Design Tools for HPC SoC

Challenges, Opportunities, or Business as Usual?

X. Sharon Hu

Department of Science and Engineering

University of Notre Dame

To “SoC”, or not to “SoC”

- If HPC does not adopt SoC design, what are the alternatives?
- If HPC adopts SoC design, should there be one or two or more SoCs?
  - Cost vs. “efficiency”
- What IPs should go into the SoC(s)?

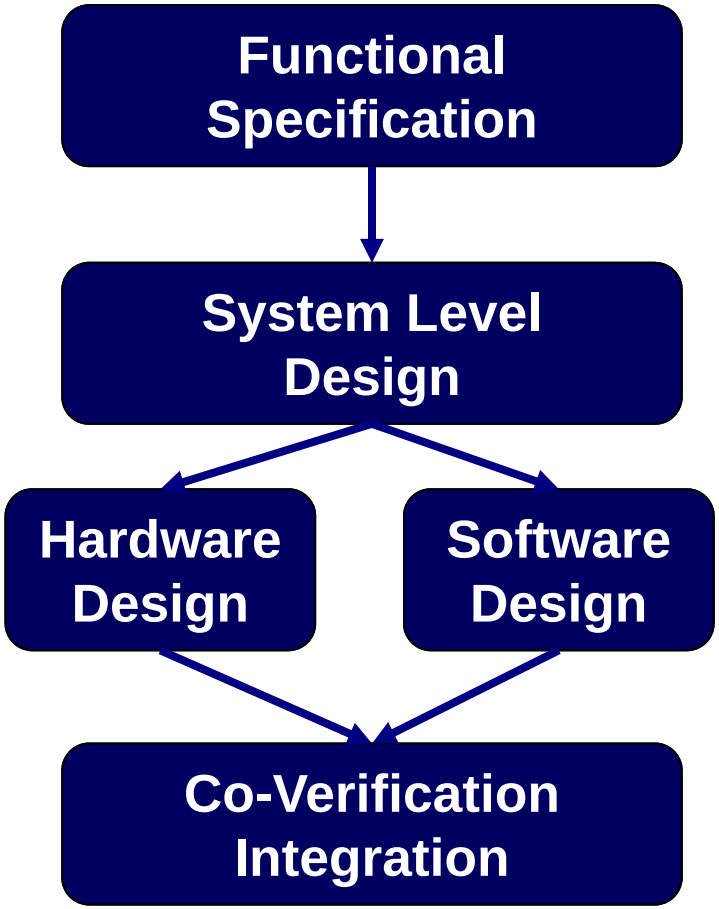
Electronic Design Automation (EDA) Tools

SoC Design for Embedded Systems

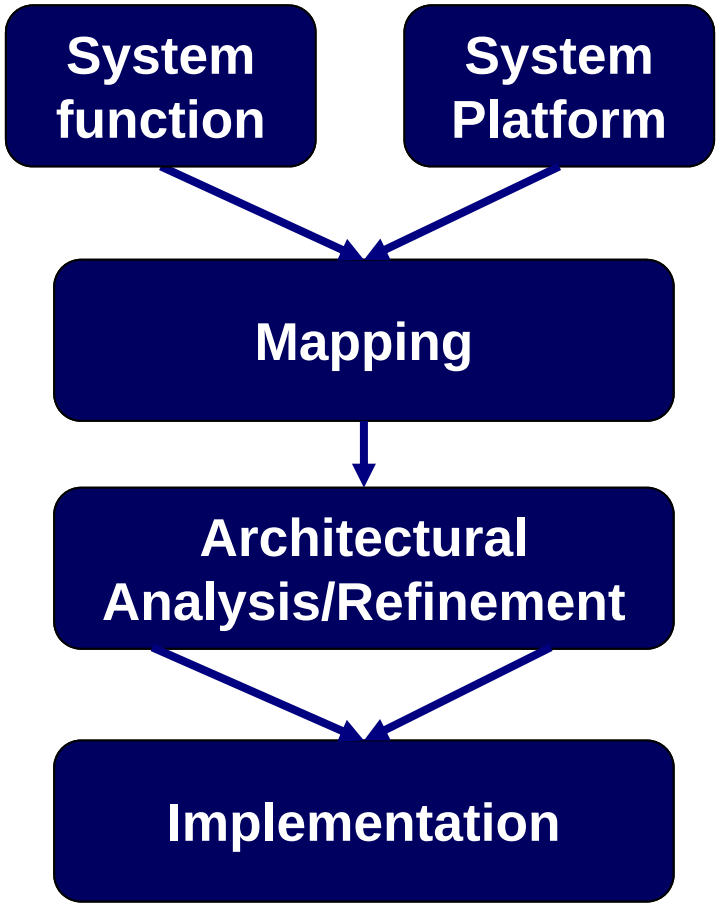
- IP reuse, SoC design and integration started close to 30 years ago
- Myriad of tools available
  - Commercial tools:
    - Carbon Design Systems (Model Studio, SoC Design Plus, CPAK)
    - Sonics (SonicsStudio Director)
    - Synopsys (System Studio, Platform Architect)
    - Cadence (System Development Suite)
    - …
  - Open source or academic tools:
    - Ptolemy II
    - Baya (edautils)
    - Verilator
    - …

