

Breakout Session: Heterogeneous Computing for IoT and CPS

NSF CSR/NeTS 2019 Joint PI Meeting
Arlington, VA
Nov. 4-5, 2019

Contributors

- Shuvra Bhattacharyya
- Alvaro Cardenas
- Roger Chamberlain
- Tam Chantem
- Changhee Jung
- Miriam Leeser
- Shivakant Mishra
- Mahdi Nikdast
- Massimiliano Pierobon
- Aviral Shrivastava
- Marilyn Wolf
- Heechul Yun
- Ting Zhu

Different Kinds of Heterogeneity

- Heterogeneity in CPS and IoT exists at many levels
 - Hardware platform
 - Communication channels and protocols
 - Software stack
 - Programming languages
 - Models of computation
 - Timing abstractions (deadlines, QoS, etc.)
 - Design evaluation metrics

Security for Heterogenous Systems

- Security is an important concern for CPS and IoT systems, since they are poised to manage many safety and mission critical applications, including electricity grid, transportation, etc.
- Security is an especially important challenge for IoT systems, as they will be composed of components designed by different manufacturers (heterogenous).
- Security is especially hard to solve in severely power-limited sensors, and intermittently powered sensors.
- Since timing is critical to correctness in IoT and CPS, the timing in the different parts of the system also needs to be secured.
- Security cannot be solved in a piecemeal manner. A globally consistent view of the system operation is required for efficient and invisible security.

Heterogeneity and Real Time, Low Power

- Programming abstractions must allow application level designers to leverage real time, low power, reliability.
- Analysis of timing and power is harder on heterogeneous platforms.
 - Transaction-level network-on-chip.
 - FPGA.
 - Heterogeneous CPS/IoT networks.
 - Useful worst-case bounds.

Towards a Science of Interoperability

- Not just an engineering challenge, or a matter of APIs.
- Interoperability includes both design time (e.g., abstractions) and run time (e.g., protocols) aspects.
- Must be able to build abstractions of complex components on multiple axes (function, timing, power, reliability, security).
- Abstractions must be able to survive redesign, degradation, etc.
- Compositions must be scalable to huge systems (continental power grid, etc.).
- Interoperable components must be able to manage their interactions with each other.
- Diagnosis and root-cause analysis are required to develop systems efficiently and reliably.
- Interoperability is interdisciplinary.