

# HCW Panel: Programming heterogeneous systems - Less pain! Better performance!

José Fortes

*University of Florida  
Gainesville, FL, USA  
fortes@ufl.edu*

**Chair:** José Fortes

**Panelists:**

Richard Graham (Los Alamos National Laboratory)  
Alexey Lastovetsky, (University College Dublin)  
Samuel Midkiff (Purdue University)

**Abstract:** Heterogeneity in computing systems has been driven by, among others, one or more of the following factors: need for better performance (e.g. in multi-core chips), applications' requirements (e.g. in digital processing systems), timing and logistics of computer facility development (e.g. clusters that are extended and upgraded over time) and emergent systems of systems (e.g. for Grid-computing). The premise that these heterogeneous computing systems (HCS's) offer cost and performance benefits is true only if they can be efficiently programmed. The perspectives and questions on programming HCS's considered by this panel include the following:

1. Should programmers be exposed to heterogeneity so that they can squeeze all the necessary performance by taking advantage of the best resources for the jobs that need them? How can we handle the glut of programmers wishing to program such complex and extensive systems?
2. Should we design compilers that schedule and optimize programs written for homogeneous systems for execution on heterogeneous systems? Will the resulting programs run with better performance than they could achieve in a homogeneous system? How should better be defined in this context?
3. Are programming languages irrelevant in the sense that one can always connect components and/or services to accomplish any task? How generally applicable is this programming model?
4. Are HCS's inherently distributed memory entities where shared-memory programming models cannot succeed? Can shared- and distributed-memory models coexist? How do currently available languages fare in this regard?