Models and Tools

Cross-Layer Theory and Models

Alex Yakovlev, Newcastle University
PRiME Industrial Event
London, 30/June/2015

PRiME
www.prime-project.org

Models and Tools
Theme 1: (Big Ideas)

- Theme 1 is developing *theory, models and algorithms*

that underpin the *interplay* between energy and reliability across system layers in a scalable many-core system

- Key challenges:
  - Complex nature of interactions between power, performance and reliability across system layers
  - New (unconventional) definitions and notions of QoS and correct behaviour
  - The effect of many-core scalability on the interplay
  - New paradigms: power proportionality, multiple operation modes, graceful degradation

Taxonomy of Tools for PRiME

- **Design-time tools**
  - to characterise and optimize our platforms for a range of platform parameters, tasks and applications
  - to develop RTM hardware and software support, i.e. tools that are run offline to produce code or hardware that runs online.
  - Tools to debug, test and evaluate RTM strategies
- **Run-time tools**
  - to implement a control strategy
  - to produce code or hardware that runs online.
- Some examples next …

Offline: Close to Implementation

- Characterization tools for modelling and simulation of platforms (especially scalable for many-core), perhaps with **high fidelity**
- This covers the work at Southampton; initially based on extending the GEM5 framework.
- Later it was extracting high-precision models based platform profiling and running regressions (based on benchmarks)
  - *Success criteria: Do these models adequately predict the run-time behaviour of real systems? Are they able to capture the explicit energy / reliability tradeoff?*

Offline: High Level

- Analytical modelling and characterisation tools (deterministic and stochastic) to enable extraction of **PER models** in the large, perhaps with **low fidelity**.
- This covers the work developed by Newcastle to model the performance of manycore systems.
  - *Success criteria: Do these high level models track the low-level models with sufficient fidelity for reasonable design decisions to be made*?

High Level Description of RTM

- This includes developments of Domain Specific Languages (DSL) to capture the main parts of multi-core system management, initiated with a DSL toolkit developed at Newcastle.
- The development process envisaged is iterative with an in-the-loop testing and evaluation and generation of Event-B models of commands and events for DSL-Kit modules and scripts.
  - *Success Criteria: Can we effectively combine the DSL with the Event-B to produce descriptions for a verified RTM of practical size and complexity?*

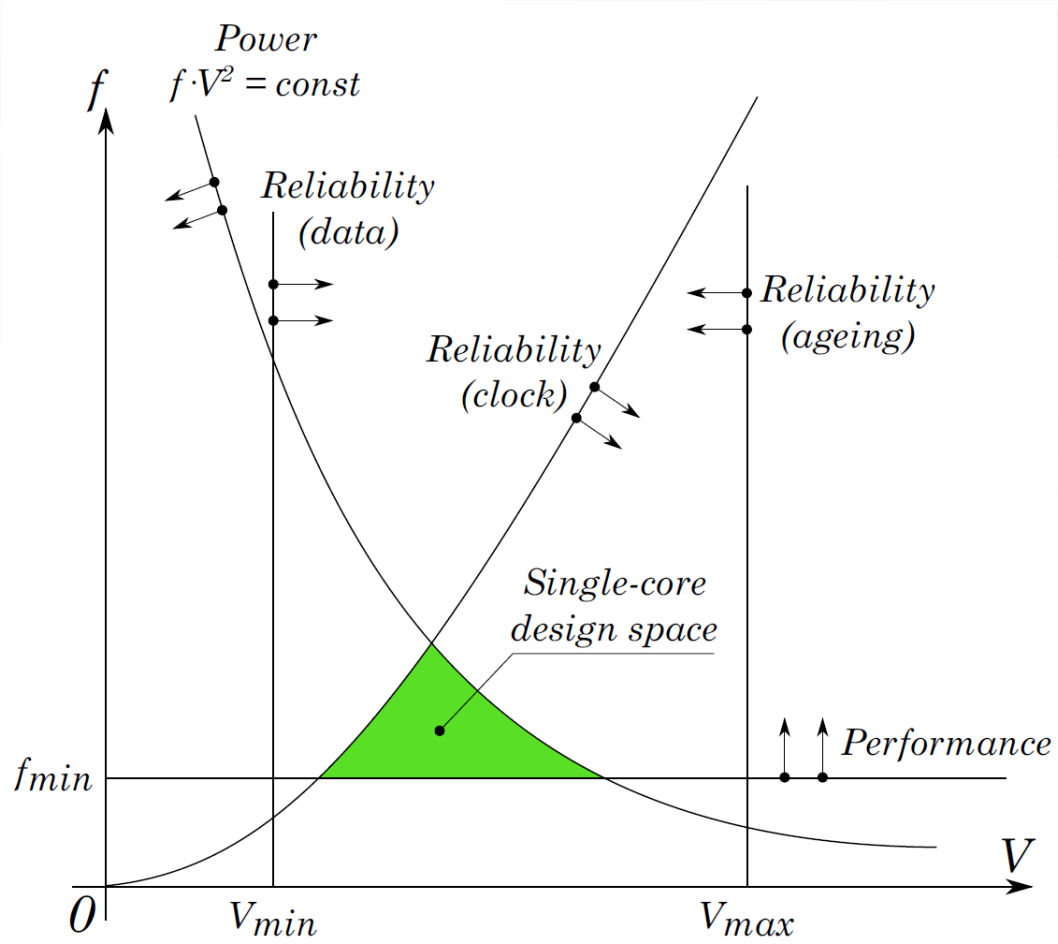
Theme 1 Focus

- **Performance-Energy-Reliability (PER) models**: analytical models using experimental data and incorporating them into scalable models
- **Resource-oriented models**: Significance-driven fidelity models for scalability and reduction of modelling effort, ultimately to improve RTM and productivity of RTM development
- **Domain-specific language (DSL) modelling** for simulation and rapid prototyping of many-core platforms in application context
- Highlights next…