ES SoC Design Methdologies

Top-Down Design



Platform-Based Design



Design Space Exploration

- Metrics: quantifiable
  - Performance (especially real-time)
  - Energy
  - Resiliency (lifetime, soft error tolerance,…)
- System-level design as multi-attribute optimization:
  - Pareto-optimal solutions
  - User-preference guided search

Design Space Exploration

- **System-level design as multi-attribute optimization:**
  - Given design x_i (i=1, … N) with attributes a_{ik} (k=1, … M)
  - Find the best design based on "user's preference" captured by an ***imprecise value function***

$$V_i = \sum_k w_k V(a_{ik})$$

  - Weight w_i are *implicitly* constrained by user's preferences, i.e., if user prefers x_i over x_j, we have

$$\sum_k w_k (V(a_{ik}) - V(a_{jk})) \geq 0$$

  - Comparing alternative designs becomes

$$\min \sum_k w_k (V(a_{hk}) - V(a_{ik})) \qquad \text{for all i}$$

$$s.t. \sum_k w_k (V(a_{ik}) - V(a_{jk})) \geq 0 \qquad \text{for all i and j}$$

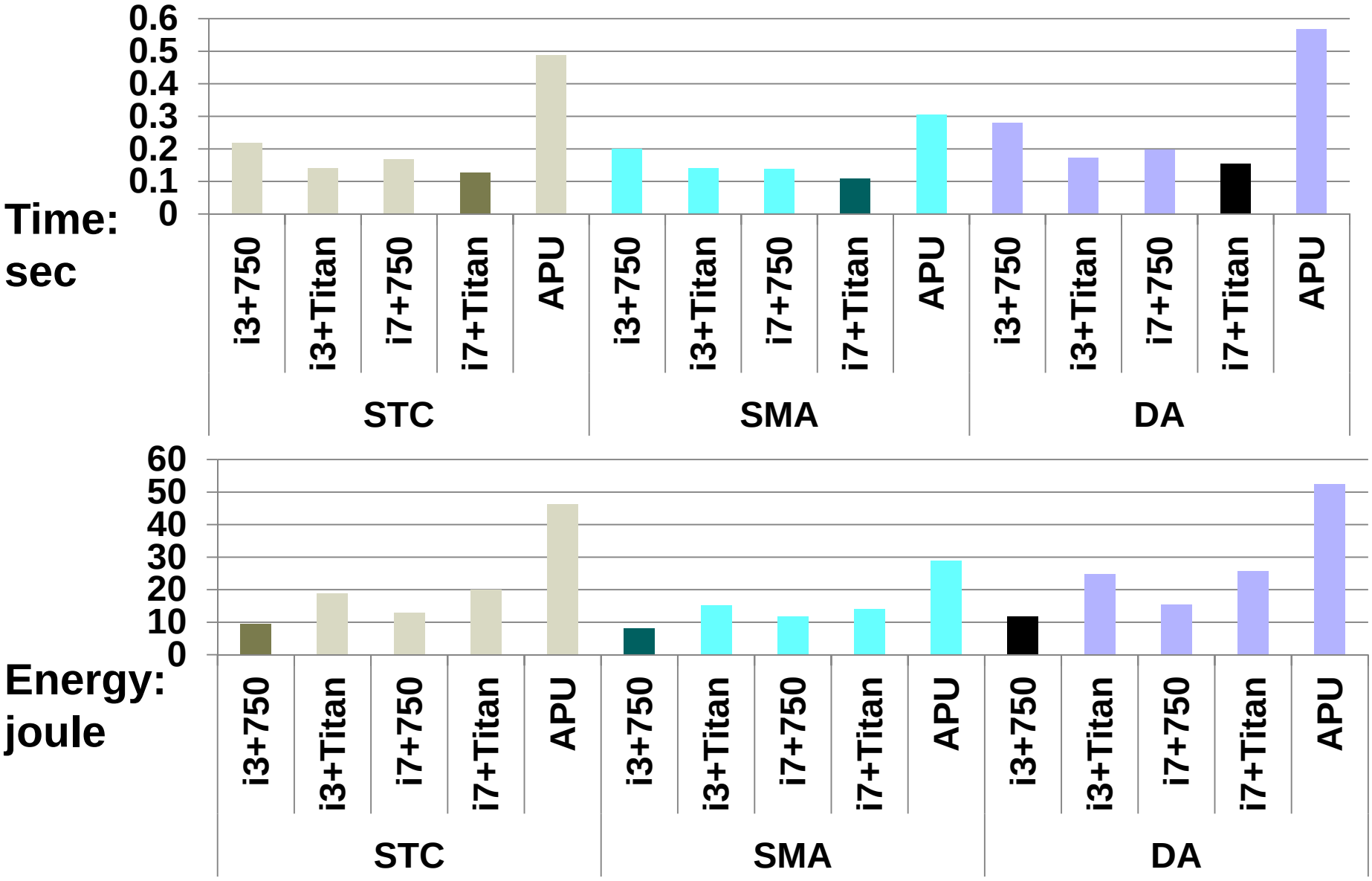
ES Design v.s. HPCS Design (1)

- Similarities
  - Ever increasing hardware capability: multi-core, multi-thread, complex communication fabrics, memory hierarchy, …
  - Increasing productivity gap
  - Common concerns: latency, throughput, energy, cost, reliability, fault tolerance, …

ES Design v.s. HPCS Design (2)

