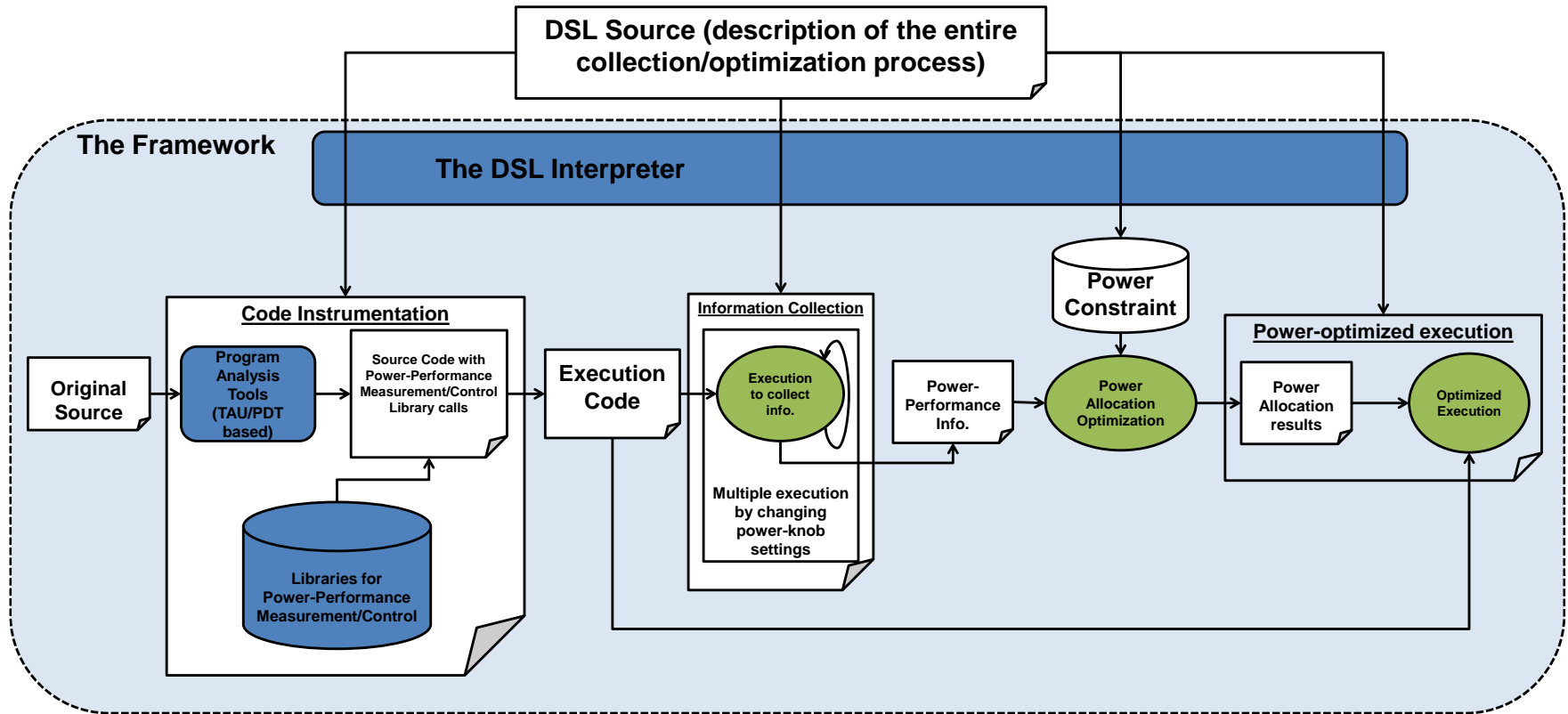


A Simple Introduction and Tutorial to the DSL

A Typical Workflow of the POMPP Framework through the DSL



- In general, the framework is capable of
 - Code instrumentation
 - Collection of performance and power information
 - Specifying power constraints
 - Optimized execution of applications
- The DSL is a front-end to the Framework which helps describing such tasks

