CyberGIS Toolkit: A Software Toolbox Built for Scalable cyberGIS Spatial Analysis and Modeling

Yan Liu [1,2], Michael Finn [4], Hao Hu [1], Jay Laura [3], David Mattli [4], Anand Padmanabhan [1,2], Serge Rey [3], Eric Shook [5], Kornelijus Survila [1], and Shaowen Wang [1,2]

[1] CyberInfrastructure and Geospatial Information Laboratory (CIGI)

[2] National Center for Supercomputing Applications (NCSA)
University of Illinois at Urbana-Champaign

[3] GeoDa Center for Spatial Analysis and Computation
Arizona State University

[4] Center of Excellence for Geospatial Information Science
U.S. Geological Survey

[5] Department of Geography
Kent State University

CyberGIS All-Hands Meeting 2013 *Seattle, WA., September 15, 2013*

Outline

- Purposes
- Software integration approach
  - Software component selection
  - Software engineering
  - Scalability analysis
- Progress update
  - CyberGIS Toolkit 0.5-alpha release
  - Continuous integration framework
  - Deployment on advanced cyberinfrastructure
- Case study: Parallel PySAL (pPySAL)
  - Parallel PySAL project
  - Illustration of parallelization strategies
- Future work and concluding discussion

Six Major Goals of the NSF CyberGIS Project

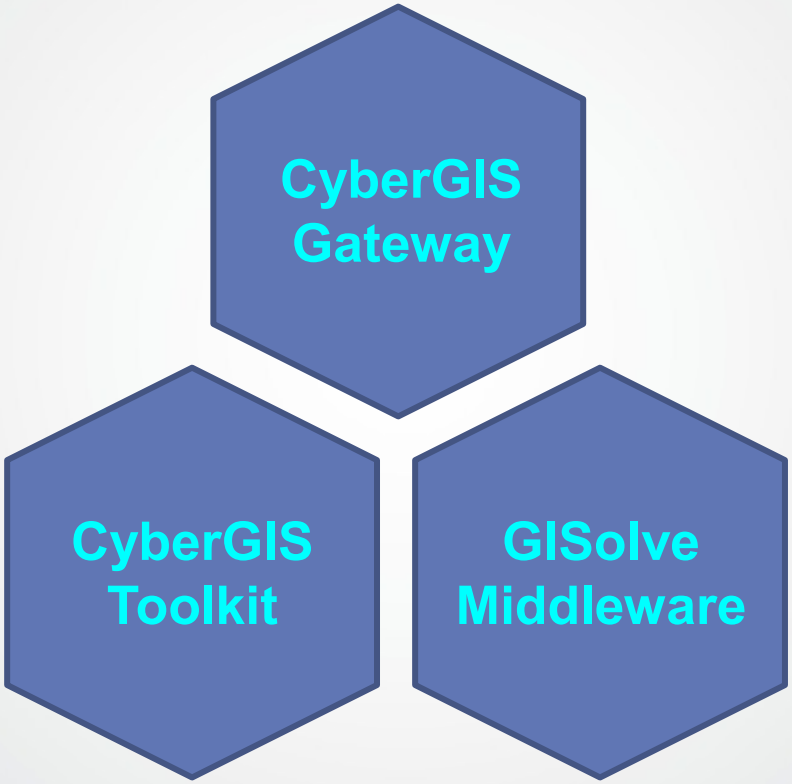
1. Enga communities through a participatory approach to re requirements;
2. Integrate and sustain a core set of composable, interoperable, manageable, and reusable CyberGIS software elements based on community-driven and open source strategies;
3. Empower high-performance and scalable CyberGIS by exploiting spatial characteristics of data and analytical operations for achieving unprecedented capabilities for geospatial scientific discoveries;
4. Enha patial problem solving environment to allow for the contr learning of CyberGIS software by numerous users e development of crosscutting education, outreach and training programs with significant broad impacts;
5. Deploy and test CyberGIS software by linking with national and international cyberinfrastructure to achieve scalability to significant sizes of geospatial problems, amounts of cyberinfrastructure resources, and numbers of users; and
6. Evalu e CyberGIS framework through domain science appli partnerships to gain better understanding of the comp man-natural systems.

Software as deliverables

Building blocks

Computational solutions

CyberGIS Software Environment



"CyberGIS Toolkit, a deep approach to CyberGIS software integration research and development, is focused on developing and leveraging innovative computational strategies needed to solve significant geospatial scientific problems by exploiting high-end cyberinfrastructure (CI) resources."

