Invitation to MARS

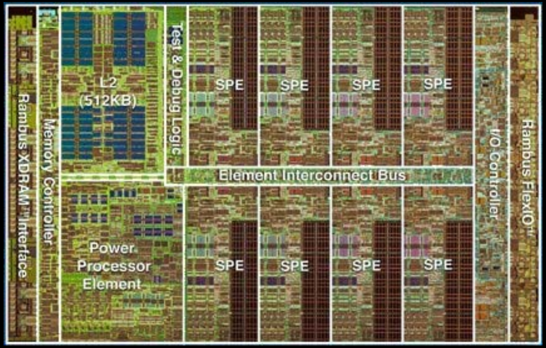
Multi-core Application Runtime System

2008.12.18

Sony Computer Entertainment Inc.

Preparation for MARS

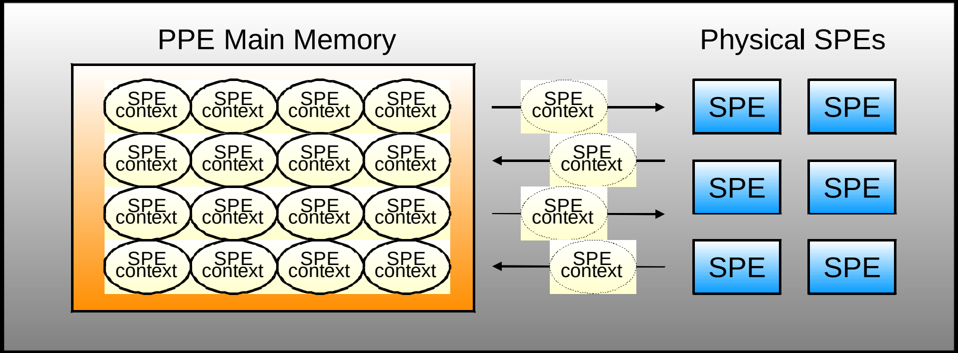
Cell Broadband Engine



- 1 PPE: general-purpose
  - PowerPC architecture
  - System management
- 8 SPEs: between general-purpose and special-purpose
  - SIMD capable RISC architecture
  - Programs and data must be located on 256KiB local storage
  - External data access by DMA via MFC
  - Workloads for game, media, etc

SPUFS - Linux support for Cell/B.E.

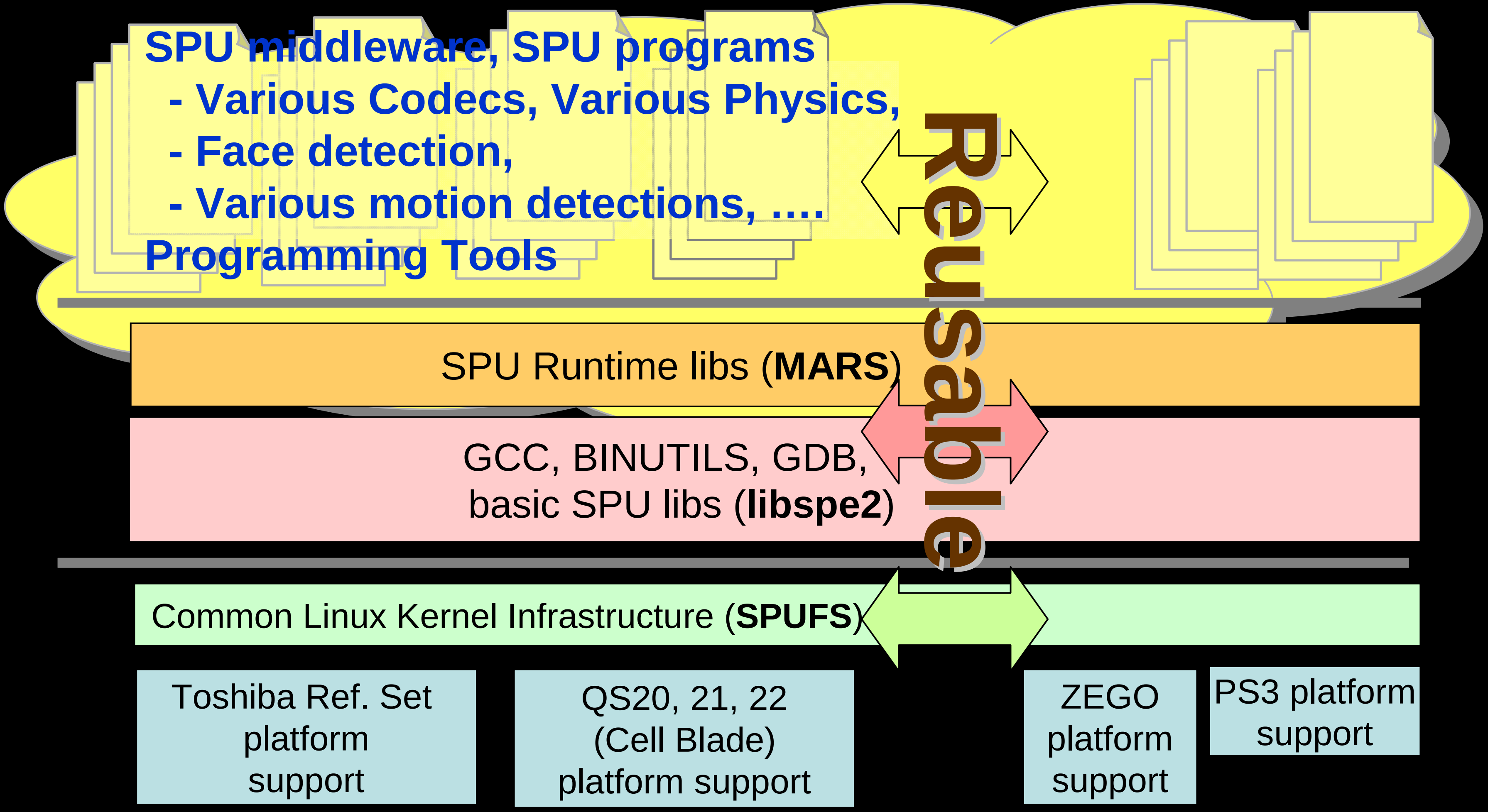
- Low level SPE control interfaces are provided as system calls and the SPUFS virtual file system
- Physical SPEs are virtualized and each SPE instance is managed as an SPE context
  - Each context is stored in and restored from the main memory by the SPE scheduler
  - Potential for high switch overhead when more contexts than physical SPEs