Performance-Energy-Reliability (PER) models

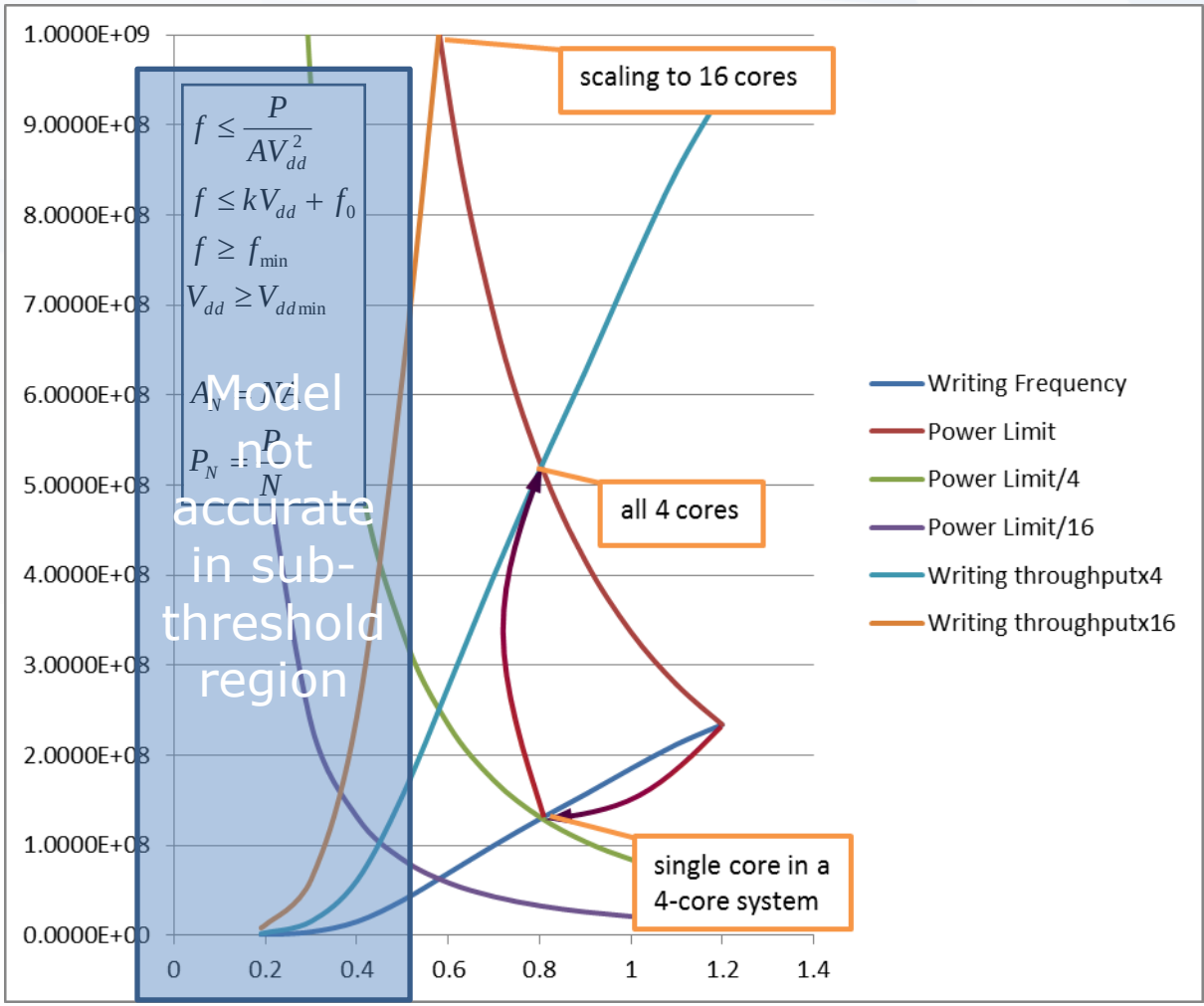
PER Primer

- The power and clock boundaries require most work



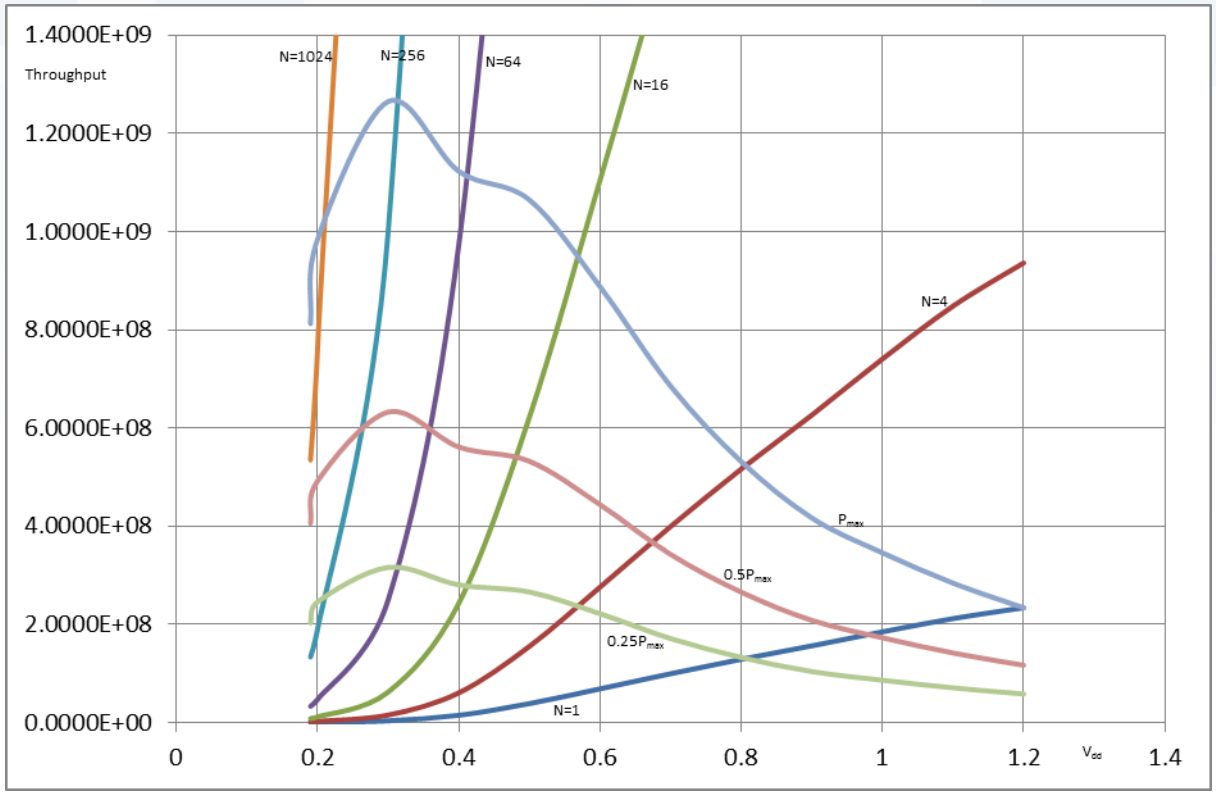
Studying ERI scalability

- Power, computation throughput (effective freq.) and reliability (real data – ASRAM)
- The throughput is however not actual performance due to the imperfections such as synchronization and shared memory locks.