Semantics of the DSL

- Statements are the basic components of this DSL
 - A statement is made of a command and some arguments
 - So far, there are 10 commands implemented
 - CREATE, DELETE, ADD, REMOVE, LIST, GET, SET, SWITCH, SUBMIT and INSERT
 - Together with necessary arguments, statements can be interpreted to carry out tasks such as calibration, profiling, optimization and so on
- There are mainly two data structures in this DSL, object and property
 - Objects can be defined with three types, MACHINE, JOB and MODEL
 - An object may have many properties

Objects

- Objects of different types reflect different purpose
 - JOBs are used to embody an application for a specific purpose, such as
 - Execution
 - Calibration
 - Profiling
 - Optimized execution
 - MACHINES are used to specify and store machine specifications
 - MODELS are used to guide optimizations

Properties

- Properties are pre-defined for the DSL
- There are current 23 properties that can be added to created objects
- OBJECT_TYPE is defined for all 3 types of objects and it tells the object type
- JOB_TYPE, OPTIONS, EXEC_PATH, NUM_OF_THREADS, ELAPSED_TIME, ITERATIONS, PVT_PATH, MODULE_POWER, CONTROL_MODE, MODEL_TO_USE and PROFILE_NAME are defined for JOBS
 - Out of these JOB-related properties, JOB_TYPE, OPTIONS, EXEC_PATH are universal for all types of jobs
 - There are 4 types of jobs, GENERAL, CALIBRATION, PROFILING and OPTIMIZATION, there are the valid values for the property JOB_TYPE
 - EXEC_PATH is used to store the absolute path to the executable of a JOB
 - OPTIONS are used to store the options for the executable
 - NUM_OF_THREADS and ELAPSED_TIME are needed by JOBS when JOB_TYPE is GENERAL
 - ITERATIONS and PVT_PATH are needed by JOBS when JOB_TYPE is either CALIBRATION or PROFILING
 - PROFILE_NAME is needed by JOBS when JOB_TYPE is either PROFILING or OPTIMIZATION
 - This means, a profile generated by a profiling run is needed to optimize the execution of the same application
 - MODULE_POWER, CONTROL_MODE and MODEL_TO_USE are needed by JOBS when JOB_TYPE is OPTIMIZATION
 - MODEL_TO_USE stores the name of the MODEL created for this optimization task
 - Details on MODULE_POWER and CONTROL_MODE can be found in Section VI.3 of the framework readme
- POMPP_NPKGS_PER_NODE, POMPP_NCORES_PER_PKG, POMPP_TOTAL_NODES, POMPP_MAX_FREQ, POMPP_MIN_FREQ, POMPP_PKG_TDP, POMPP_DRAM_TDP, POMPP_PKG_MIN, POMPP_DRAM_MIN and POMPP_MODULE_MIN are defined for MACHINES
 - They are used to specify the parameters of the underlying system for the framework, more details can be found in Section II.1 of the framework readme

Format of Statements

- A statement is made of a command and some arguments
 - The command is the action to be taken
 - The arguments are used to embody all other information such as the target of the action and necessary information to take the action
 - Example: CREATE JOB EP
 - An object of type JOB is created with a name of “EP”

CREATE and DELETE Statements

- These two statements are used to create and delete objects
 - Formats
 - CREATE Type Object_Name
 - DELETE Object_Name

ADD and REMOVE Statement

- These two statements are used to add or remove properties in a created object
 - If the following statement is executed
 - `ADD EP ELAPSED_TIME 00:00:10`
 - Then an object named EP will have a property named `ELAPSED_TIME` added
 - Similarly, “`REMOVE EP ELAPSED_TIME`” will remove this property

GET, SET and LIST Statements

- GET and SET are used to retrieve and modify the value of existing properties in created objects
 - Formats are similar to ADD and REMOVE
- LIST is used to show all/some collection of created objects
 - LIST ALL shows all created objects
 - LIST JOB will show all objects of the type JOB

INSERT Statement

- INSERT is used for manual instrumentation
 - Format:
 - INSERT Absolute_Path Line_number Source_Language POMPP_Function_Call Section_Name
- Instrumentation functions are:
 - POMPP_Init(argc, argv), POMPP_Finalize(), POMPP_Start_section("name") and POMPP_Stop_section("name") for C
 - call POMPP_Init(), call POMPP_Finalize(), call POMPP_Start_section("name") and call POMPP_Stop_section("name") for Fortran