Cell/B.E. Programming Environment

- PPE toolchain
  - One of PowerPC targets
  - gcc and binutils with Cell/B.E. specific enhancements
- SPE toolchain
  - New target architecture
  - spu-gcc, binutils, newlib (libc), ...
- libspe
  - SPE management library
  - Provides OS-independent API by wrapping the raw SPUFS interfaces

Cell/B.E. Programming Environment



SPU middleware, SPU programs

- **Various Codecs, Various Physics,**
- **Face detection,**
- **Various motion detections, ….**

Programming Tools

SPU Runtime libs (**MARS**)

GCC, BINUTILS, GDB,
basic SPU libs (**libspe2**)

Common Linux Kernel Infrastructure (**SPUFS**)

Toshiba Ref. Set platform support

QS20, 21, 22 (Cell Blade) platform support

ZEGO platform support

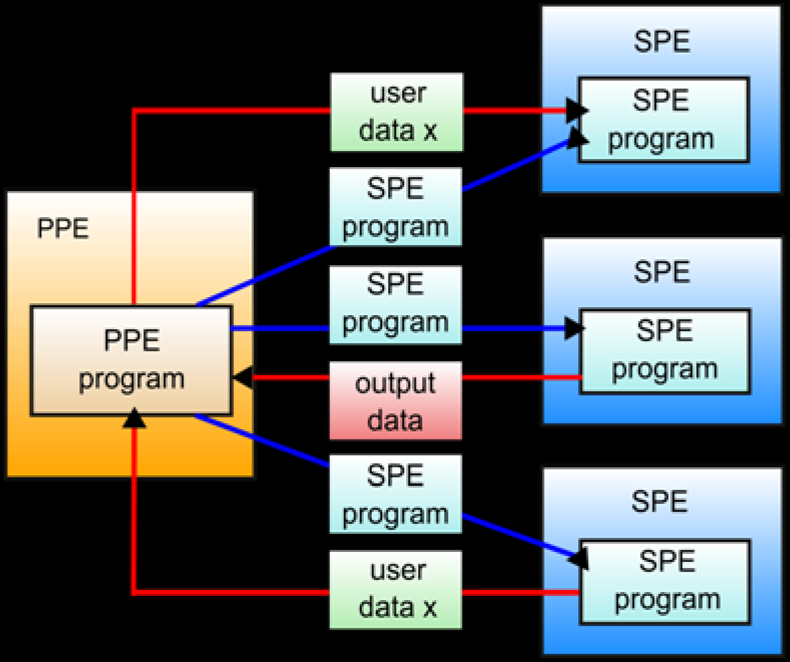
PS3 platform support

Typical Cell/B.E. Program

- 1 PPE program
  - User interface
  - Data input/output
  - Loading and executing SPE programs
- Multiple SPE programs
  - Image processing
  - Physics simulation
  - Scientific calculation

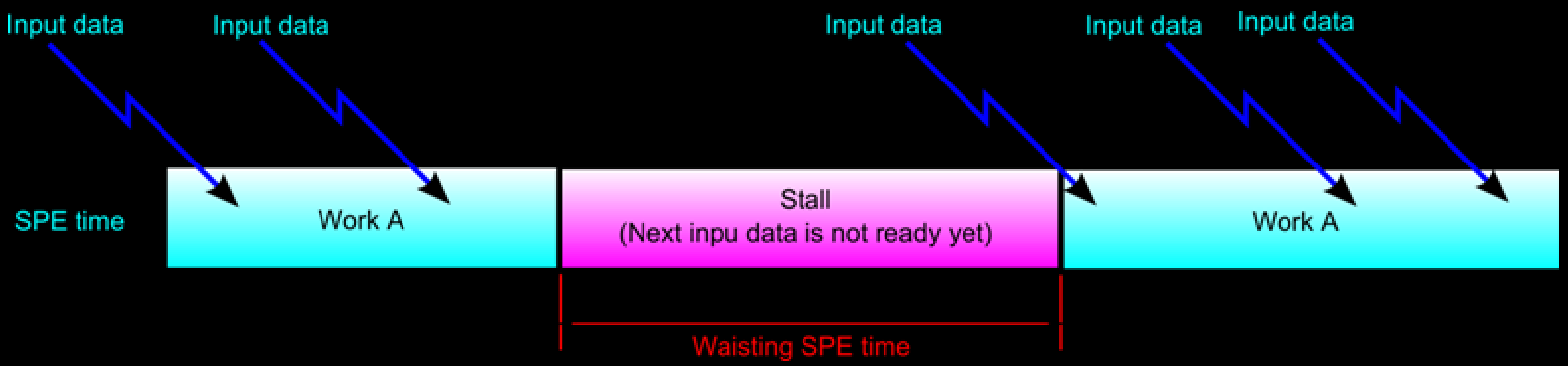
PPE Centric Programming Model

- PPE is responsible for:
  - Loading/switching of SPE programs
  - Sending/receiving of necessary data to its SPE programs



