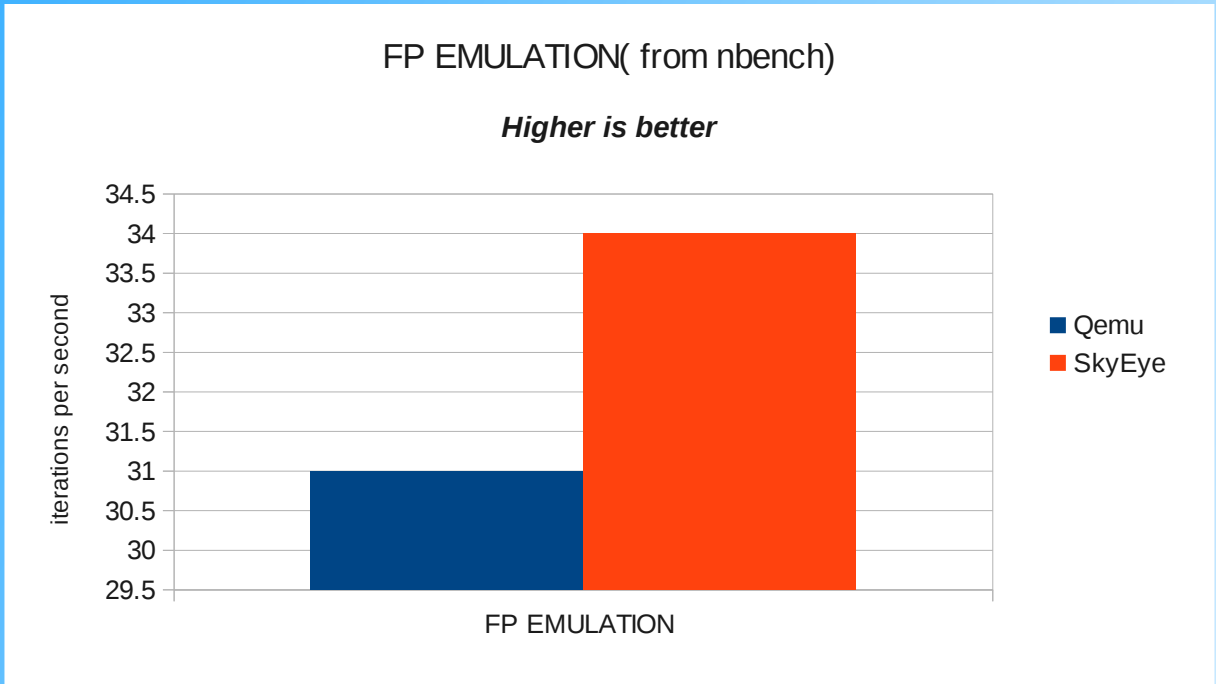
nbench benchmark under full sytem mode



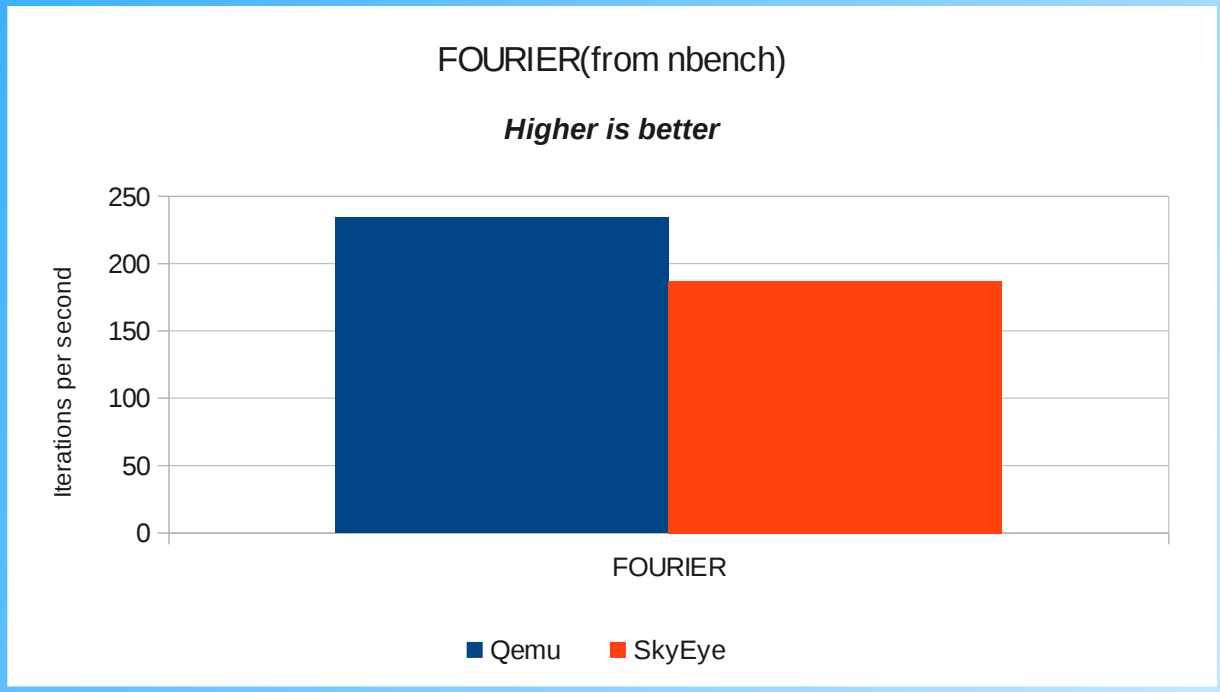
nbench benchmark under full sytem mode



nbench benchmark under full sytem mode



nbench benchmark under full sytem mode



The lie in benchmark result

(*So I am not responsible for any benchmark data in my slides, judge by yourself*)

- Benchmark Selection
  - Long-term running or short-term running
  - With or without float operation
  - IO intensive or Computation intensive
  - The same page or not due to the size of binary
- Your hardware
  - Burst technology in Intel processor.
  - Current load in your host
  - Multithread benchmark in multicore host

Superblock and BasicBlock

- The BasicBlock is one code block which have one single entry and one single exit
- The SuperBlock is composed of several related BasicBlock and have multiple entry and multiple exit.
- Qemu uses BasicBlock as the unit of translation, and SkyEye use SuperBlock as the unit of translation.

