Panel on the value of HPDC

Rob van Nieuwpoort
Introduction

- 10 years of Grid computing
- 5 years of many-core computing, radio astronomy
- Netherlands eScience center
- ASTRON: Netherlands institute for radio astronomy
- VU Amsterdam, CUDA teaching center
Our “products”

• Q: Is it our actual research that is of value, or is our main "product" our graduates, who can go to industry and apply the skills that we taught them?

• A: We must do both: science > training
  – We need venues for fundamental CS research and increase our value for industry
  – Are we teaching the right skills?
Our impact

Q: What is the impact of the research of the HPDC community?
   - Does the industry read and apply our ideas?

A: Impact in
   - Academia / labs: clearly a major impact (catch 22)
   - E-science: significant impact
   - Industry: some impact, room for improvement
High-impact work in HPDC

- Grids and Globus (LOFAR)
- Security in grids
- Cactus
  - Climate modeling, Hydrology, Astrophysics
- Application-level scheduling
- Nimrod
Our topics and motivation

Q: Do we work on the right topics and do we have the right motivation for our research?
   - Why is there little attendance from industry?

A:
   - We jump from topic to topic quickly
     - Grid, Cloud, p2p, many-core, BigData, Exascale
     - Many ideas never outgrow prototype stage
     - Duplication and fragmentation
   - This widens the gap between research and industry
The HPDC mix

• Q: Do we have the right mix of design, modeling, analysis and prototype development?
• A: More prototype / demo development
  – Move prototypes into production, analyze
  – Stronger applications track
    • GPUs, desktop grids and clouds bring HPDC ideas to the masses
Interesting topics for industry

• Applications
  – GPUs, desktop grids and clouds bring HPDC ideas to the masses

• Big Data: combine large heterogeneous distributed data collections
  – Volume of data, complexity of data (virtual labs)
  – Distributed sensor networks
  – How and where do we process

• Programming models / programmability
  – Many-cores: many explicit levels of parallelism
  – Scaling and fault tolerance
  – Memory-centric programming
    • Algorithms and the memory wall
    • Algorithms and energy