Development Experiences
WSSSPE 2013

Juliana Freire

Polytechnic Institute of New York University
NYU Center for Urban Science and Progress
NYU Courant Institute
NYU Center for Data Science

Summary

	MVAPICH	khmer	MINRES-QLP	DUNE	VisIt	VisTrails	FLASH/ Chombo
Description	MPI libraries	nucleotide sequence analysis tools	suite of Krylov subspace methods for solving linear systems	partial differential equations	large scale visualization	reproducible data exploration	structured adaptive mesh refinement
product experience or practic	product	product	product	product	product	product	products
Location	Ohio State et al.	Michigan State	CI, U of Chicago, ANL	HPC Simulation Software & Service	LLNL, Sandia, LBNL, et al.	NYU, LLNL, NASA, USGS	U. of Chicago
License	BSD			GPL-2 + “runtime exception”.	BSD	BSD	
Longevity	12 years	4 years		6 years	13 years	9 years	
Developers	developers from multiple institutions	7 developers/ contributors		developers from multiple institutions	developers from multiple institutions	developers from multiple institutions	
Software Engineering Practices		No		Yes	Yes	Yes	Yes
User base	183k downloads, available through major Linux distributions, adopted in many production environments				>200k downloads, 35k visits to visitusers.org	>40k downloads	
Mailing lists	Yes				Yes	Yes	
Funding	NSF	NIH, USDA			DOE	NSF, DOE, NASA	

NYU

CUSP CENTER FOR URBAN SCIENCE+PROGRESS

NYU·poly

Lessons Learned

- *Developing and maintaining software is **hard***
- Software engineering practices help
  - Version control, documentation, and testing are a must!
- But sometimes are hard to follow…
  - Documentation is hard to keep up to date
- Documentation is useful both for users and (current and future) developers
- Backward compatibility is not always desirable and can be costly
  - Clean up code, remove unused features → reduce maintenance cost

Lessons Learned and Challenges

- Community building and involvement are Important for sustainability
  - Mailing lists, bug tracking, training
  - Time consuming and expensive
  - *Mailing lists: other users don't often answer other users' questions*
- Managing growth in user base
  - Need *distributed* development and the infrastructure to support it
  - Luckily, the are many open-source tools

Lessons Learned and Challenges

- Recruiting qualified personnel
  - Steep curve to learn a large code base
  - Domain experts often lack programming expertise; programmers often lack domain expertise
    - Need to train *domain scientists* in software development and best practices
  - Lack of long-term career paths in academia for interdisciplinary researchers who build tools
    - The Moore-Sloan Data Science Initiative: http://nyti.ms/1bABf1D
  - Competition from industry – much higher salaries

Some Additional Challenges

- Little funding for software maintenance
  - Commercialization as a path, e.g., DUNE
  - Have larger, well-funded projects as customers
  - Long tail suffers…
- Balance between research and software development
- Rewards for developing and maintaining software
  - NSF now requires “Products” instead of publications in the biosketches
  - Tenure committees are starting to consider software
- How to measure impact?
  - Citations, downloads, visits to Web site, mailing-list traffic
  - Code re-use

Conclusions

- Software tools are crucial for science
  - Lots of data, complex computational processes
  - Need for transparency and reproducibility
- Developing and maintaining them is hard, expensive, and often hard to fund
- Funding agencies have tried with limited success, but not much for the *long tail*
- Need better *CS* training for scientists that do data-intensive research
- Need long-term career paths for applied researchers that cross domains

NYU

CUSP
CENTER FOR URBAN SCIENCE+PROGRESS

NYU·poly

Thanks