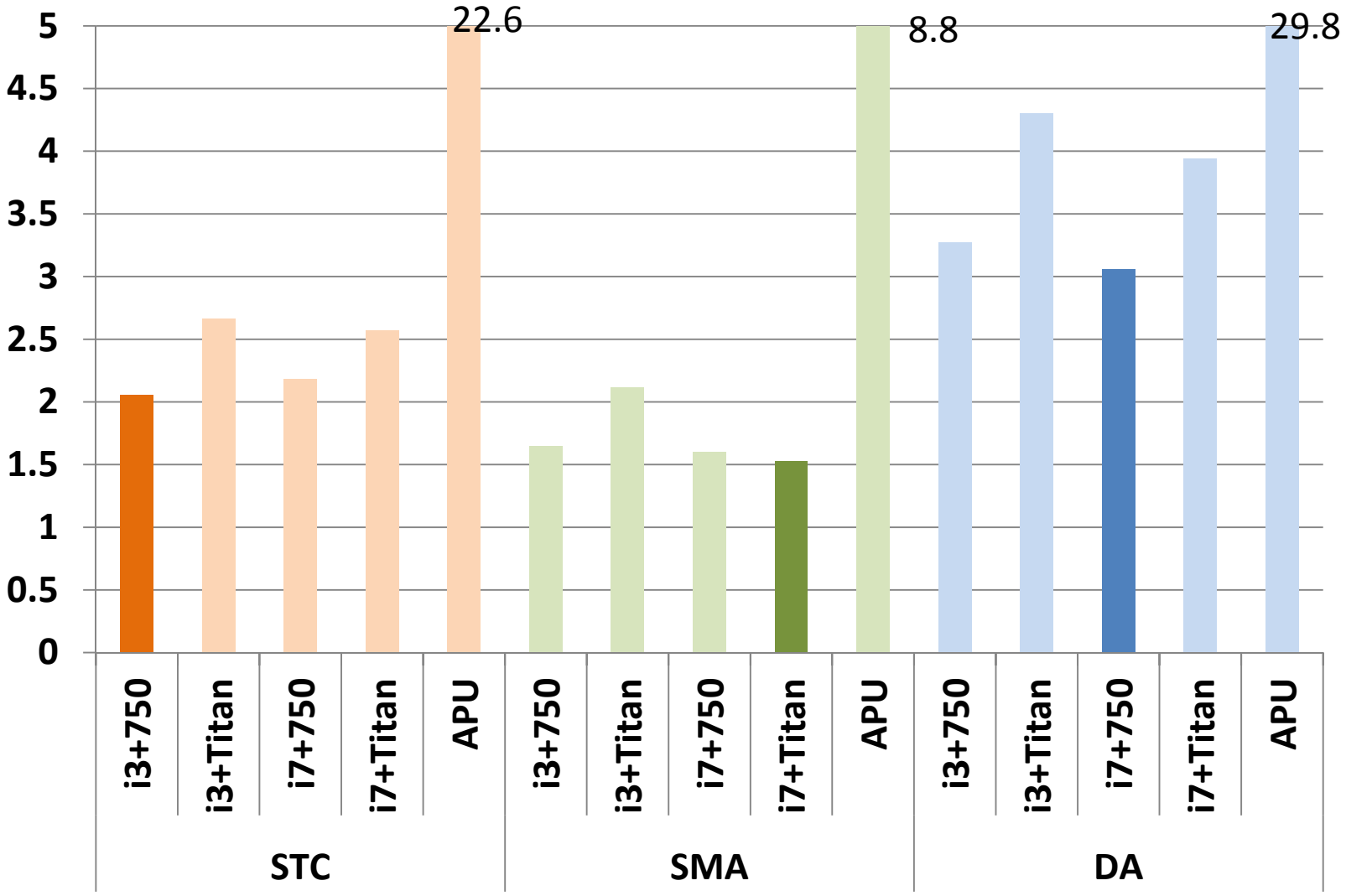
- Differences (maybe)
  - Application specific workloads v.s. domain specific workloads
    - Miniapps can be a great asset for addressing this aspect
  - More or less heterogeneity
  - Constraints, objectives, desirables?
    - Latency, throughput, energy, cost, reliability, fault tolerance, IP protection/privacy, T to M, …
  - Other issues: problem size, level of complexity, user expertise,…
- Facing similar issues even today

Which Platform Is Better?