-- NSF CyberGIS Project, 3rd Year Annual Report (2013)

Objectives

- Identify and integrate a set of loosely coupled scalable geospatial software components into the CyberGIS Toolkit
- Establish and sustain the CyberGIS Toolkit as a reliable software toolbox through an open and rigorous software building, testing, packaging, and deployment framework
- Capture computational and spatial characteristics of a software element focusing on computational performance, scalability, and portability in various CI environments
  - XSEDE (Extreme Science and Engineering Discovery Environment. http://xsede.org)
  - Open Science Grid
  - Extreme-scale supercomputers
- Provide a software environment for computational and data scientists to easily configure and use CyberGIS Toolkit components

Software Integration Approach

- Software component selection
  - Open source strategy
  - Community-driven component identification
    - Benefits to related science areas
- Software engineering
  - Complexity in software integration
  - A holistic framework for streamlined component integration
- CI-based scalability evaluation and enhancement
  - Computational intensity analysis – theoretical approach
  - Performance analysis and profiling – experimental approach

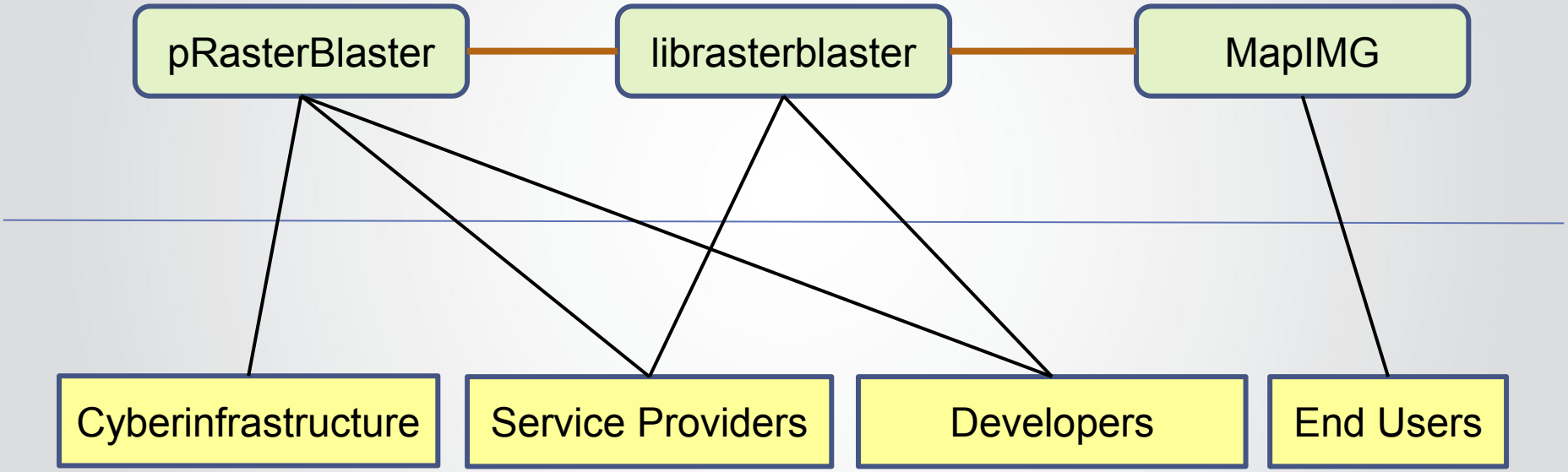
Open Source Strategy

- Components
  - Each component must be open sourced
  - No restriction on any particular open source license
- Toolkit
  - Improves the accessibility to integrated software capabilities through the establishment of a software toolbox
  - Focuses on the robustness, portability, compatibility, and scalability of each component within advanced CI environments

Component Selection Example: pRasterBlaster

- Need for scalable map reprojection in cyberGIS analytics
  - Spatial analysis and modeling
    - Distance calculation on raster cells requires appropriate projection
  - Visualization
    - Reprojection for faster visualization on Web Mercator base maps
- pRasterBlaster integration in CyberGIS Toolkit and Gateway
  - Software componentization: librasterblaster, pRasterBlaster, MapIMG
  - Build, test, and documentation
  - Gateway user interface