Problems of PPE Centric Programming

- Difficult for the PPE to know SPE's status
  - Stalls, waits...
  - Inefficient scheduling of SPE programs



- Extra load of the PPE
  - Communication
  - Scheduling

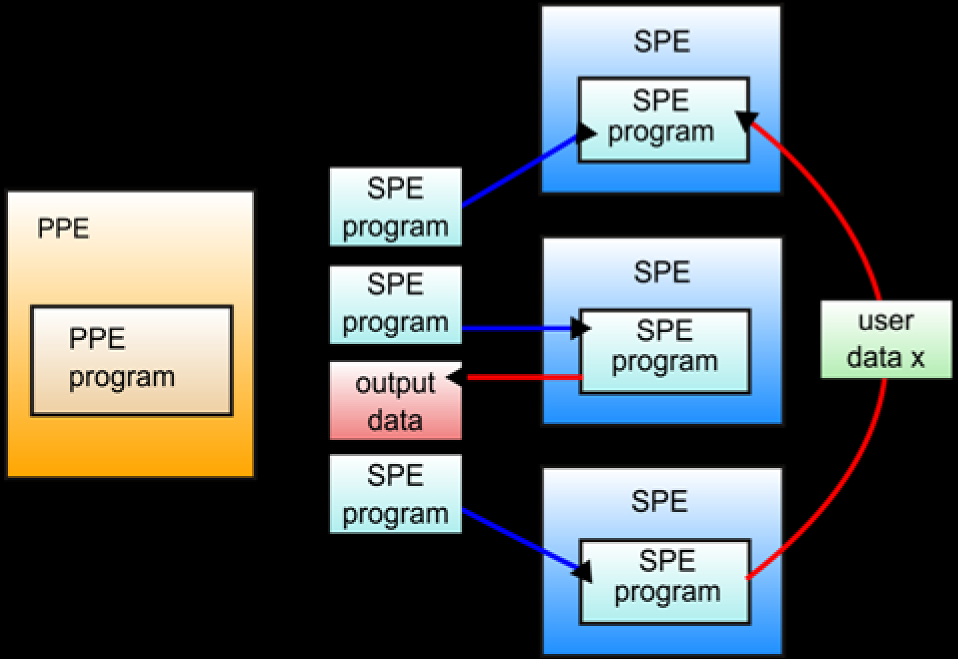
What is MARS?

MARS

- MARS stands for Multi-core Application Runtime System
- Provides efficient runtime environment for SPE centric application programs

SPE Centric Programming Model

- The individual SPEs are responsible for:
  - Loading, executing and switching SPE programs
  - Sending/receiving data between SPEs