SWITCH Statement

- SWITCH statement is used to switch to a new set of machine settings
 - MACHINE is an object that can be created
 - When SWITCH MACHINE_NAME is executed, all properties added to this MACHINE will be translated into the machine.conf file of the POMPP Framework
 - So we can create a lot of machines with different settings and switch to any of them when the hardware environment is actually changed

SUBMIT Statement

- SUBMIT statement is used to generate the job submission script
 - As previously mentioned, there are four types of JOBS, each of them will be translated to a submission script which is also based on a MPI wrapper script provided by the POMPP Framework
 - JOB_TYPE of GENERAL will generate a typical job submission script
 - JOB_TYPE of PROFILING will generate a job submission script based on the mpirun-prof.sh for profiling purpose
 - JOB_TYPE of OPTIMIZATION will generate a job submission script based on the mpirun-opt.sh for optimizing purpose under a specified power cap
 - JOB_TYPE of CALIBRATION will generate a job submission script based on the mpirun-ubmt.sh for hardware calibration
 - The workload in such a job is used as a microbench

A Few DSL Sample Sources

Machine settings

CREATE MACHINE M1  A machine object is created

ADD M1 POMPP_NPKGS_PER_NODE 2

ADD M1 POMPP_NCORES_PER_PKG 12

ADD M1 POMPP_TOTAL_NODES 965

ADD M1 POMPP_MAX_FREQ 16

ADD M1 POMPP_MIN_FREQ 12


ADD M1 POMPP_PKG_TDP 130.0


ADD M1 POMPP_DRAM_TDP 62.0

ADD M1 POMPP_PKG_MIN 64.0

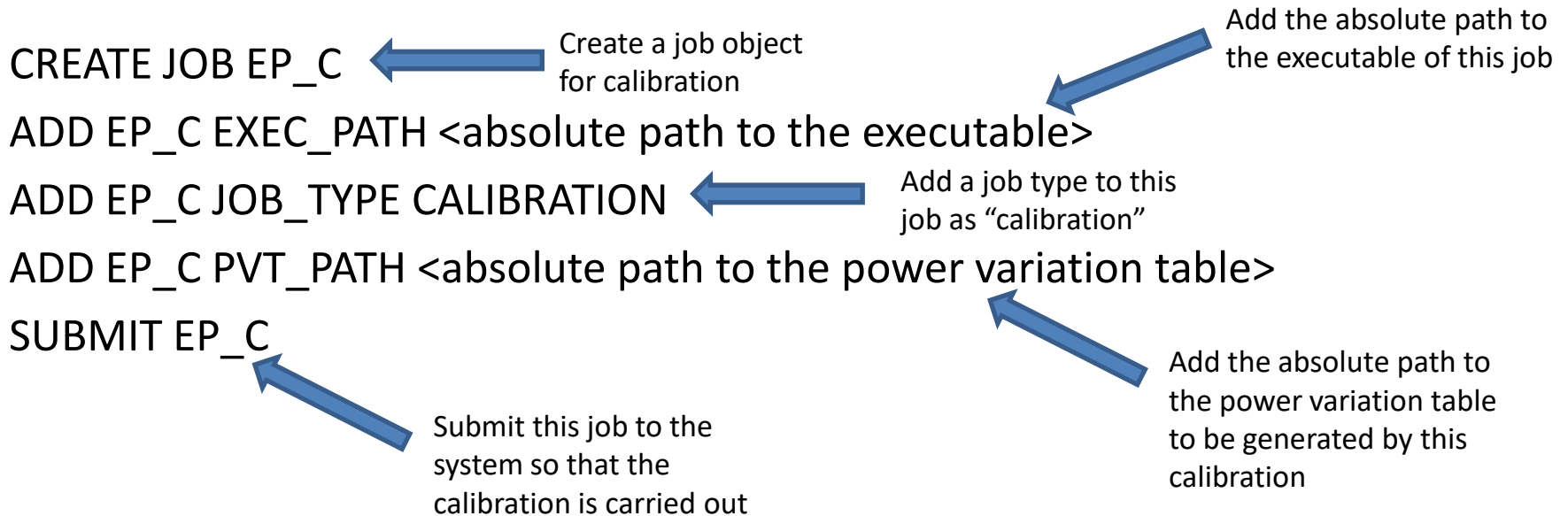
ADD M1 POMPP_DRAM_MIN 30.0

ADD M1 POMPP_MODULE_MIN 46.0

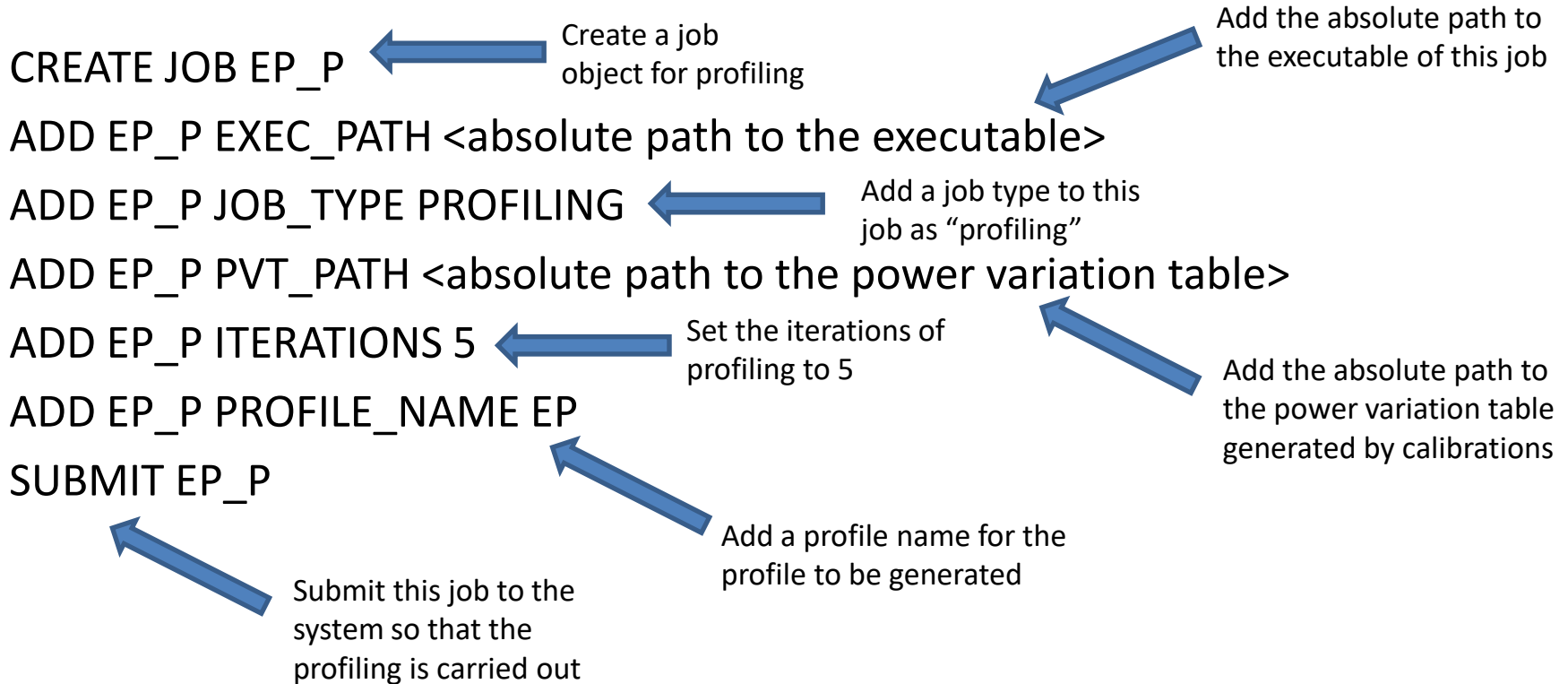
SWITCH M1  Switch to this machine object to apply these settings to the current system

 Settings are added as properties to the object

Hardware calibration



Profiling an Application



Run an Application with a Provided Optimization Model and a Power Cap

