XPDL

eXtensible Platform Description Language

EXCESS

Execution Models for Energy-Efficient Computing Systems

Christoph Kessler — Linköping University, Linköping, Sweden

Lu Li

Aras Atalar — Chalmers University, Gothenburg, Sweden

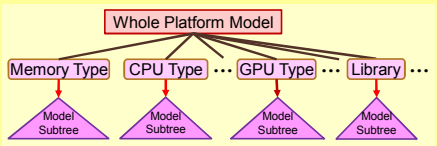
Alin Dobre — Movidius, Dublin, Ireland

Goals: Retargetability of Toolchains and Adaptivity for Holistic Energy Optimization

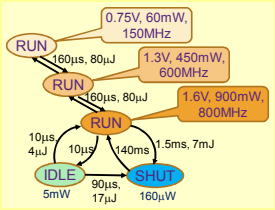
- Automation of platform-dependent optimizations and deployment
- Prerequisite: A formal target platform modeling language for modeling optimization-relevant platform features

Platform = Hardware + System Software

XPDL Features

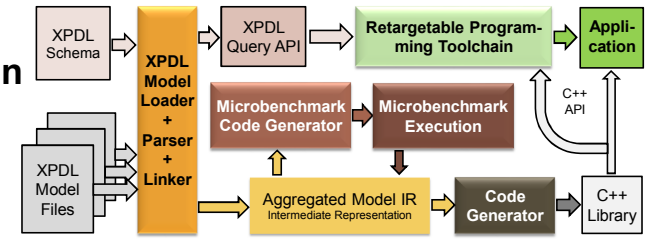


- **Modular and extensible**
- **Syntax: XML**
  - Mature tool support
  - Syntactic flavor does not restrict its applicability
- **Control relation decoupled from hardware structure**
- **Inheritance**
- **Support for Configurability**
  - Parameters and Constraints
- **System software modeling**
- **Power modeling**
  - Instruction Set with Dynamic Power Costs
  - Power and Frequency Domains
  - Power State Machines with Parameters



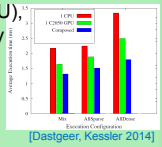
- **Generating Deployment-time Microbenchmarking Code**
  - For deriving statically-unknown model parameter values
  - For generating performance and energy models
- **Query API for static and run-time model introspection**
  - Available to application (run time) and tool chain (libraries, compilers, runtime systems, autotuners etc.)
  - Browsing the model tree, parameter queries, model analysis

XPDL Toolchain



Use Case: Platform-Constraints on Variant Selectability

- **Multi-Variant Components**
  - e.g. *SkePU* skeletons or *PEPPHER* components
  - Platform-specific implementation variants (Seq., OpenMP, CUDA, …)
- **Conditional composition** [Dastgeer, Kessler 2014]
  - Platform constraints on selectability expressed based on XPDL's predecessor PDL (*PEPPHER* Platform Description Language) [Sandrieser *et al.* 2012]
  - Case study: *Sparse Matrix-Vector Multiply* variants (CPU, GPU), selectability constrained by platform-specific library availability
- PDL Problems: Flexibility, scalability, modularity issues
  - e.g., main structuring by control relation (a *software* aspect), rather than hardware organization



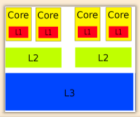
[Dastgeer, Kessler 2014]

References

- U. Dastgeer, C. Kessler: **Conditional Component Composition for GPU-based Systems**. Proc. MULTIPROG'14 workshop at HiPEAC'14, Vienna, Jan. 2014.
- M. Sandrieser, S. Benkner, S. Pllana: **Using explicit platform descriptions to support programming of heterogeneous many-core systems**. *Parallel Computing* 38, 2012

Pre-defined Typed Modeling Elements and Attributes

```
<cpu name="Intel_Xeon_E5_2630L">
  <group prefix="core_group" quantity= "2" >
    <group prefix="core" quantity= "2" >
      <!-- Embedded definition -->
      < core frequency="2" frequency_unit="GHz" >
      <cache name= "L1" size="32" unit="KiB" />
      </core>
    </group>
    <cache name= "L2" size="256" unit="KiB" />
  </group>
  <cache name= "L3" size="15" unit="MiB" />
  <power_model type="power_model_E5_2630L" />
</cpu>
```