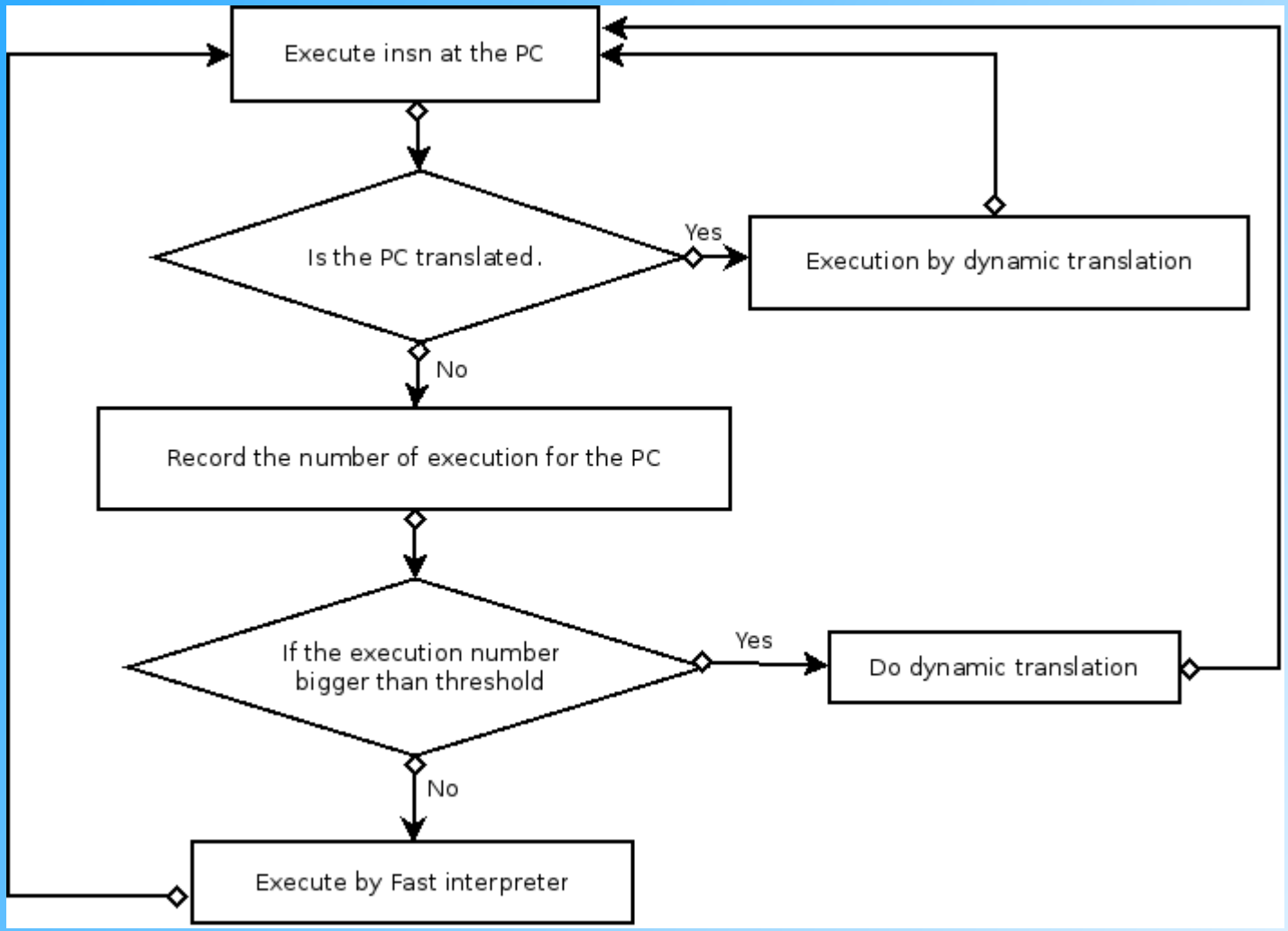
How to get a SuperBlock

- Some considerations
  - Big superblock will bring long translation time and large memory consumption
  - BBs in SuperBlock should have big affinity.
- Many ideas on the superblock.
  - Page region
  - Edge
  - Trace or Path
  - Process

Shortcoming of Dynamic Compilation and the solutions

- Shortcoming of DC
  - Slow warm-up of fresh code in Dynamic Compilation
  - More memory consumption
- Solutions
  - Use another interpreter of less optimization and faster translation speed.
  - Detect the hotspot in code execution and run the hotspot with Dynamic Compilation
  - Use multi-thread to do translation and execution in parallel.