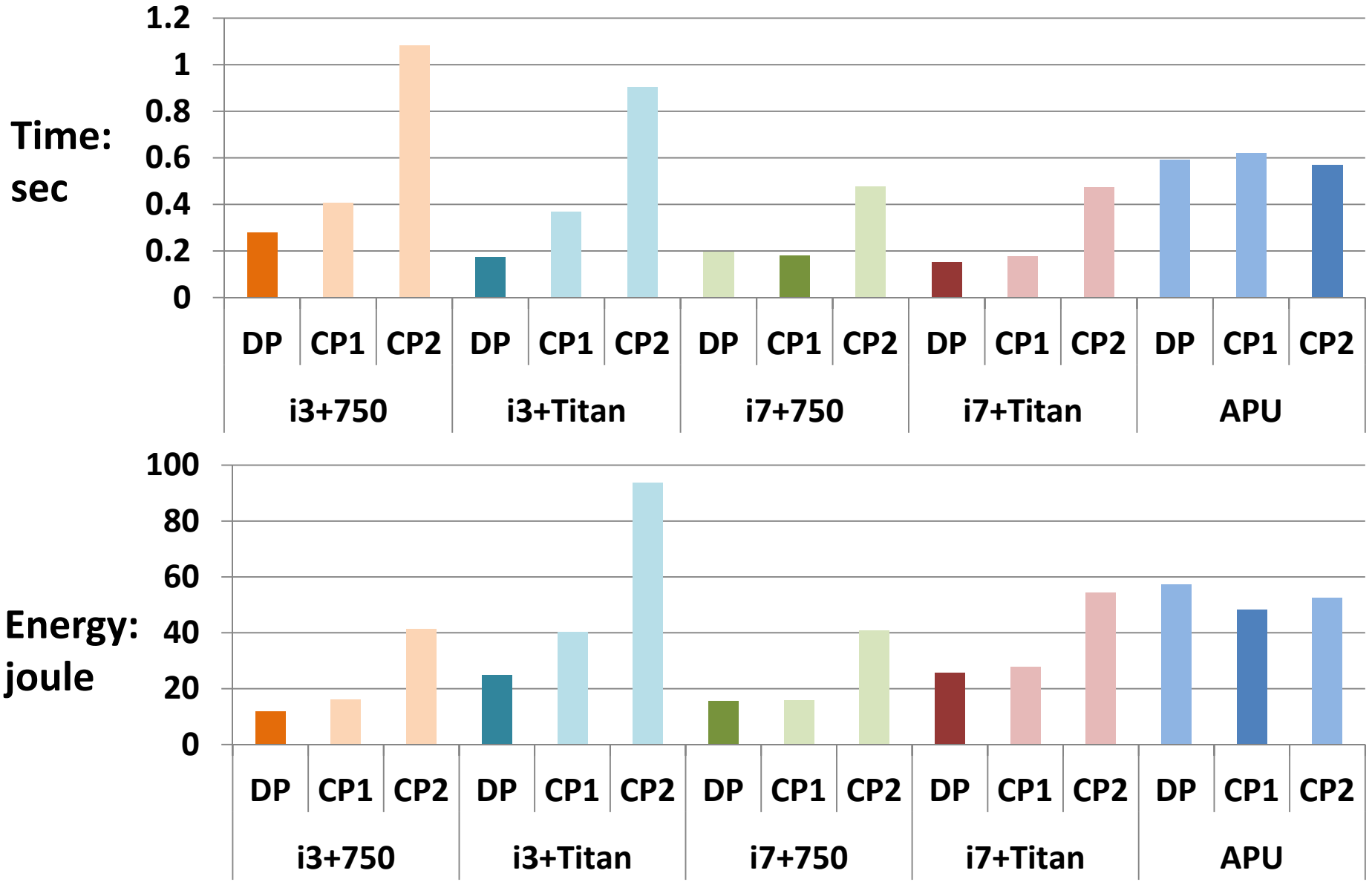


Which Platform is Better?