pRasterBlaster Component View



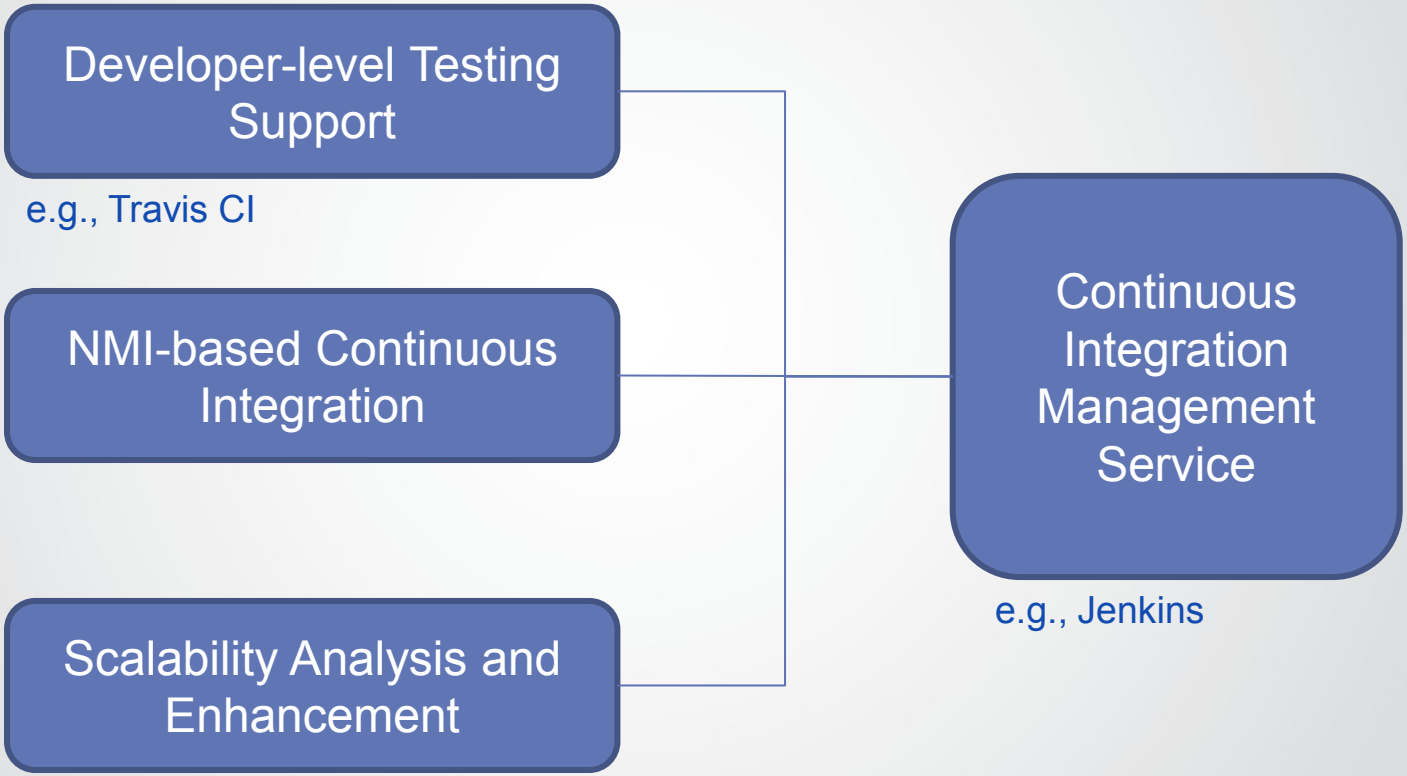
Integration Challenges

- Diversity and heterogeneity of software components
  - Programming languages
  - Application *vs.* library
  - Dependent libraries
  - Code availability
  - Programming models
  - Cyberinfrastructure resources
- Multiple levels of integration
  - Desktop level
  - Heterogeneous software environments
  - Cyberinfrastructure
- Software distribution
  - CI deployment
  - Packaging for broader distribution

Strategies

- Build and test
  - Establish a streamlined process to build and test software codes
- Computational intensity analysis
  - Computational bottleneck evaluation
  - Scalability analysis
  - Generalize solutions as computational and spatial knowledge
- Packaging and distribution
  - Package software for download and build in user computing environments
  - Provide common distribution packages (Debian, RPM, Windows installer, etc.)
  - Establish CyberGIS Toolkit as a configurable software module that can be loaded/unloaded on supercomputers
- Documentation and training
  - Build a CyberGIS Toolkit web site to host the software suite, user guide , development documentation, education materials, user feedbacks, and forums
  - Develop online training materials

Continuous Integration Framework



NMI: National Middleware Initiative (http://batlab.org)

Component Integration Process

- Selection
- Parallelization
- Development of test cases
- Development of build and test plans
- NMI regular build and test
- Scalability analysis on CI
  - Identification of potential computational bottlenecks
  - Scalability to the number of processors
  - Scalability to problem size
- Release in the CyberGIS Toolkit
- Accessibility in CyberGIS
  - Cyberinfrastructure deployment for high-end users
  - Lowering access barriers for community users through Gateway and GISolve
  - Incorporation in advanced application workflow

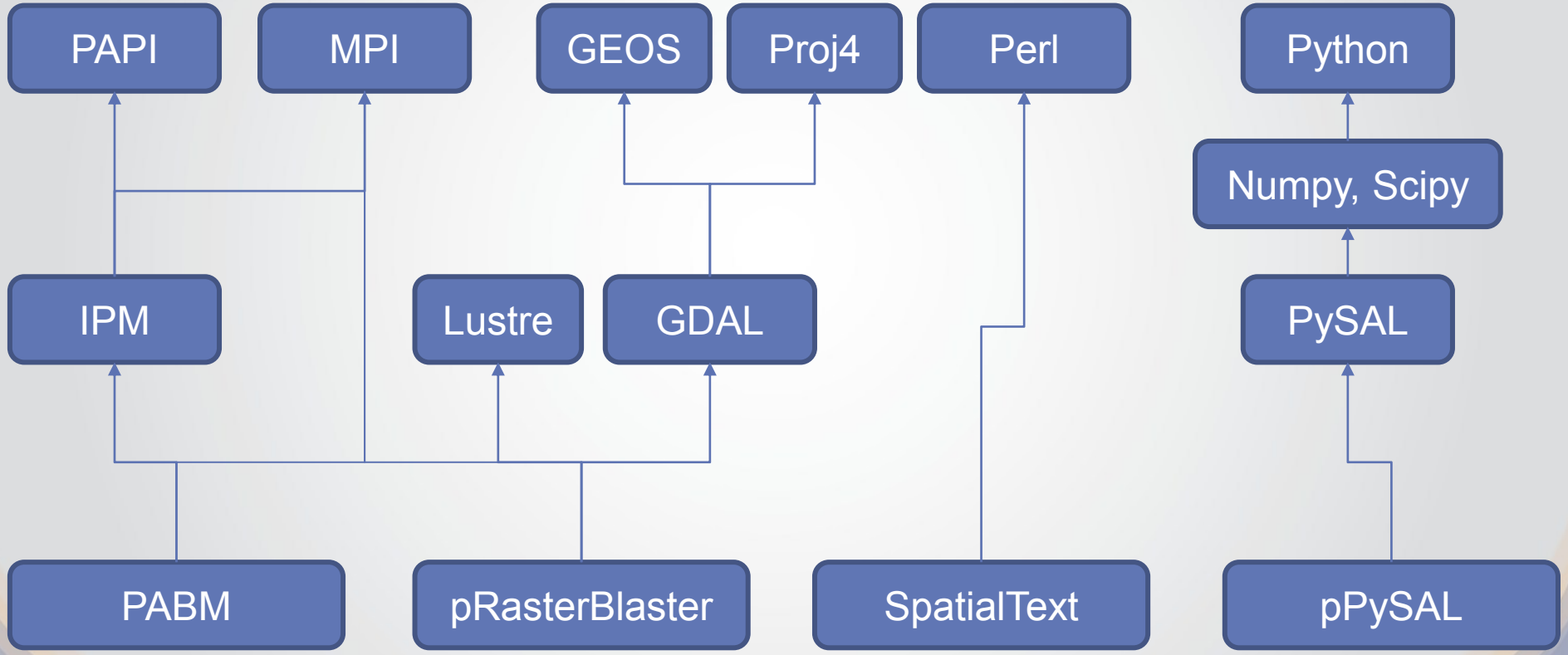
Demo

- CyberGIS Toolkit 0.5-alpha release
  - http://cybergis.cigi.uiuc.edu/cyberGISwiki/doku.php/ct

Software Components

- Parallel Agent-Based Modeling – PABM
  - HPC models: MPI, parallel I/O
  - Contributor: UIUC team
- Parallel PySAL
  - Parallel python implementation
  - Contributor: ASU team
- Parallel map reprojection – pRasterBlaster
  - HPC models: MPI, parallel I/O
  - Contributor: High-performance mapping group, CEGIS, USGS
- SpatialText
  - Full-text geocoding of massive social media and text data
  - Contributor: Kalev Leetaru and UIUC team

Dependency Graph – CyberGIS Toolkit 0.5-alpha



Toolkit Deployment on XSEDE

- XSEDE resources
  - Stampede@TACC: 10 petaflops; cluster computing, multi-threaded, GPU computing
  - Lonestar@TACC: 0.3 petaflops; cluster computing, multi-threaded, large memory
  - Trestles@SDSC: 0.1 petaflops; cluster computing, data I/O intensive computing
  - Gordon@SDSC: 0.34 petaflops; cluster computing, data I/O intensive computing, large memory
  - Blacklight@PSC: 0.037 petaflops; shared memory
  - Keeneland@NICS: cluster computing, GPU computing
- Compiling and install
  - