TeamPlay

http://teamplay-h2020.eu

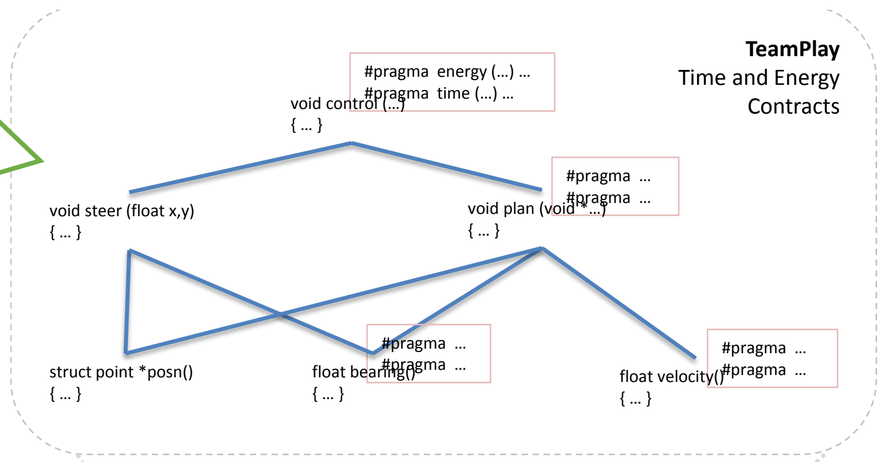
Time, Energy and security Analysis for Multi/Many-core heterogenous PLAtforms

This project has received funding from the European Union's Horizon2020 research and innovation programme under grant agreement No 779882.

- ***Context:*** parallel multicore/manycore and heterogeneous systems for mobile applications, IoT, …
- ***Problem:*** energy efficiency is critical, but no effective analyses can predict energy usage, and no analyses allow the programmer to balance properties such as energy efficiency, time, and security.
- ***Goal:*** to effectively manage execution time, energy usage, security, and other important non-functional properties of parallel, heterogeneous systems.

TeamPlay from the application programmer's perspective: we propose a toolbox that treats non-functional properties (execution time, energy usage, security, …) **effectively** and as **first-class citizens.**

Formally-verifiable and programmable contracts, expressed as source-level annotations on C programs. Contracts are specified internally using a DSL. Implied sub-contracts can be determined and verified automatically.



EnergyAnalyzer

energy ('control') = 0.24j

time ('control') = 0.315µs

Automatic static analyses based on models of energy usage, time, and security, allowing these to be determined and/or checked automatically and accurately.

TeamPlay Time- and Energy-Aware Compiler

```
control:
  move $1, energy
  xadd $sp, -16
  move $fp, $sp
  …
```

Multi-objective optimization techniques that balance conflicting demands on energy, time, security etc., taking into account the requirements of the software.

Industrial use cases in computer vision, satellites, drones, medical, and cybersecurity domains. We aim to support the use of heterogeneous, parallel multicore hardware in a wide range of systems. More advanced algorithms could thus be used on-board, providing new autonomous capabilities, conserving power, and providing static guarantees regarding non-functional properties.

TeamPlay Time- and Energy-Aware Runtime System

```
control:
  …
energy ('control',vel,t,…) = …
time ('control',posn,dir,…) = …
```

Run-time mechanisms to dynamically enforce system-wide energy, time and security properties. Run-time environment is aware of system-level goals and can enforce them by manipulating and re-configuring the scheduling of the system's individual applications.

TeamPlay from a technical perspective: we propose to consider key non-functional properties such as energy usage, time, and security systematically and at all abstraction layers, ranging from programming language level through multi-objective optimising compilation down to the runtime system level.

Energy, timing and security contracts.

- Energy, time, security and other properties are first-class citizens, reflected throughout the compilation, analysis, and runtime environment.
- Express energy, time, security etc. contracts formally as effectful operations, generated from an input program in a high-level language (C), based on information obtained using analysis/measurement techniques.
- Verify contracts through normal type checking mechanisms.

Multicore coordination.

- Multicore coordination considering energy as a primary concern
- System designer can specify complex energy constraints and energy-related optimisation targets.
- Combines coordination, real-time scheduling, and scheduling/coordination using sophisticated energy-specific constraints.

Compilation and optimisation.

- Compilation and optimisation in the context of energy, performance and security being exposed to the programmer as first-class citizens.
- Multi-criterial optimisations that are systematically able to trade energy usage with performance and security level.
- Six-dimensional optimisation space including both average and worst-case energy usage, execution time, and security.

Energy modelling and analysis.

- Energy usage of code determined using static analysis combined with advanced architecture level and resource usage modelling.
- Considers energy usage through the entire stack of abstraction levels: hardware, compilers, programming languages, coordination level.
- Energy transparency on multicore is addressed by exposing energy usage effects of inter-core communication and of interference due to accesses to shared resources.

Security.

- Focus on side-channel attacks when considering energy and time as an exploitable resource.
- Precise modelling of energy and time enables definition of security weaknesses through all abstraction levels.
- Automated countermeasures based on hiding techniques adapted for multicore systems.

Inria · ThalesAlenia Space · AbsInt · TUHH Technische Universität Hamburg-Harburg · SECURE-IC The Trusted Computing Company · IRIDA LABS · University of Bristol · SKY-WATCH · SDU University of Southern Denmark · Universiteit van Amsterdam · University of St Andrews