Unit: Joule*sec

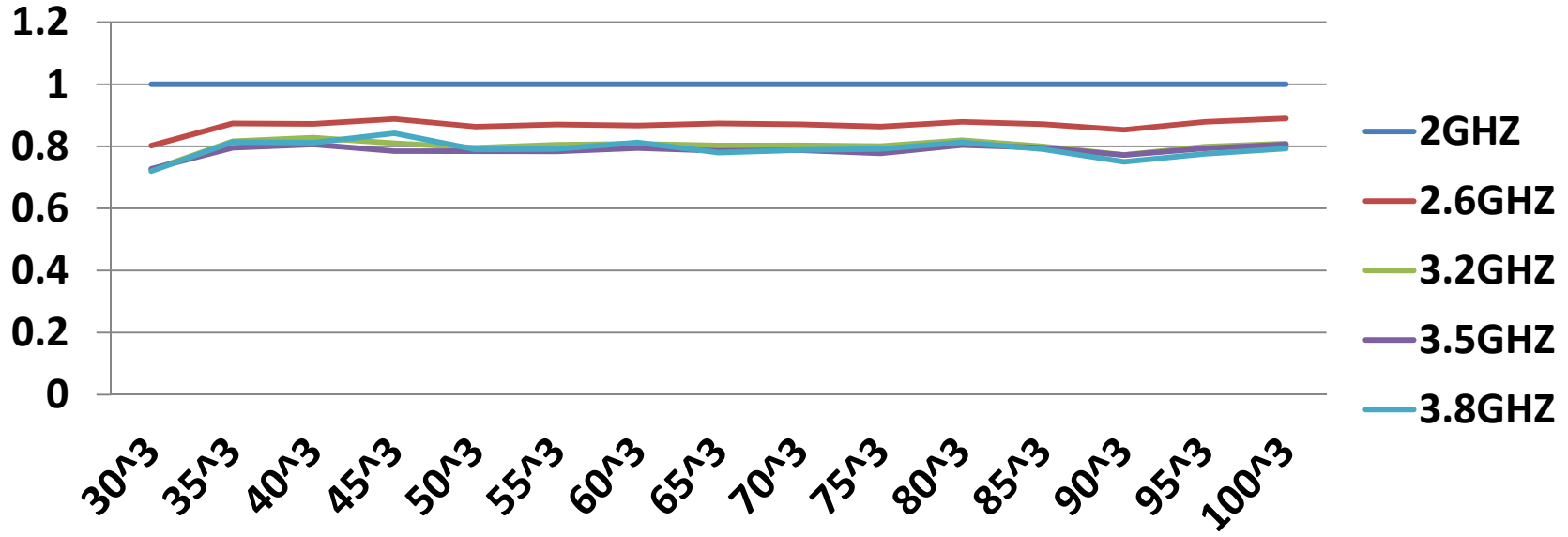


Which Partition to Use?

