Notes Group 1
"Straw man outline and roadmap"

John Taylor (Moderator)
Bernd Mohr (Note taker)
Plan of CDC*: Anshu

- Tight interactions between all groups involved needed
- Need interactions with other CDC
  - Also to establish common terminology
- Process
  - Leaders of different teams start to design the process
  - Start with compact / skeleton apps
    - Create repository
    - Port different apps to new programming models (CUDA) and evaluate them
    - Regular meetings to synchronize with others (every 3months / 6months?)
  - As it takes about 3 years of transforming the production code
    - Need to start working in 2015 to be ready for 2018
- Time for Plan B? yes, given enough time/resources

*CDC = Co-Design Center
Plan of CDC: Alice

- Different context: multiple codes on various levels of quality
  - Bring codes to the same level first
  - Same for compact / skeleton apps
    - But make sure results/experiences make it back to the production codes
  - Experiments with coupling framework (facets)
    - Would start with facetizing compact apps
  - Use CoDEx tools: SST system simulation, and creating skeleton apps "automatically" with ROSE
  - If enough funding also interests in post-processing aspects (VisIT)
Plan of CDC: Jackie

- Very much like the other CDC
- Vertical integrated 3 teams
- Will also try to leverage CoDEx/SST ...
- Full stack important including I/O and storage
General comments on this processes

• How to synchronize / coordinate CDC + ESC?
  – Important for common repository for compact / skeleton apps
  – Also for common terminology

• Because of limited resources not all paths can be explored
  ⇒ Precise info needed from SW and HW teams
      to make the right decisions

• Reproducibility/Verification essential
  – Important for work in large team all touching the code

• Impact on HW might be difficult because of short window of possible influence
  – Reality = not enough resources to build the ideal exascale HW
  – Really need to start to work now
  – Need to have tools like SST (CoDEx)
Comments from SW: Bill G. / Barbara

• Need real data (not guesses) what code is really doing
• Performance / complexity models of apps are very important
  – Even multiple models to show range of values
• Create challenge for performance model teams:
  – Create models from skeleton apps automatically!
• Another issue: how and when to select the exascale programming models?
  – Already to late?!?
• How to get interactions between apps and prog model teams?
  – Jesus: EU text project example for MPI/SMPSs model
  – Anshu: need prog model people as part of the team
• Not enough time for complete new prog model design:
  – Plan to use C++ approach for prog model evolution:
    • Start with routine / API
    • Make it pragma/directive
    • Make it language feature
Key metrics for software

• Obvious: time to solution * power used
  – Solutions per Joule
  – Second priority: costs?
• Readability/maintainability of the code?
• Complexity: layer of code/ SW abstractions?
Summary Outline Roadmap

- 2012
  - Common terminology
  - Create compact/skeleton apps and repository
- to 2015
  - Use this to evaluate choices
  - Create performance / complexity models
- to 2018
  - Rewrite production code based on experiences gained