Insertion Paths Panel

Alice Koniges

Lawrence Berkeley National Laboratory/NERSC

ICiS Multiphysics Workshop

August 5, 2011

What challenges do application codes face in incorporating new algorithms?

a. Funding – new development is not funded?
b. Social – new methods are not explained in ways that applications scientists can understand, or math/cs researchers do not work in the areas of greatest need because they do not understand the challenges of the applications?
c. Technical – new methods do not exist that can improve on current methods?

What challenges do application codes face in incorporating new algorithms? Bullet a

- Funding – new development is not funded?
  - Depends on funding organization and extent of legacy code
    - NNSA not likely to fund new methods or to write a new code with a new method—too much “benchmarked legacy code” and designers comfortable with it
    - Office of Science – depends on length of new code and if they development can be fit into existing programs
      - Examples: M3D-C1
      - Ice Sheet
    - NSF/NASA/Others ?
    - Use of graduate students

What challenges do application codes face in incorporating new algorithms? Bullet b

- Social – new methods are not explained in ways that applications scientists can understand, or math/cs researchers do not work in the areas of greatest need because they do not understand the challenges of the applications?
  - A rewrite won’t happen for a small advantage – slightly better convergence, slight performance improvement
  - It might happen if:
    - Memory constraints cause old code to choke
    - Someone shows a much better method (e.g. through a grad student’s work)
    - There is a special funding initiative
  - Frameworks—like Chombo, Boxlib, even big solvers like Petsc, unlikely to undergo major changes because they are so widely used as is – more adaptable to incremental changes like NNSA codes

What challenges do application codes face in incorporating new algorithms? Bullet c

- Technical – new methods do not exist that can improve on current methods?
  - Even hybridization has not been shown to be a really big advantage yet
    - Example – new NERSC incentive to use more cores and teach hybrid
  - Many math improvements will in the future depend on architectures – we have come very far and discovered a large part of the algorithm space. Now the changes come from adapting them to the changes in architecture and programming languages

How do we design code to allow for reasonable insertion of new algorithms and software into applications?

- Obviously, modularity is key
  - Ability to swap out different algorithms in a multiphysics simulation
  - Multiphysics simulations are better for this, because can change a method while leaving other parts alone
  - Solvers are the most common way to try new things, e.g. through an interface that calls a library

How do we adapt legacy code to incorporate new methods?

- Again, depends on the world you are coming from, NNSA vs. certain office of science
- Fusion codes vs. chemisty codes with scores of users
- Some code will be changed only incrementally
- Other codes will be re-written from scratch with new programming models and algorithms
- Only when this last group of codes “jumps” ahead of legacy codes, might the legacy codes change

How can funding agencies support insertion of new methods into existing codes?

- They should first support the “skeleton approach” that we proposed in codesign – capture the small essence of a code, and put it out there for experimentation by CS and Math people
- Then, if/when a method is really superior, either a total re-write or incremental changes to support the new model will become obvious
- So the support process needs to be a two-pronged approach

A Blast from the Past (slide from last week's ASCR Programming Challenges Workshop)

- PATP – Parallel Applications Technology Program (largest DOE CRADA) for vector to parallel transition
- Approximately 10 large application codes
- About ½ rewrote from scratch, others added incrementally
- All those who wrote from scratch had parallel performing codes at least one year faster than those who modified
- We need to support those efforts who are willing to redesign and redo from the bottom up