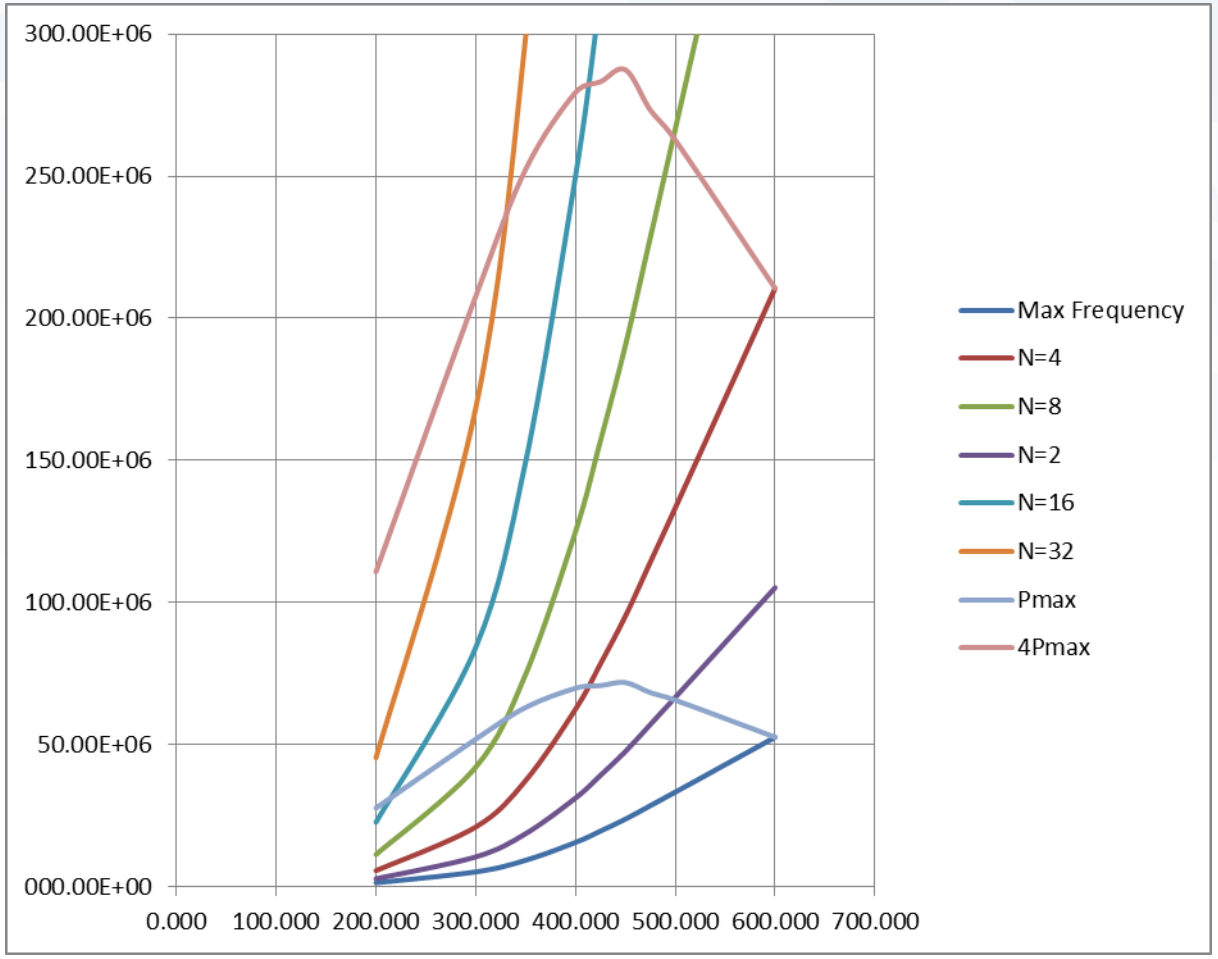
Studying ERI scalability

- Power, computation throughput (effective freq.) and reliability (real data – ASRAM across entire Vdd range)



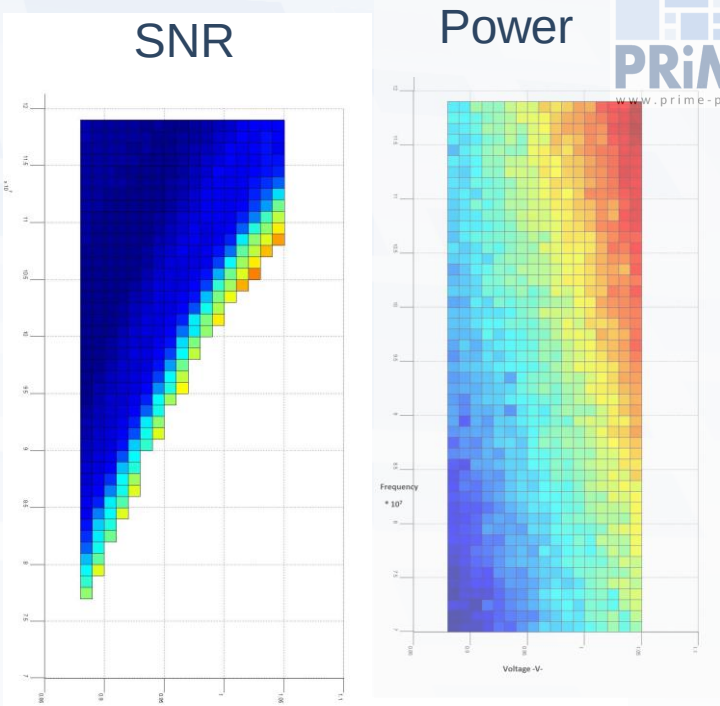
Studying ERI scalability

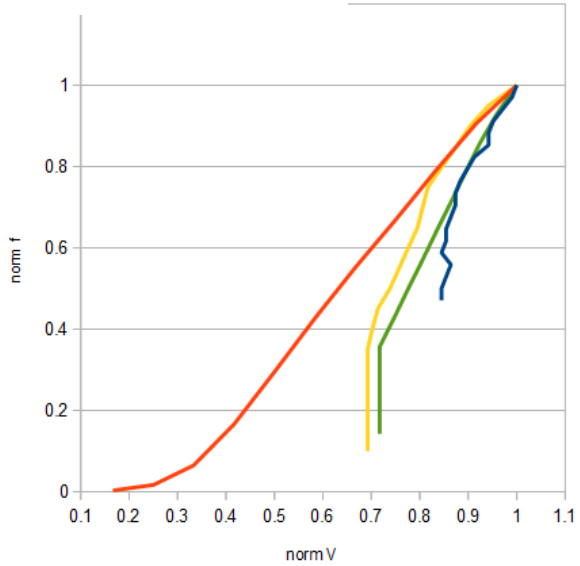
- Power, computation throughput (effective freq.) and reliability (real data – M0 in trans- and sub-threshold regions)



PER modelling

- Building analytical models using actual experimental data:
  - Xeon/Core i for Southampton,
  - FPGA for Imperial,
  - Odroid for Newcastle
- Incorporating PER models into high level scalable models
- The models are used in RTM demos





Cores with flatter (more proportional?!) F(V) curves are better for scaling

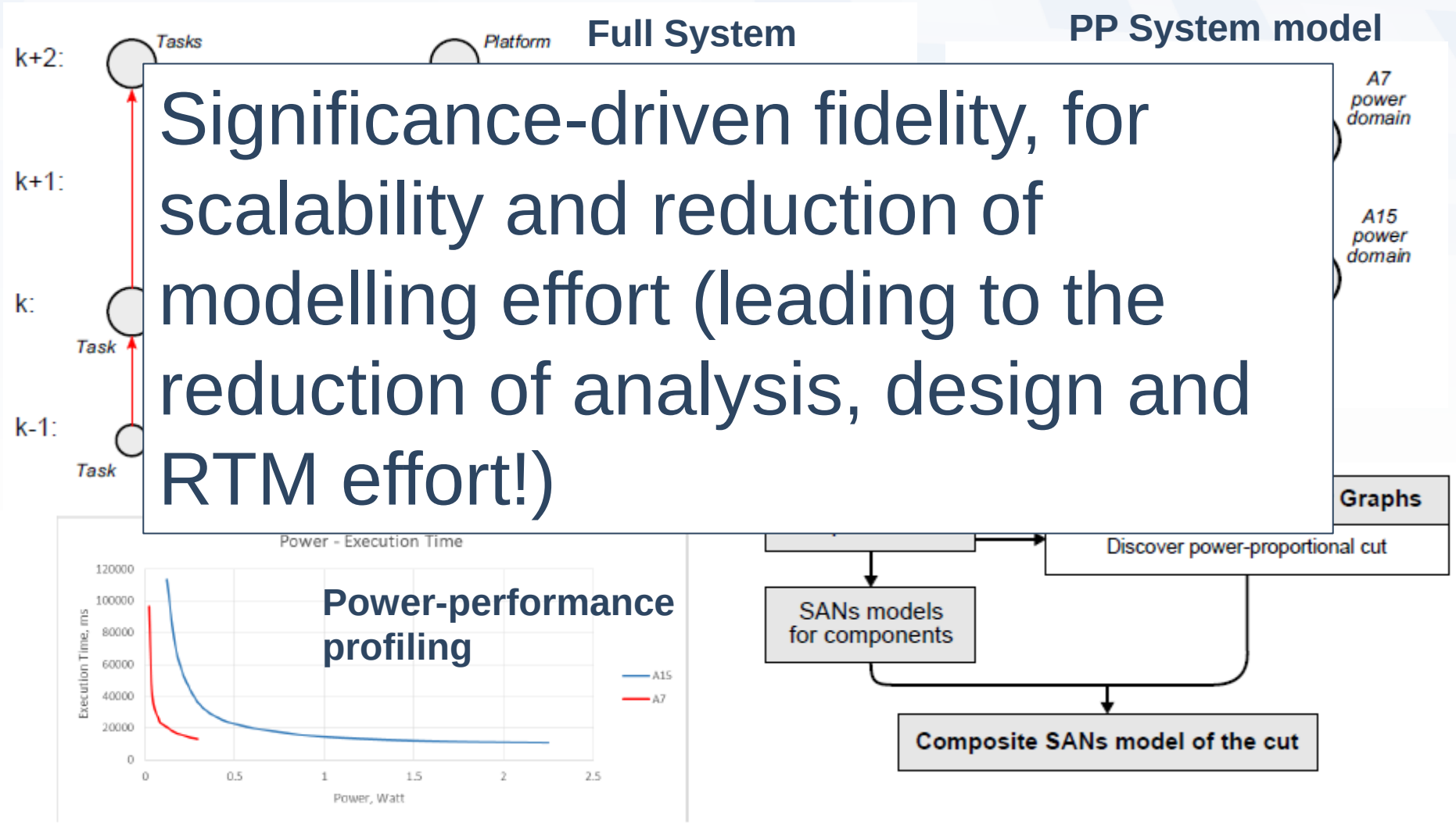
PER Modelling tool (demo)

PRiME
www.prime-project.org

Resource-oriented models

Resource modelling

Significance-driven fidelity, for scalability and reduction of modelling effort (leading to the reduction of analysis, design and RTM effort!)



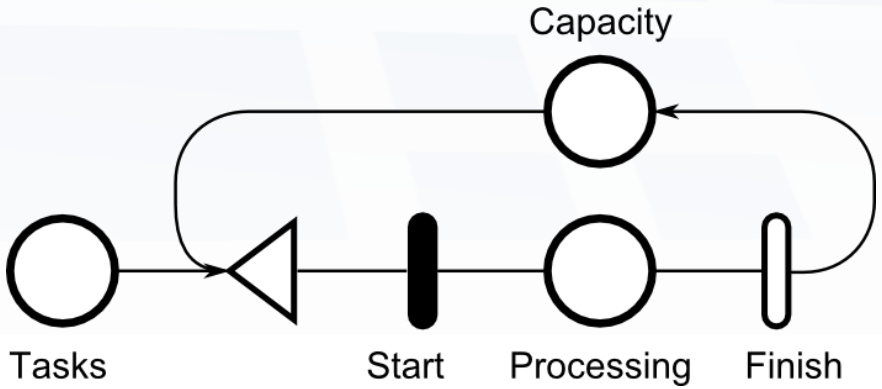
Parametric modeling

Parametric modeling is focusing on quantitative estimation of a set of (physical) parameters.

This can be done using SANs (Stochastic Activity Networks).

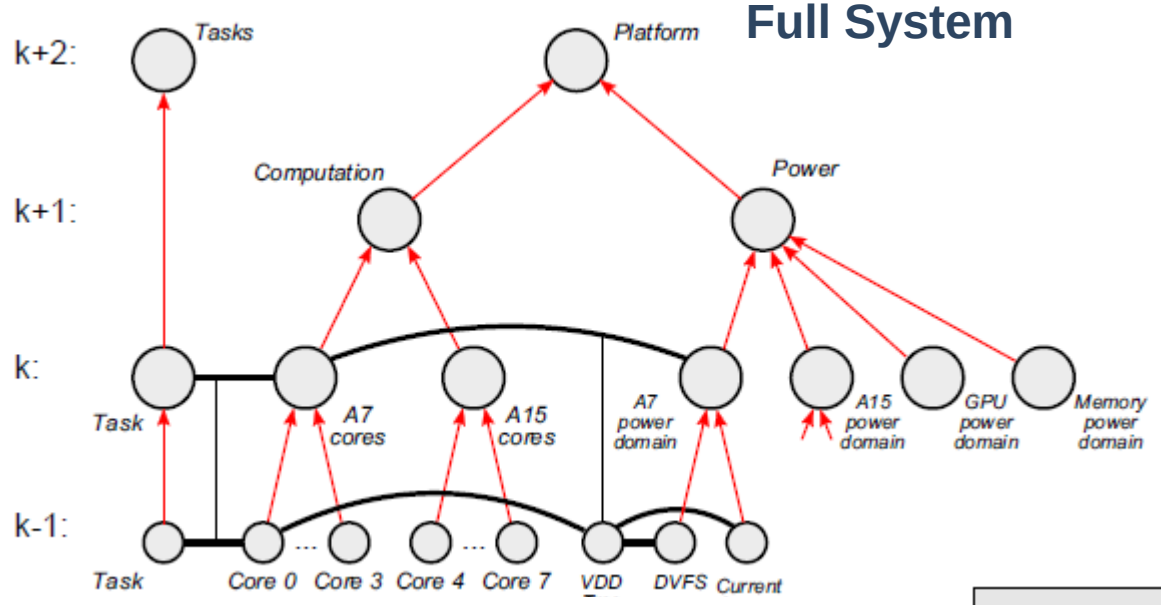
We use **Möbius tool** to edit and analyze SANs models.

https://www.mobius.illinois.edu

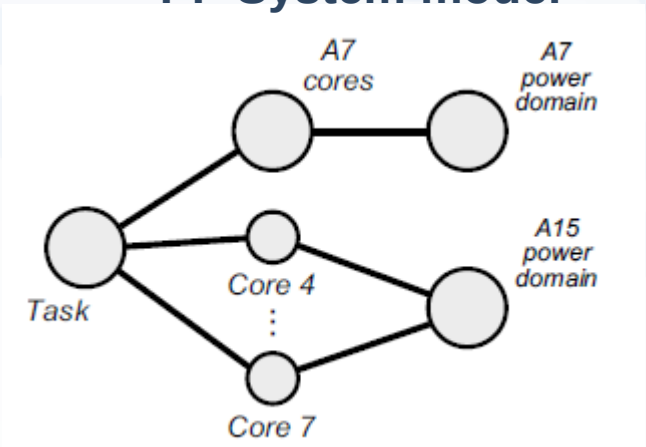


Power proportional modelling fidelity (using Odroid)

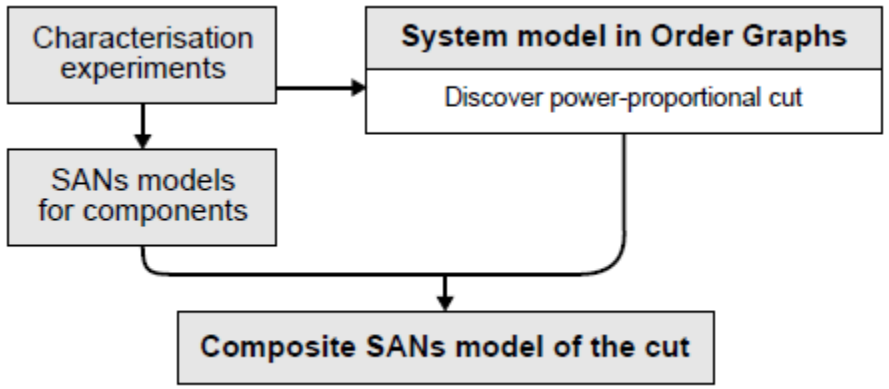
Full System



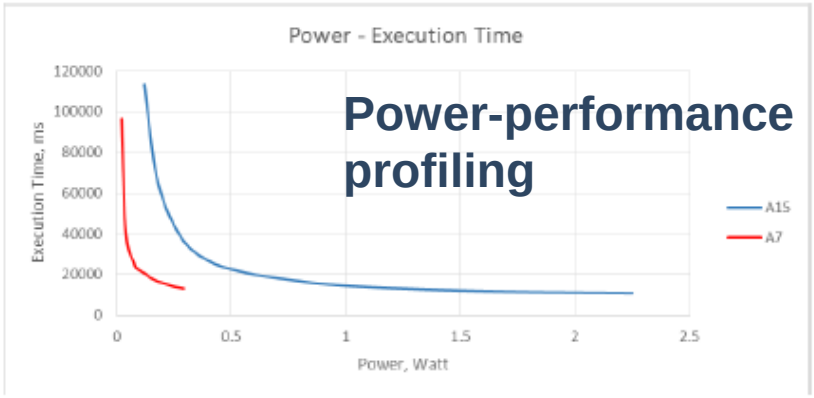
PP System model



Modelling Flow



Power-performance profiling



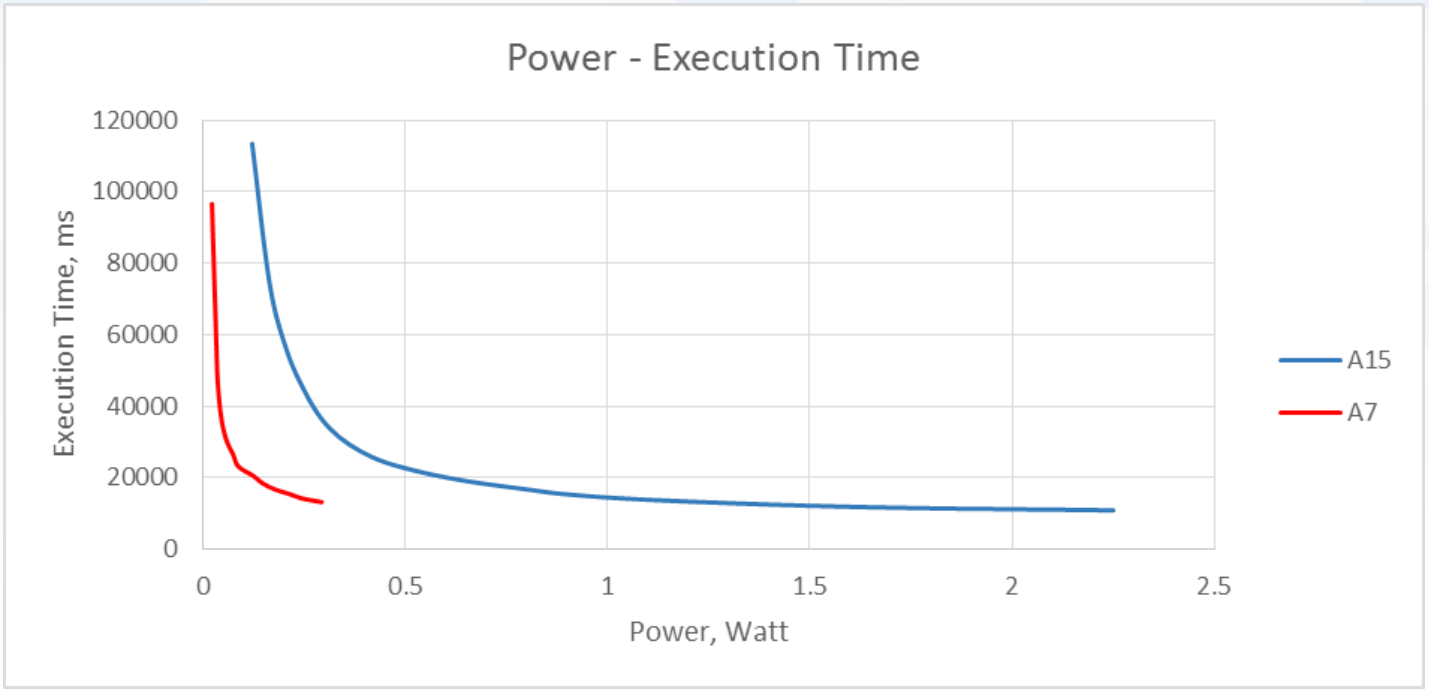
ODROID XU3

- **Heterogeneous** platform based on ARM big.LITTLE:

  4 high performance cores (**A15**).

  4 low power cores (**A7**).
- Same SoC as in Samsung S4.
- Power sensors for each power domain.

ODROID XU3



- A15 consumes 4-10 times power per computation compared to A7.
- Measuring resolution is 0.02mW for A15 vs 0.003mW for A7.

Does it make sense to model A7 to the same degree as A15?

Order Graphs

- Resource-driven modeling.
- Modeling hierarchies.
- Cross-layer modeling.

Resources and dependencies

*Any system is represented as a **Resource graph:***

- nodes are resources,
- edges are dependencies between them.

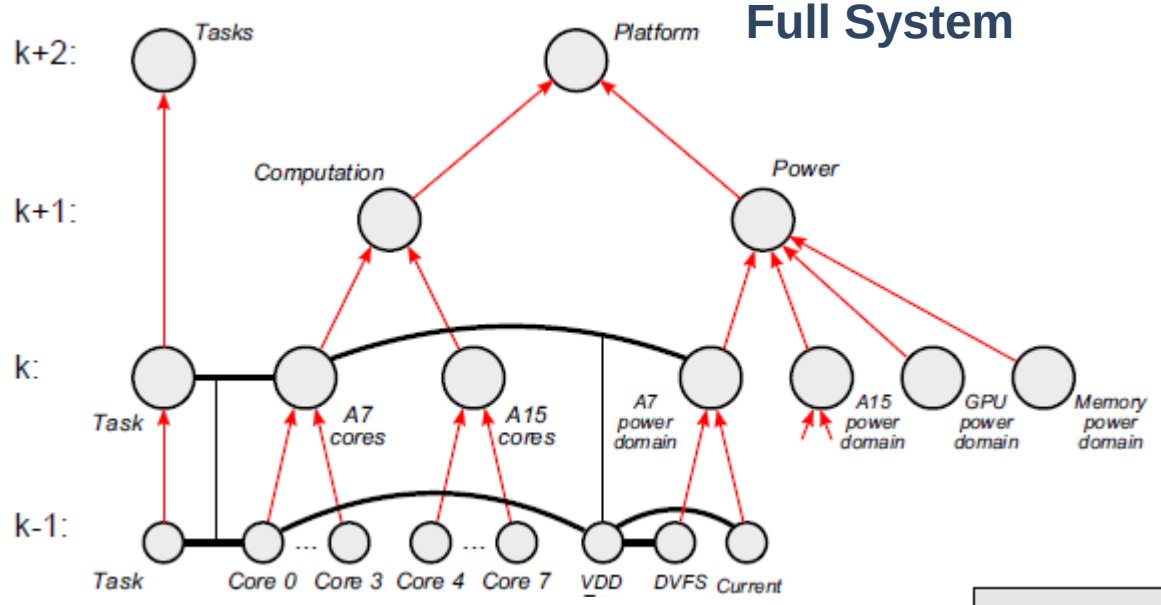
Types of resources: **Hardware Logic, Memory, Energy, Task, Thread,** etc...