- Can define new types by extending the meta-model
- 1 file per element
- Globally browsable model repository

Type Referencing and Inheritance

```
<device name="Nvidia_K20c" extends="Nvidia_Kepler">
 <compute_capability="3.5" />
 <param name="num_SM" value="13" />
 <param name="coresperSM" value="192" />
 <param name="cfrq" frequency="706" ...unit="MHz"/>
 <param name="gmsz" size="5" unit="GB" />
 ...
</device>
```

Metamodel (Element Types)

```
<system id="liu_gpu_server">
  <socket>
    <cpu id="gpu_host" type="Intel_Xeon_E5_2630L"/>
  </socket>
  <device id="gpu1" type="Nvidia_K20c" />
  <interconnects>
    <interconnect id="connection1" type="pcie3"
                  head="gpu_host" tail="gpu1" />
  </interconnects>
</system>
```

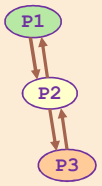
Concrete Platform Model (Instance)

```
<device name="Nvidia_Kepler" extends="Nvidia_GPU"
        role="worker">
 <compute_capability="3.0" />
 <const name="shmtotalsize"... size="64" unit="KB"/>
 <param name="L1size" configurable="true"
        type="msize" range="16, 32, 64" unit="KB"/>
 <param name="shmsize" configurable="true"
        type="msize" range="16, 32, 64" unit="KB"/>
 <param name="num_SM" type="integer"/>
 <param name="coresperSM" type="integer"/>
 <param name="cfrq" type="frequency" />
 <param name="gmsz" type="msize" />
 <constraints>
   <constraint expr=
       "L1size + shmsize == shmtotalsize" />
 </constraints>
 <group name="SMs" quantity="num_SM">
   <group name="SM">
     <group quantity="coresperSM">
       <core type="..." frequency="cfrq" />
     </group>
     <cache name="L1" size="L1size" />
     <memory name="shm" size="shmsize" />
   </group>
 </group>
 <memory type="global" size="gmsz" />
 ...
 <programming_model type="cuda6.0,...,opencl"/>
</device>
```

Constraint

Power and Microbenchmark Modeling

```
<power_state_machine name="power_state_machine1"
                     power_domain="xyCPU_core_pd">
  <power_states>
    <power_state name="P1" frequency="1.2"
                 frequency_unit="GHz"
                 power="20" power_unit="W" />
    <power_state name="P2" frequency="1.6" ... />
    <power_state name="P3" frequency="2.0" ... />
  </power_states>
  <transitions>
   <transition head="P2" tail="P1"
               time="1" time_unit="us"
               energy="2" energy_unit="nJ"/>
   <transition head="P3" tail="P2" ... />
   <transition head="P1" ... />
  </transitions>
</power_state_machine>
```




```
<microbenchmarks id="mb_x86_base_1"
                 instruction_set="x86_base_isa"
                 path="/usr/local/micr/src"
                 command="mbscript.sh">
  <microbenchmark id="fa1" type="fadd" file="fadd.c"
                  ...gs="-O0" lflags="..." />
  <microbenchmark id="mo1" type="mov" file="mov.c"
                  ...gs="-O0" lflags="..." />
</microbenchmarks>
```

```
<instructions name="x86_base_isa"
              mb="mb_x86_base_1" >
 <inst name="fmul"
       energy="?" energy_unit="pJ" mb="fm1"/>
 <inst name="fadd"
       energy="??" energy_unit="pJ" mb="fa1"/>
 ...
```

References

- **XPDL** web page: http://www.ida.liu.se/labs/pelab/xpdl
- C. Kessler, L. Li, A. Atalar, A. Dobre: **XPDL: An Extensible Platform Description Language Supporting Energy Modeling and Optimization.** Proc. ICPP-2015 workshop on Embedded Multicore Systems, Sep. 2015, IEEE. DOI: 10.1109/ICPPW.2015.17.
- C. Kessler *et al.*: **EXCESS project deliverables D1.2** (2014), **D1.4** (2015)
- **Related work**: PEPPHER PDL, REPARA HPP-DL, INRIA hw-loc, ALMA-ADL …