SkyEye solution

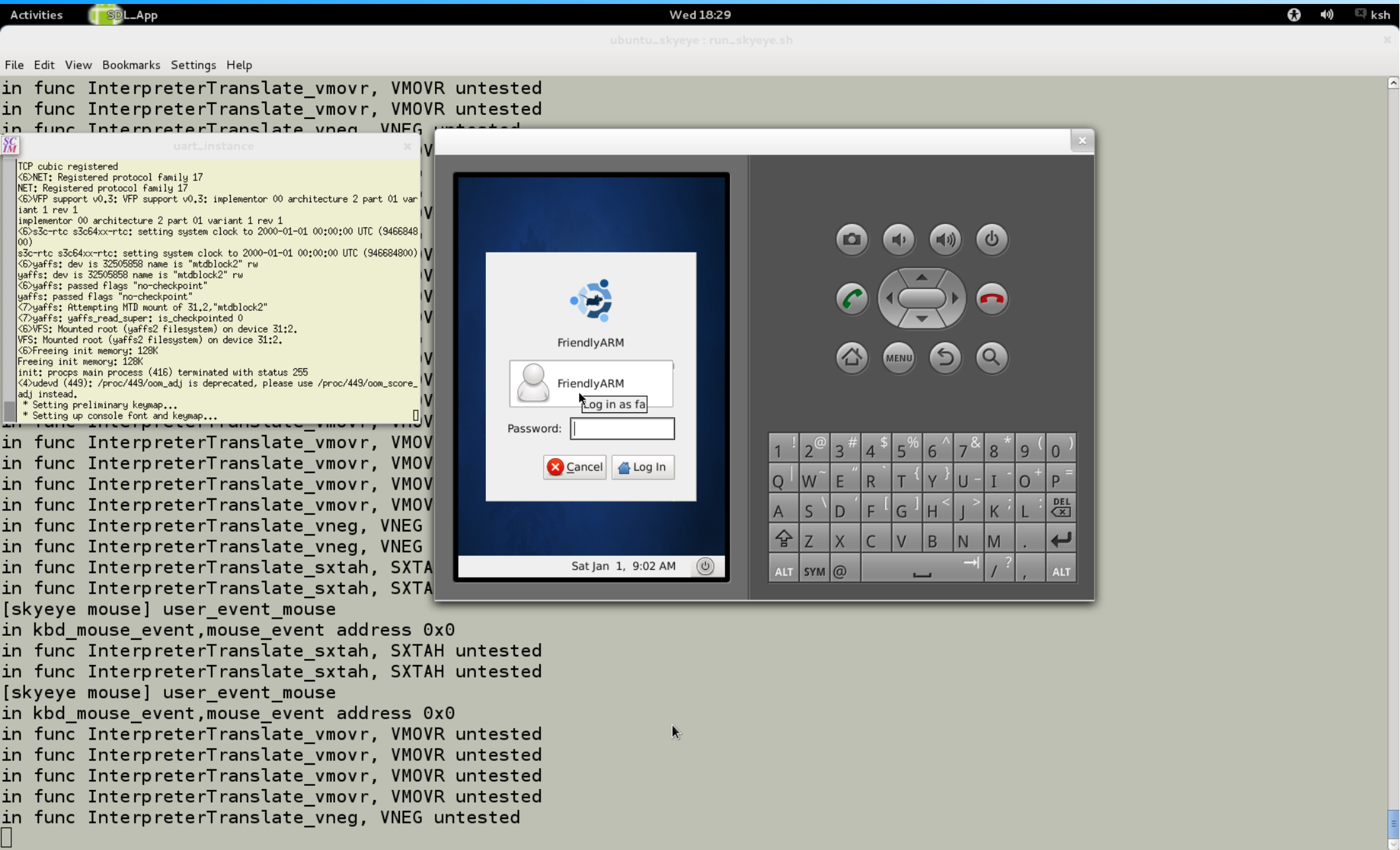


Current: SkyEye android simulator

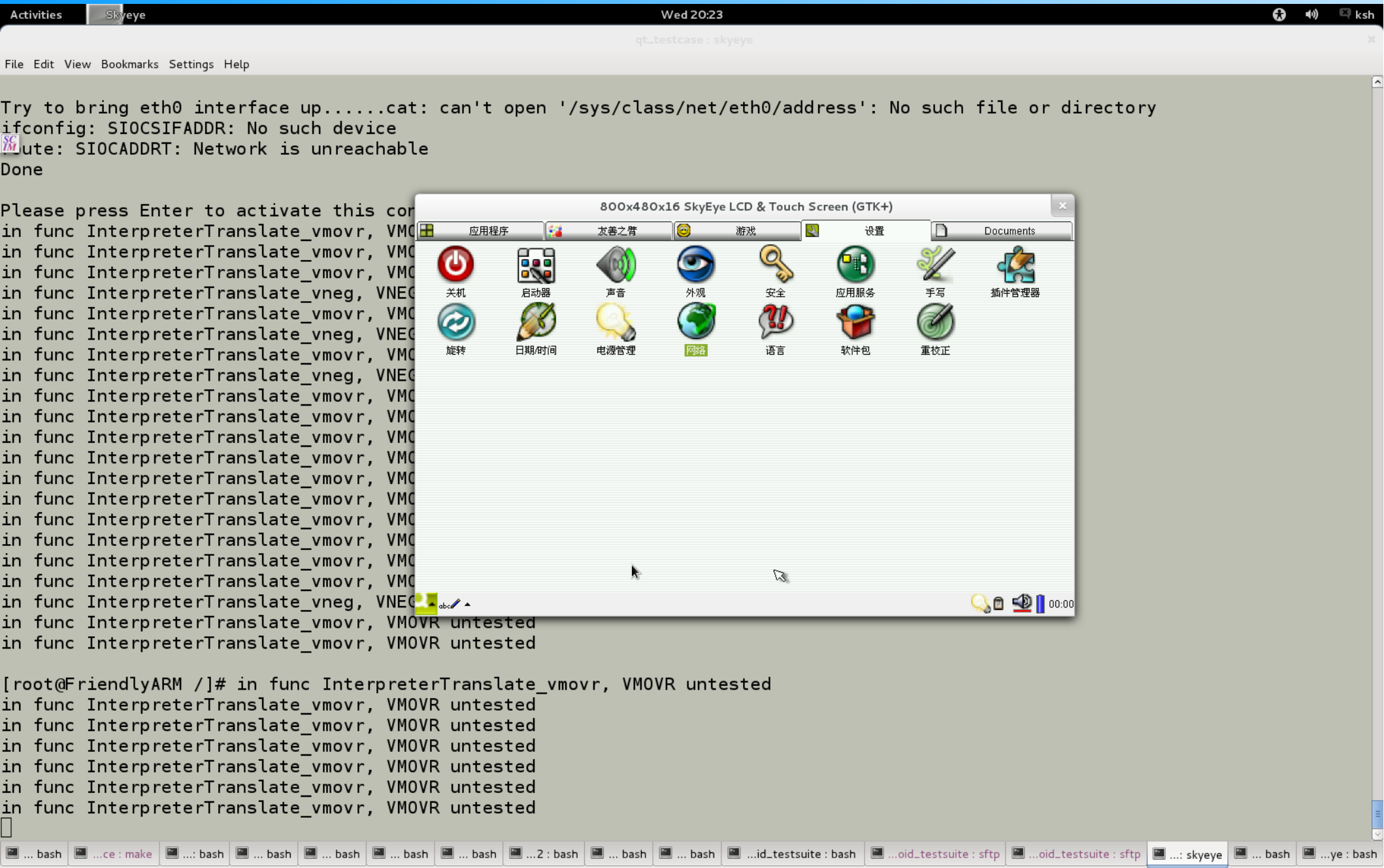
- Only support armv5.
- only support simulation of S3C6410.
- The support of VFP simulation is done.
- Play popular application, such as angrybirds, lianliankan etc.

Current: SkyEye android simulator

Current: Run ubuntu with SkyEye



Current: Run Qt with SkyEye



Future: the development plan

- The continuous speedup investigation of DC
- Support Goldfish simulation
- Support GPU acceleration
- Support armv7, Thumb-2, NEON instructions.

Some resource of SkyEye

- The release package and the testsuite package
- Some articles of SkyEye Dynamic Translation
- Wiki and maillist

Welcome contribute to SkyEye Project

Q & A
Thanks