Anything can be a resource.

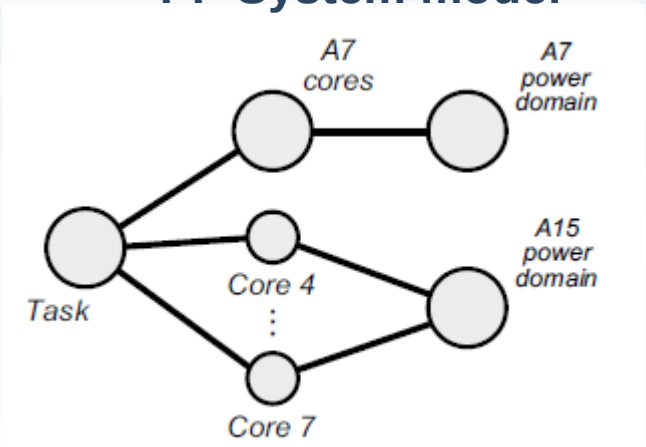
Types of dependencies: **Energy dependency, Data dependency**, and other...

Power proportional modelling fidelity (using Odroid)

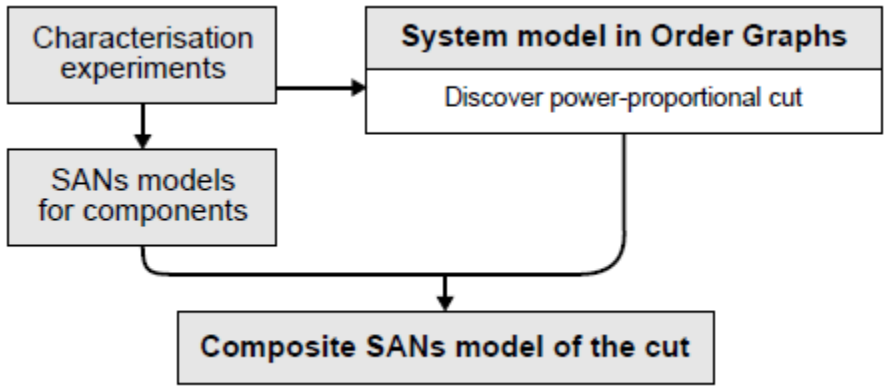
Full System



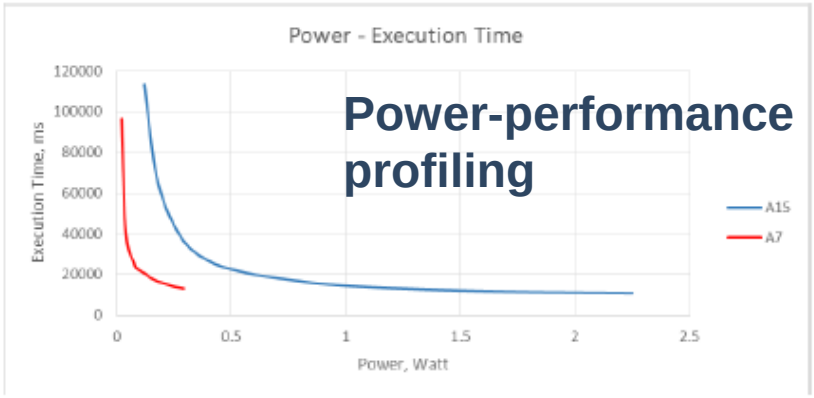
PP System model



Modelling Flow



Power-performance profiling



Structurally we do not model the system to the same level of details.

Parametrically we do model the system to the same level of precision.

Fidelity Balancing, Costs

For each component of the system in every level of abstraction we have a power model, where:

- P is the **estimated power** of the component.
- E is the model **error** (100% - precision) for this component.
- C is complexity, cost of modelling this component.

The goal is to find such a *cross-layer cut* (flat model) that:

- total error in the composed power model and the computation cost of this model satisfy our requirements;
- variance of (P/(E*C)) is minimal across the components in the cut.

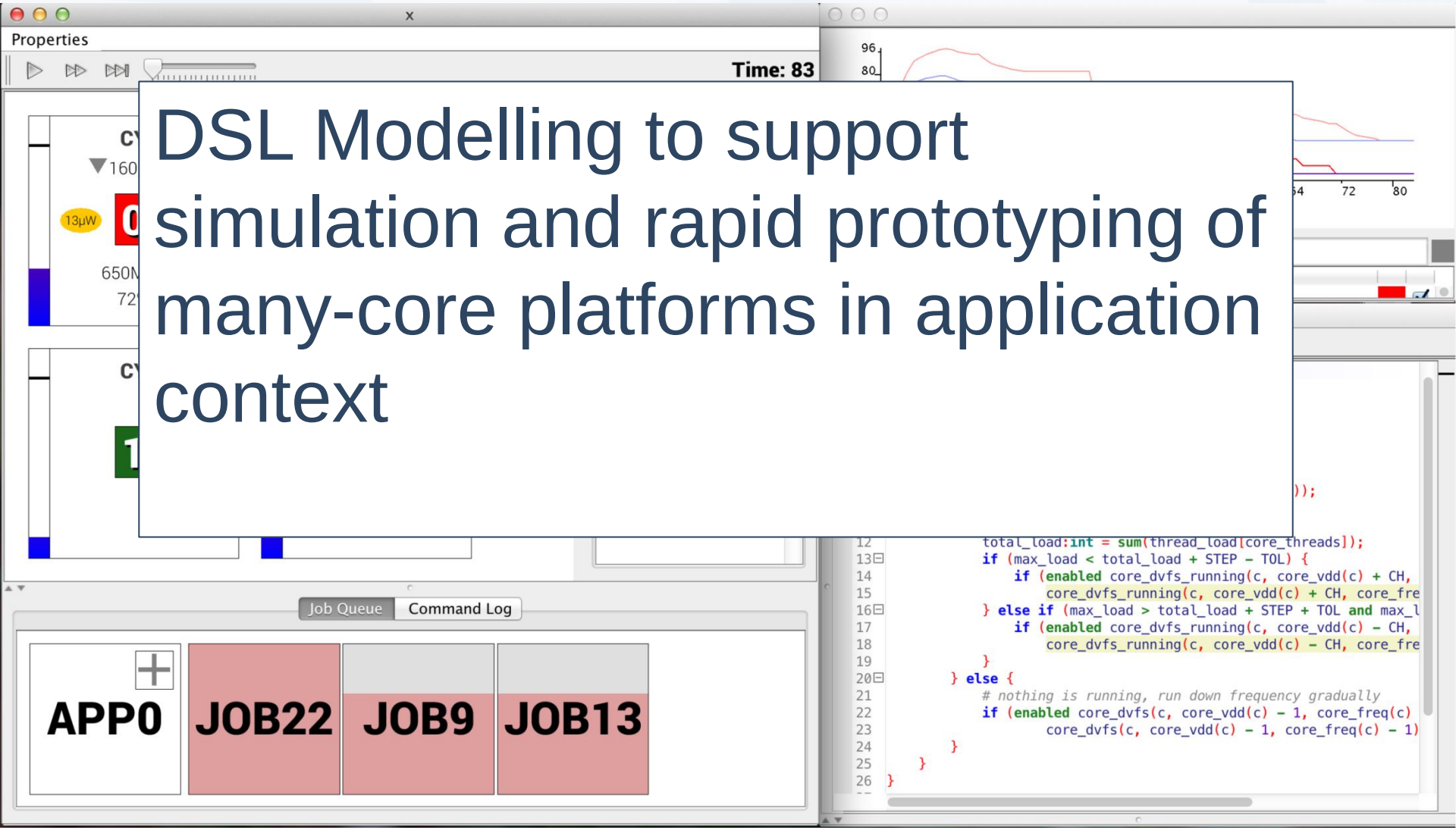
Since computation cost increases for lower orders, we can save computation by going iteratively from top.