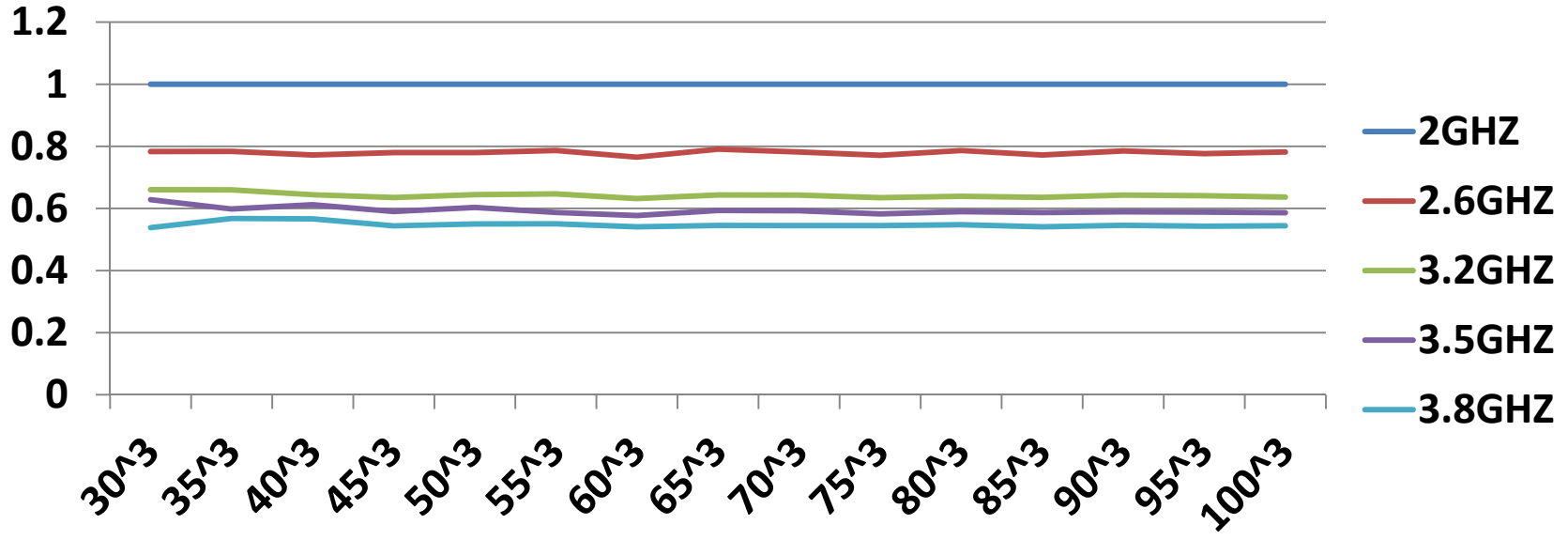


What Runtime Strategy to Use?

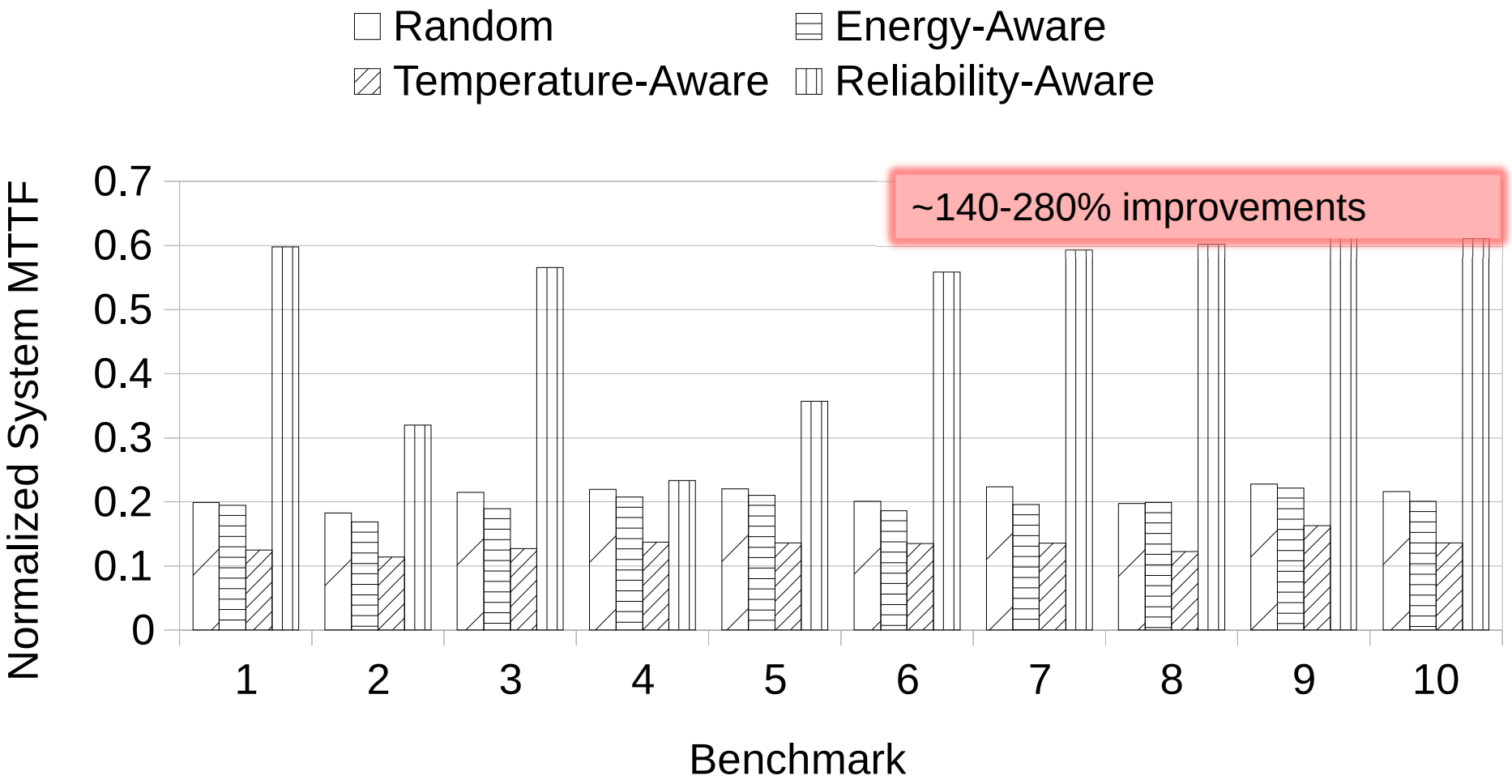
CG: speedup



DA: speedup



Runtime Reliability-Aware Load Balancing



SoC Design Tools for HPC

- Modeling
  - Multi-scale, multi-abstraction-level
  - More sophisticated system SW models
- Simulation and analysis
  - Miniapps
  - Methods to aggregate results for different applications
  - Memory and communication analysis and design
- Design space exploration
  - Multiple-attribute optimization
- Runtime resource management
  - Tradeoffs between “cost” and “benefit”

Thank you!

Comments?