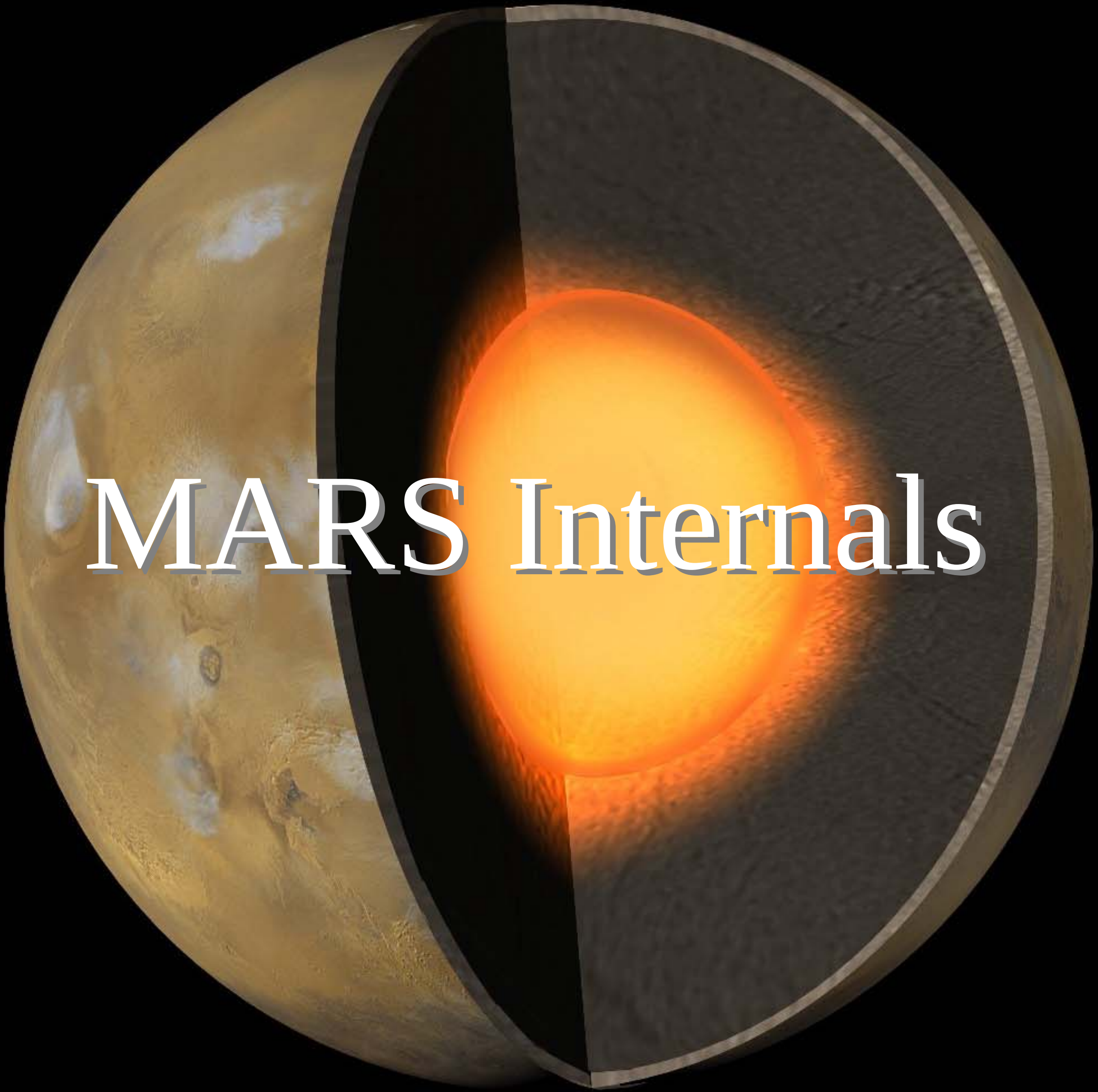


What MARS Provides

- SPE centric runtime:
  - Scheduling workloads by SPEs
  - Lightweight context switching
  - Synchronization objects cooperating with the scheduler

MARS Advantages

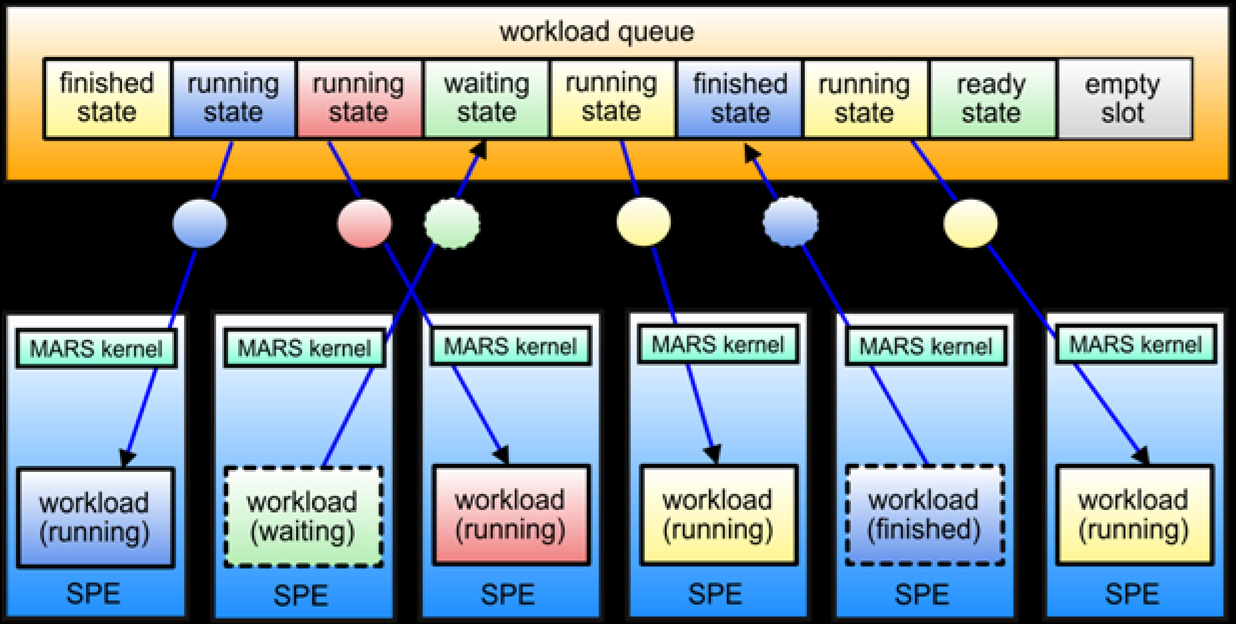
- Simplifies maximizing SPE usage
  - Efficient context switching
  - Minimizes data exchanged with PPE
- Minimizes runtime load of the PPE