Order Graph Modelling tool (demo)

PRiME
www.prime-project.org

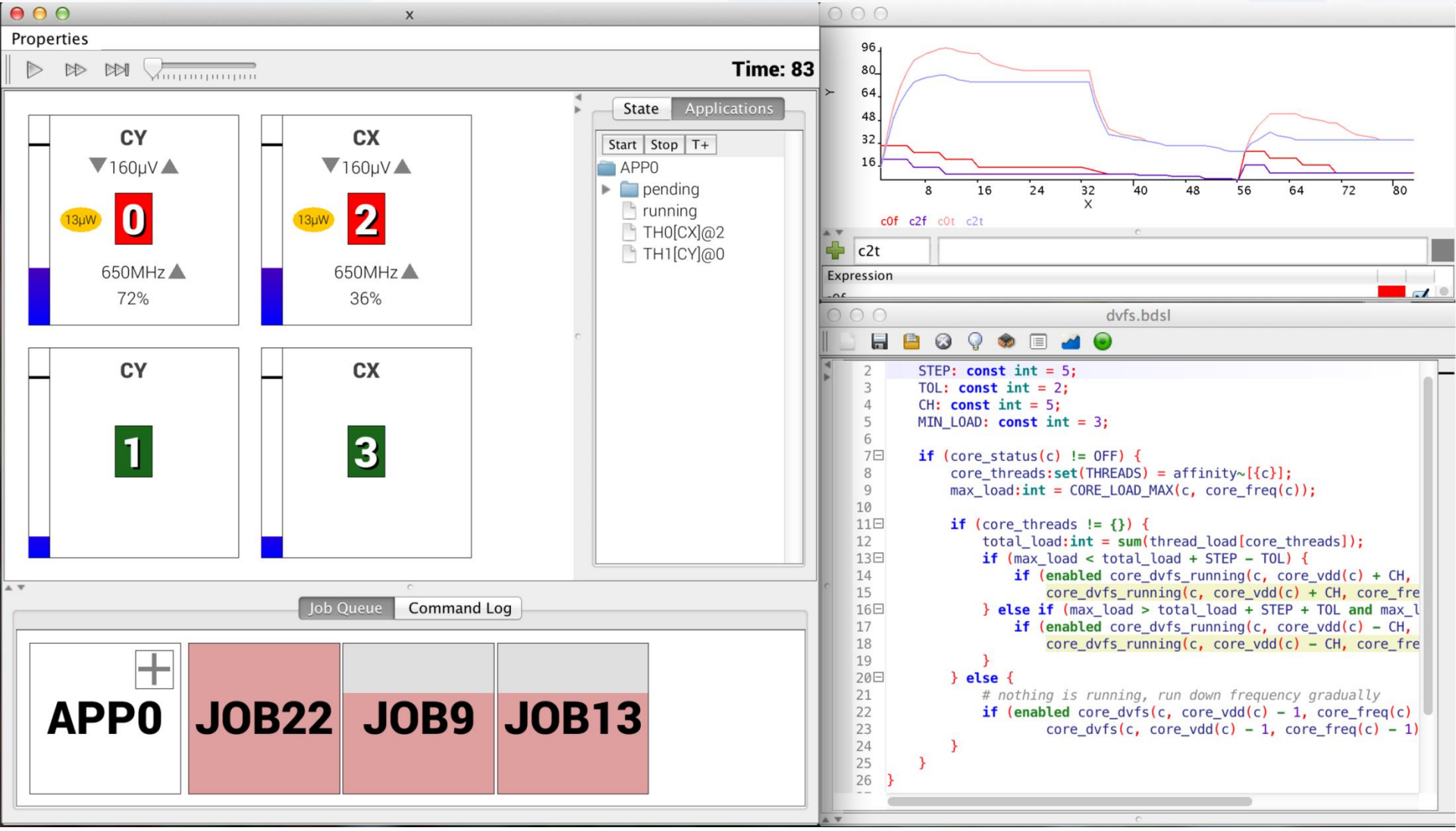
Domain Specific Language (DSL) Modelling

DSL Modelling



DSL Modelling to support simulation and rapid prototyping of many-core platforms in application context

DSL Modelling



Conclusions

- PER models help in assessing scalability and can be used in RTM.
- We propose a systematic approach for power-proportional (parametric-proportional) modelling, which is based on building cross-layer cuts.
- Future/current work: better tool support – graphical editing, formal proving (via integration with Event-B).

Use flat models on parameters.

Do not use functionally (structurally) flat models.