Applications and Frameworks

Challenges of Climate Modeling
Infrastructures as Research Tools on
“Exascale Platforms”

Jeff Durachta
09/07/2011
Overview

• Introduction
  – A little about me
  – More questions than answers

• Computer model enabled research: FMS - A Case Study
  – A brief history of the last decade of software development and science at GFDL

• “Computer Science” vs “Scientific Research”
  – But it worked just fine on one processor...

• “Exascale” architectures, software development and scientific collaboration
  – Could we win the battle, but lose the war?

• Summary
Introduction

• A little about me
  – Started with IBM in the (waning) days of the 3090 vector and saw the birth of the POWER architecture
  – User application development for the IBM SP series
    • “Porting” was a euphemism for “parallelizing the *&^%&*! application”
  – Began working with the GFDL during their transition from vector to parallel in late 1990s
  – Relatively brief stint with the ill-fated Sicortex startup as lead for the user applications and performance group

• This talk: More questions than answers
  – What is the relationship between software development and scientific research?
  – What’s necessary for software development to enable science?
  – How can we continue to enable scientists to develop software?
Computer model enabled research
The “Flexible Modeling System” (FMS): A Case Study

• Developed by a small team of scientists and software engineers:
  – Common core infrastructure
    • Parallel communications, domain decomposition and connectivity, I/O, model diagnostic and history generation, etc
    • Data objects for communication between model components
    • Exchange grid infrastructure for boundary layer science

• Over last decade, there’s been significant development of the computer science as well as the earth science
  – “Cubed sphere” atmosphere
  – MPI / Shared Memory hybrid
  – Overlap of communication with computation

• Infrastructure has been stretched from terascale to petascale
  – Major infrastructure components have been re-written
  – Science marches on; but little is ever dropped
Computer Science vs Scientific Research
But it worked just fine on one processor...

• At least at GFDL, “computer science” is (almost) completely subordinated to the goals of scientific research
  – While the infrastructure group steps in to assist scientific software development, the latter largely remains in the hands of the scientists

• FMS infrastructure is designed and implemented to hide complexities of the parallel environment

• Enables those with a wide range of programming skills the ability to change code and try something new
  – Great (and perhaps essential) for science
  – Still lots of “But it worked.....”, especially in the first 3-4 years
  – Periodic stubbed toes over (F90+) language subtleties
“Exascale”, Software Development....

- Like others, GFDL has a “GPU Initiative” thanks to collaborations with Jeff Vettor, ORNL and Cray
- Like others, NOAA has understood that given what’s ahead, it has no choice but to have an “Exascale” Initiative as well
- With a number of people and organizations some represented here, I’ve recently worked on application projects for future processor architectures featuring
  - Actively managed memory hierarchies
  - “Co-processing” elements such as GPUs
  - Large thread count nodes
- From the standpoint of “Getting it done at any cost” there seems to be a lot both interesting and hopeful
  - As embodied by this conference
...and Science?

• As the lab works through the transition from “local” computing at GFDL to remote computing at ORNL
  – Technology alone does not enable science

• While I work with others on this new and exciting CS Geek stuff
  – Technology alone does not enable science

• If we succeed in enabling applications on these complex new platforms
  – But “access” remains only for the “Techno-rati”
  – With a vast gulf between “Developers” and “Users”

• Could we win the battle, but loose the war?
  – Can collaborative, computer aided science continue “as we know it”?
Summary

• Recent progress such as that being reported here gives me hope that the “technology wall” is not unassailable from a CS perspective

• What give me pause?
  – We seem to be having some success figuring out how to do things “the hard way”
  – But this is not the environment that has made for successful collaborative science not just at GFDL, but other groups with whom I've worked

• The still unanswered and largely untouched question to me is:

  How do we enable collaborative scientific software development?