MARS Internals

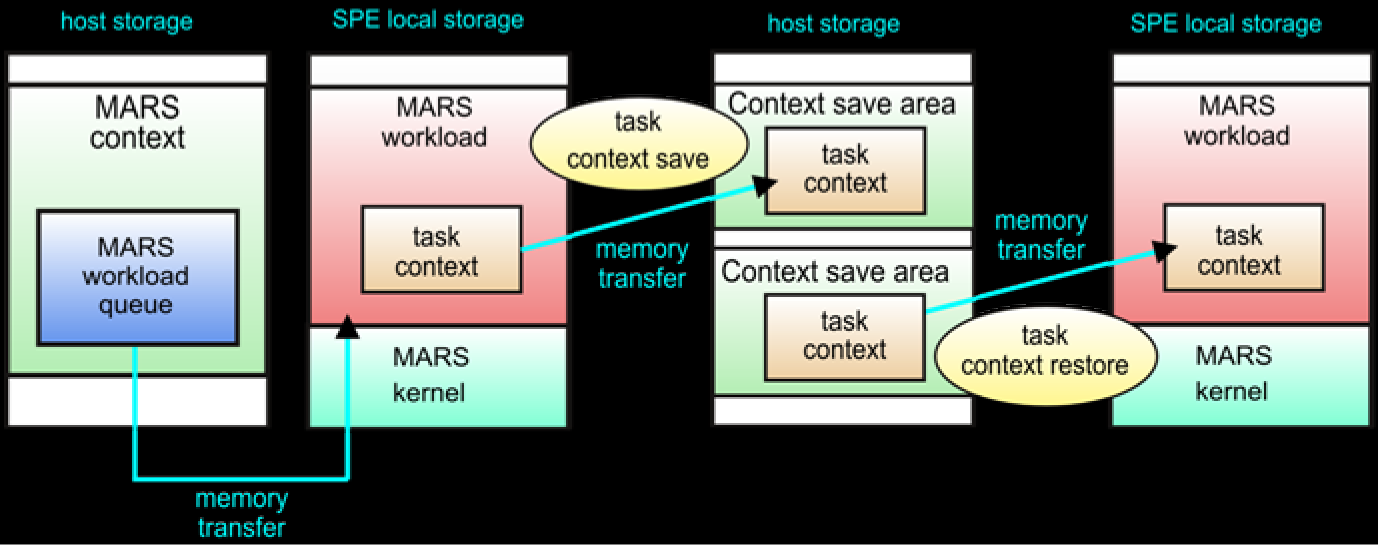
MARS Kernel and Workloads

- The kernel is a small piece of code that stays resident on each SPE
- The kernel is responsible for the scheduling workloads on to SPEs



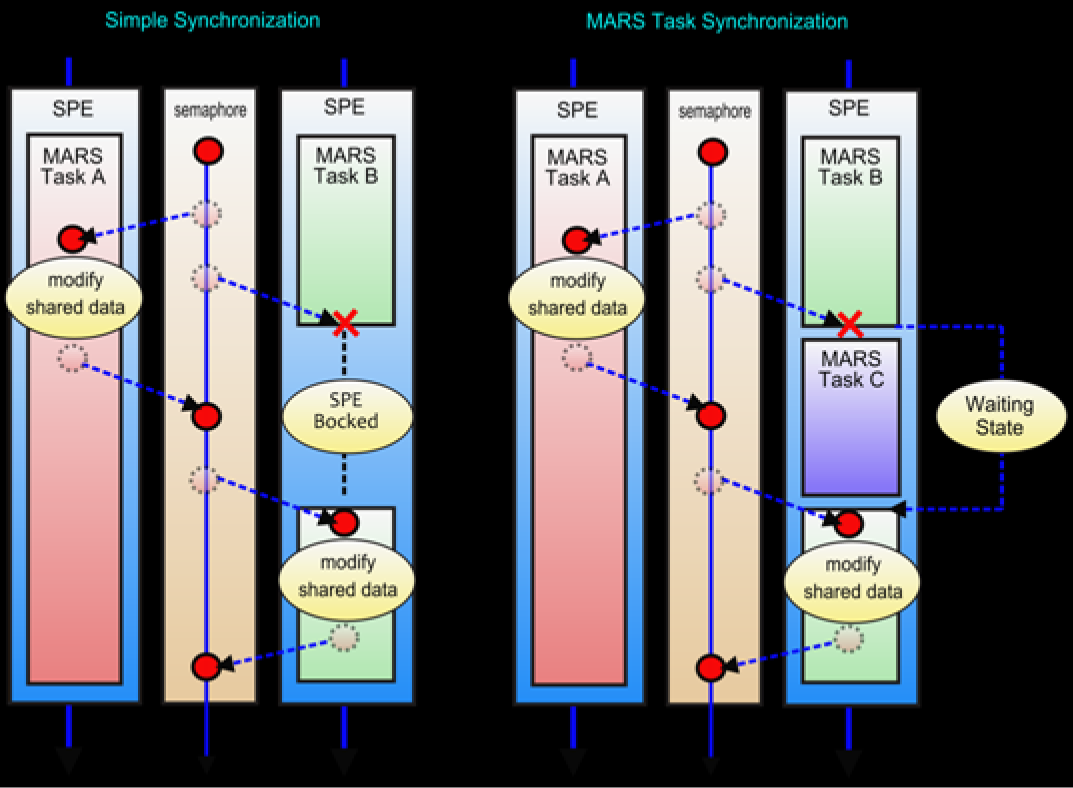
MARS Task and Task Switching

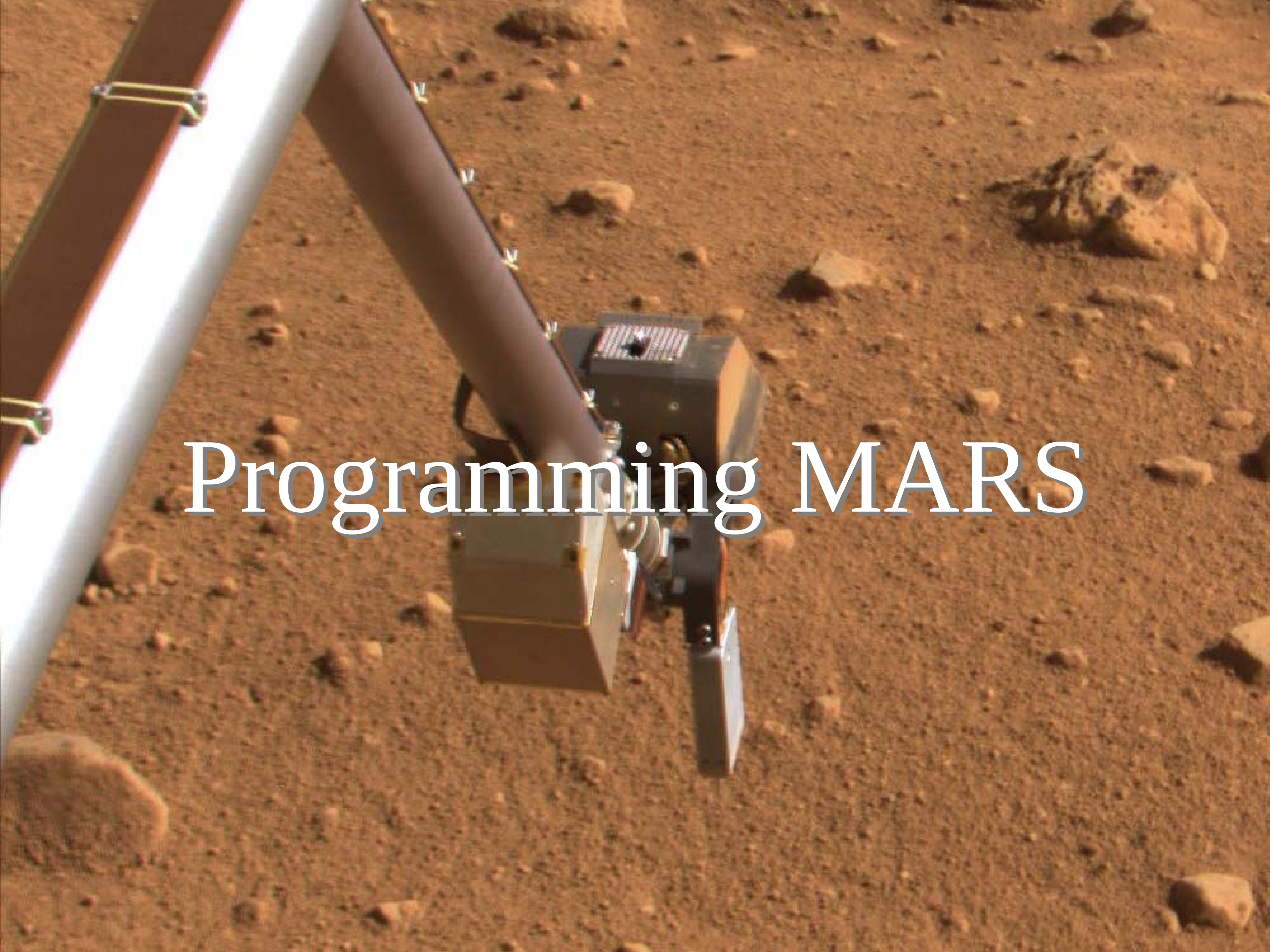
- MARS task is one type of MARS workload
- Task is for large grained program that takes long amounts of time to process
- Context switching is available
- Sync API is provided



MARS Task Sync Objects

- Semaphores, event flags, queues...
- Waiting condition results in a task switch
  - Avoiding wasting time just on waiting



