TEMU 3 AND TEMU 4

HIGH PERFORMANCE GR740 VIRTUAL PLATFORM

DR. MATTIAS HOLM

Introduction

- TEMU 2:
  - Released in 2015.
  - Latest release: 2.3.5 in 2022
  - Features:
    - Supports ARMv7, PowerPC 32 and SPARCv8
    - Multicore emulation (on a single thread, round-robin)
    - ERC32, LEON2, LEON3, LEON4 (with peripherals)
    - APIs for I/O bus modelling (UART, Ethernet, CAN etc)
    - Bus models can often dump PCAP files for analysis in Wireshark
    - Lots of models of GRLIB peripherals
    - Memory SEU, MEU injection
    - Fully deterministic
  - Example usage with external customers and internally:
    - Real-time GR712RC simulator (both cores)
    - HIL (StarDundee adapter for SpW (Terma IPR), CAN bus bridge to USB-CAN adapter)
    - SVFs
    - OBSW development
    - ESTEC EagleEye SVF

TEMU 3

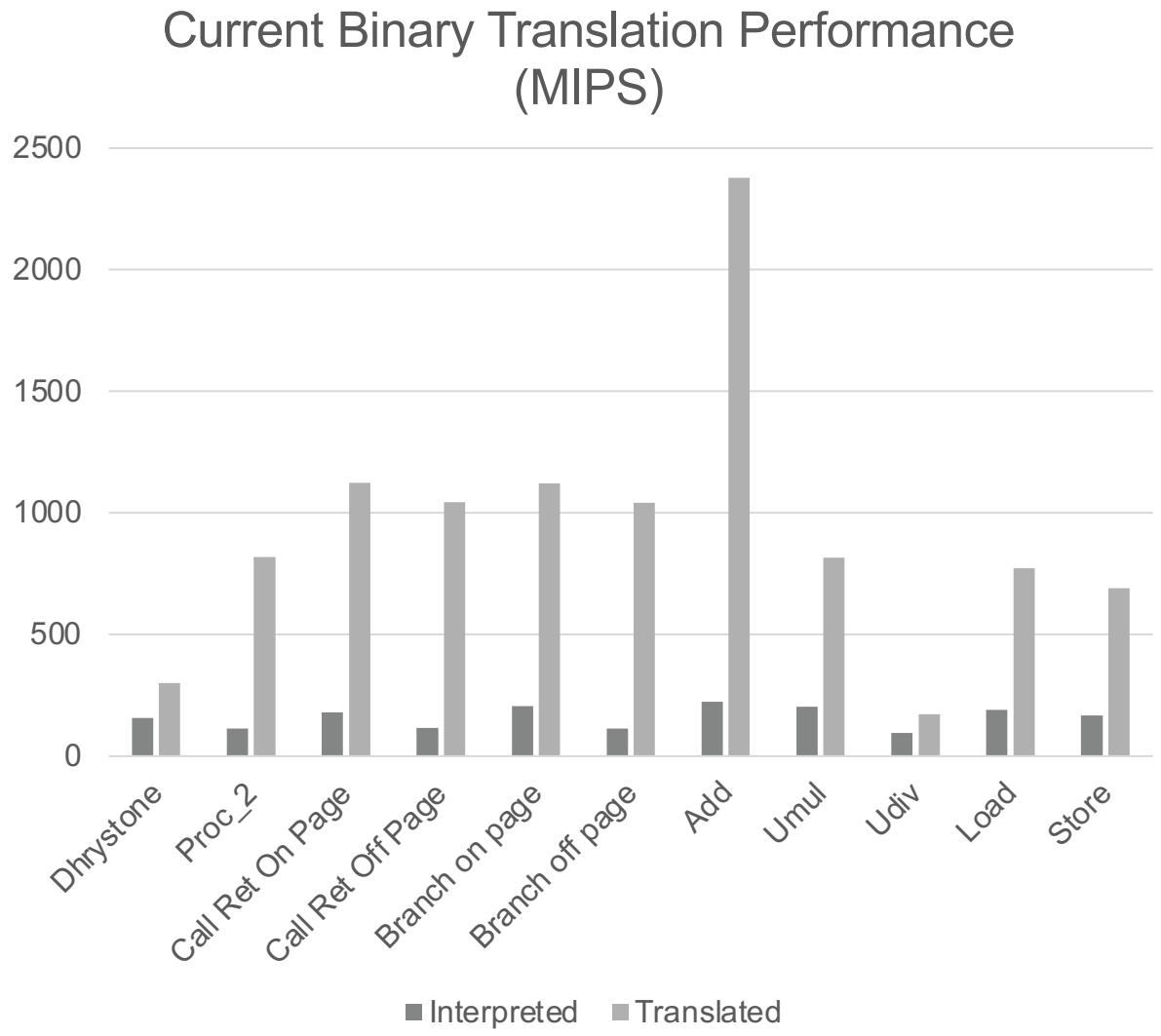
- TEMU 3 released 2022-11-11
- Several improvements in system:
  - New command line recursive decent parser replacing libedit tokenizer
    - Expressions
    - Custom commands
    - If statements
    - Etc
  - Performance:
    - New instruction set simulator interpreter mechanism (50 % performance gain)
    - Binary translator (100 % performance gain (on-top of TEMU3 interpreter))
    - Capable of emulating a single GR740 LEON4 core in real-time
  - Models and Buses
    - P2020 (E500 PPC core with vector instruction set, peripherals)
    - PCIe peripheral bus model
    - GR740 (GRIOMMU, IRQAMP, GRSPW2, SPWROUTER, GR1553B, GRCAN)
  - Other:
    - Multi-endian bus modelling (e.g. processor (big), PCIe (little))
    - SMP2 runtime library (run SMP models inside of TEMU)

TEMU 4 and Onward

- Under development:
  - Adds parallel multi-core scheduler (prototype exists that works)
  - Will run quad-core GR740 in real-time (given enough host cores)
  - More models
  - SystemC runtime library (run SystemC models inside of TEMU, prototype exist that works)
  - RISC-V support?
  - Improved software debugging support
    - E.g. better scripting, automatic context switching, etc.
  - Likely higher OS requirement (moving to C++20)

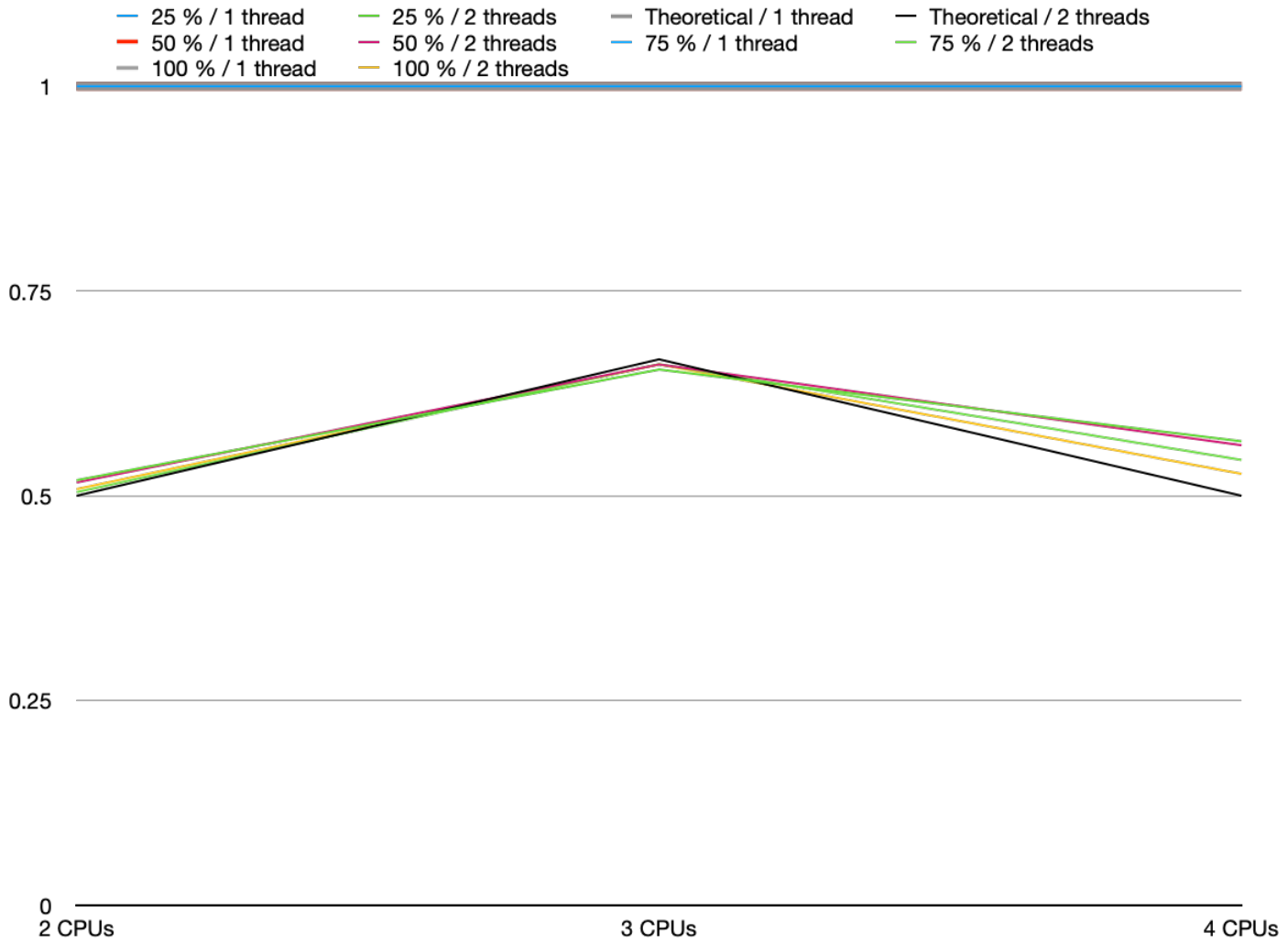
TEMU 3: Performance

- Currently almost 100% faster than interpreter
- Some benchmarks run significantly faster
- Data from:
  - MacBook Pro (2019)
  - 2.3 GHz 8-core Intel Core i9
  - 32 GiB 2667 MHz DDR4 RAM



TEMU 4: Preliminary Performance

- Test Case
  - 4 simulated cores
  - Some cores idle (see x-axis)
  - A certain fraction of idle time
  - 1 or 2 threads
- Graph shows normalised time, smaller is better
- Close to theoretical speedup (without synchronisation overhead)
- Note, setting CPU affinity per thread will be important for end user.



TEMU Virtual Platforms

- TEMU is pre-configured with the following SPARC platforms:
  - ERC32
  - AT697F (LEON2)
  - UT699 (LEON3)
  - UT700 (LEON3)
  - GR712RC (2xLEON3)
  - GR740 (4xLEON4)
- Complete systems simulated, including buses: serial, GPIO, 1553, Ethernet, SpW, CAN, etc.
- API for buses is straight forward => easy to implement remote terminal models.
- User can add to, remove from and re-configure the built-in configurations, or create a new one.
  - Want more RAM, change the RAM model allocation.
  - Want to add an extra UART, just create and map to memory in two lines.
  - Want to change the clock frequency, go ahead.

TEMU Models

- The following buses are modelled in TEMU out of the box:
  - AMBA (with PNP)
  - CAN
  - Ethernet
  - MIL-STD-1553
  - PCIe
  - Serial
  - Signal (e.g. used for GPIO)
  - SpaceWire
  - SPI

User can assemble their own virtual platform configuration from bundled models.

For an up-to-date list see: https://temu.terma.com/features.html

- SPARC Related Models
  - MEC (ERC32)
  - LEON2 on-chip devices
  - GRLIB
    - AHBCTRL
    - AHBSTAT
    - APBCTRL
    - APBUART
    - B1553BRM
    - CAN_OC
    - CLKGATE
    - FTMCTRL
    - GPTIMER
    - GR1553B
    - GRCAN
    - GRETH
    - GRETH_GBIT
    - GRGPIO
    - GRIOMMU
    - GRSPW1
    - GRSPW2
    - IRQAMP
    - IRQMP
    - L2CC
    - MCTRL
    - SPWROUTER

Software Debugging

- Built-in GDB RSP server.
  - Each simulated core mapped to a GDB thread
  - Non-intrusive
- Built-in assembler level debugger
- Basic DWARF / source level debugging support (breakpoints by lines, function names etc), with multiple contexts
- Focused on command line to improve programmer productivity when debugging interactively at source level
- Debug multiple contexts/applications at the same time:
  - Boot Software
  - OS kernel
  - Applications
- Other features can be added if requested

```
temu> exec gr740.temu
…
temu> load obj=mem0 ctxt=ctxt0 file=${TEST_DIR}/test.elf
temu> ctxt0.remap-path from="/build/testsw" \
                       to="/home/jdoe/Projects/testsw"
temu> ctxt0.break loc=test.c:16
Breakpoint 1 set at test.c:16 @ 0x400012e8

temu> machine0.run
Breakpoint 1 hit by cpu0 at /home/jdoe/Projects/testsw/test.c:16
  (main) (VA = 0x400012e8 PA = 0x400012e8)

temu> ctxt0.pvar 'a'
a = 5

temu> ctxt0.list-variables
param: a : int
param: b : int
variable: c : int
```

Control and Integration

- User Interaction with TEMU
  - Command line, using the “TEMU scripting language”
  - C-API for writing models and controlling TEMU
    - Publish models and their state (snapshots will work out of the box)
  - GDB RSP
- Embedding
  - TEMU is designed for embedding in other programs using the C-API
  - Main command line driver is just a program linking to the TEMU runtime libraries
  - Example:
    - Embedding in an SMP based simulator (SimSat, etc)
- I/O models:
  - Implement TEMU model handling the memory transactions, and forward calls to external models (e.g. SMP models).

QUESTIONS

Please Visit:

- https://tgss.terma.com/
- https://temu.terma.com/

Contact:

- Sales: Mr. Roger Patrick <rmp@terma.com>
- Technical: Dr. Mattias Holm <maho@terma.com>

Find us at www.terma.com and

LINKEDIN
Terma Group

INSTAGRAM
Terma Group

TWITTER
@Terma_Global

YOUTUBE
Terma Group