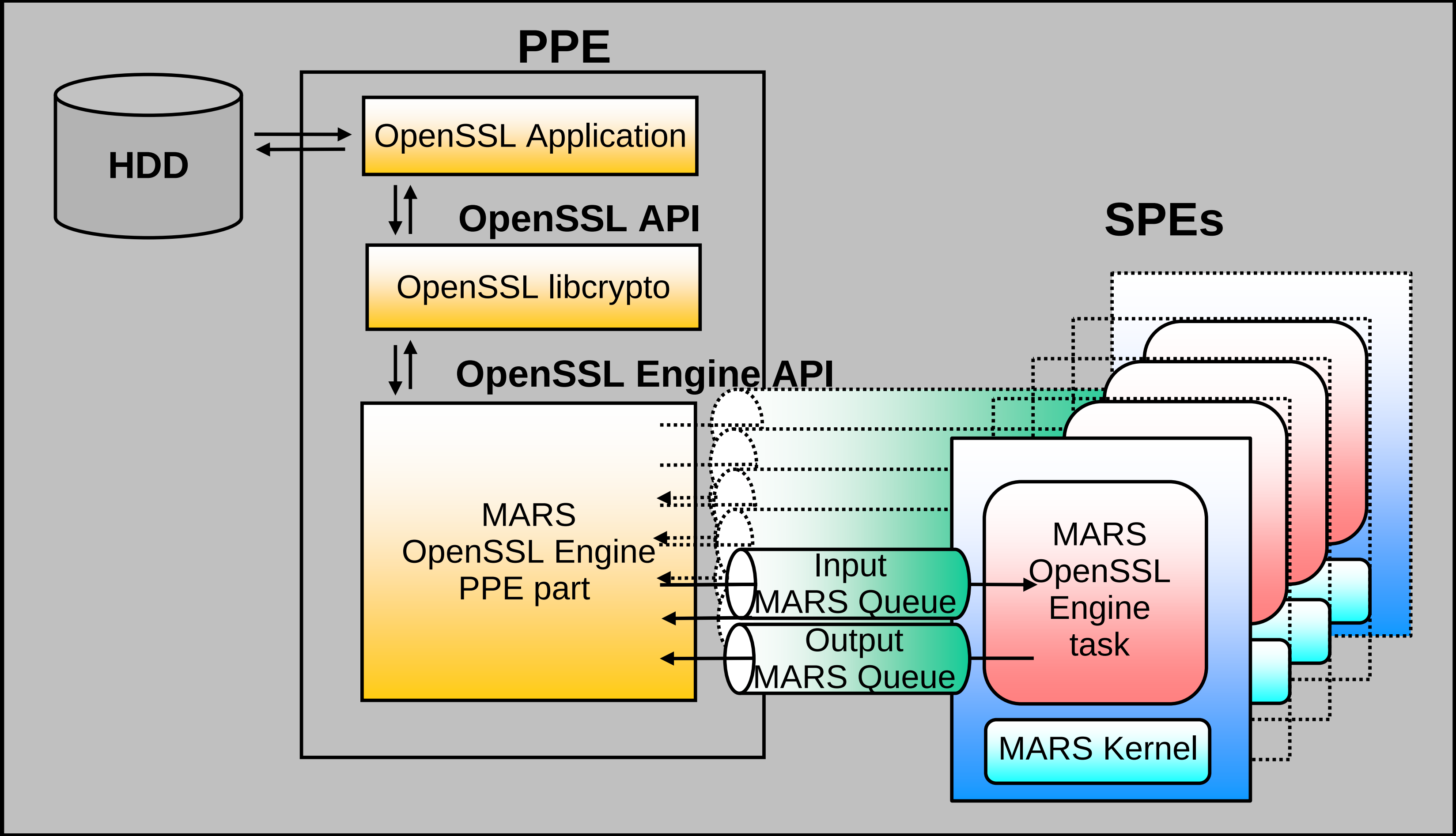


Programming MARS

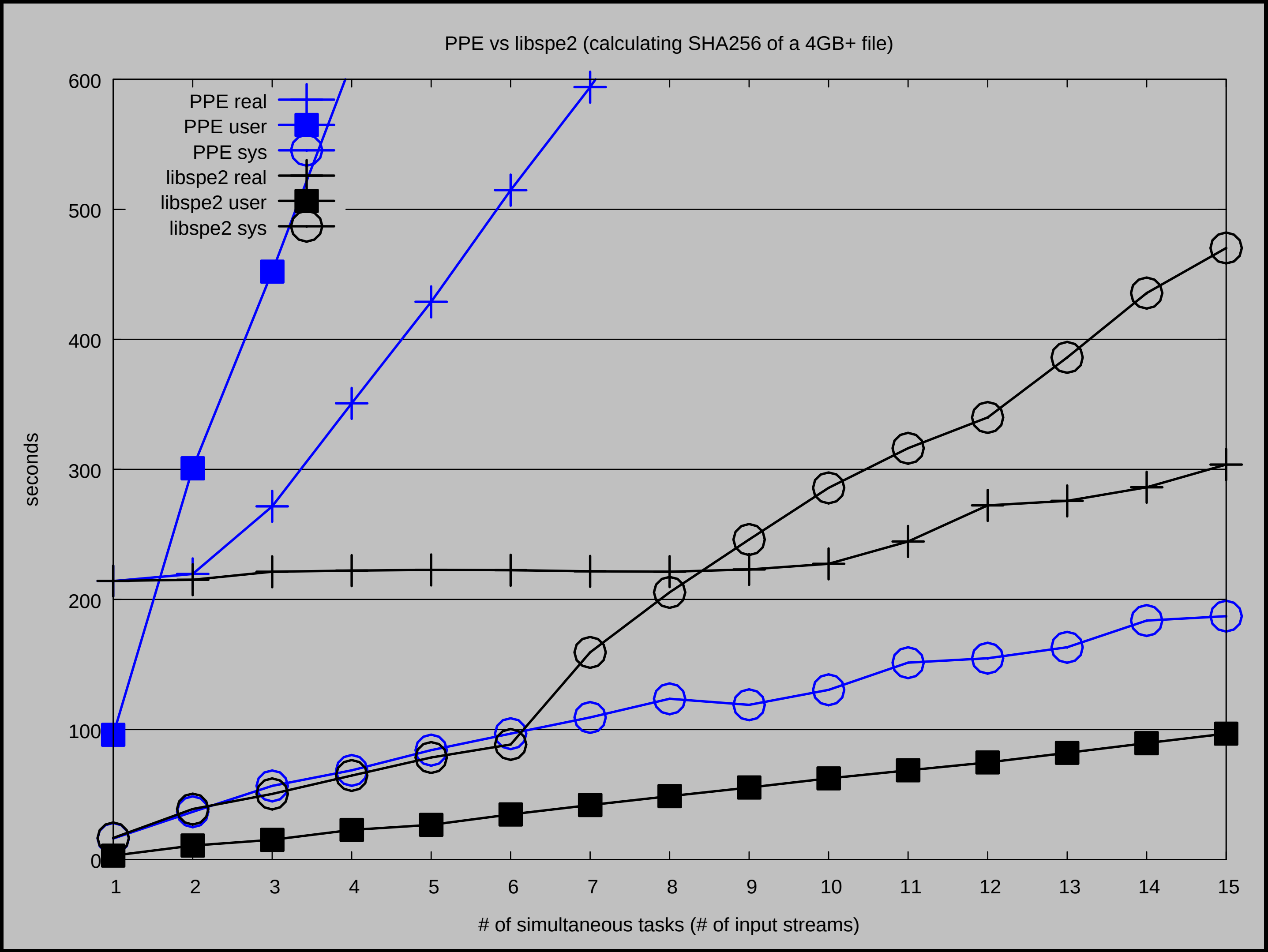
Typical Usage Scenario

1. PPE creates MARS context
2. PPE creates task objects
3. PPE creates synchronization objects
4. PPE starts the initial tasks
5. The existing tasks start additional tasks
6. The tasks do application specific works
7. PPE waits for tasks
8. PPE destroys task objects and sync objects
9. PPE destroys MARS context

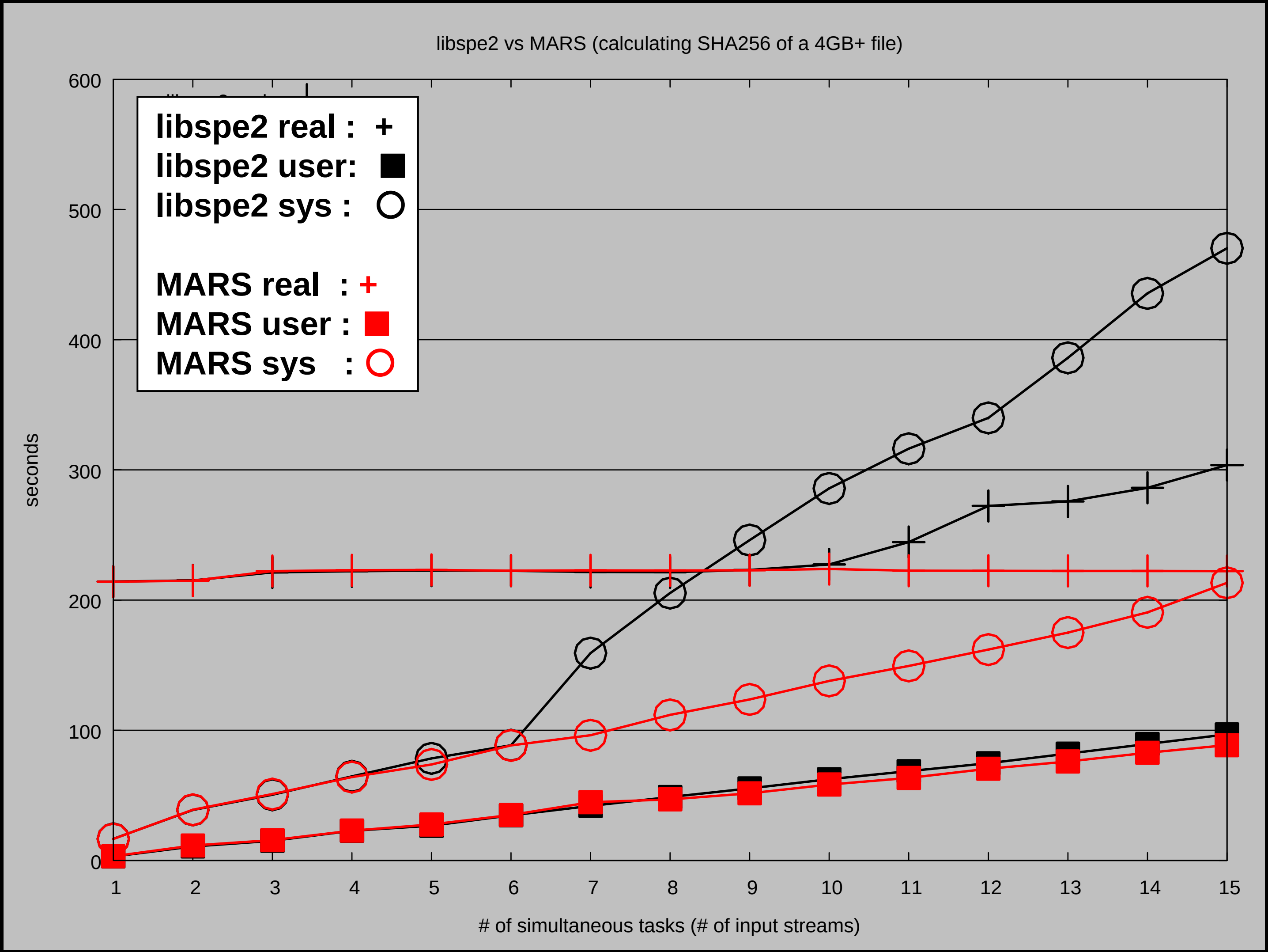
Sample Application: OpenSSL



Benchmarking: OpenSSL PPE vs SPE



Benchmarking: OpenSSL libspe2 vs MARS



How to Approach MARS

Information on MARS

- MARS Releases, Source Code, Samples
  http://ftp.uk.linux.org/pub/linux/Sony-PS3/mars/
- Mailing List Discussions:
  cbe-oss-dev@ozlabs.org
  https://ozlabs.org/mailman/listinfo/cbe-oss-dev
- IRC Discussions:
  #cell at irc.freenode.org
- MARS Development Repositories:
  git://git.infradead.org/ps3/mars-src.git
  http://git.infradead.org/ps3/mars-src.git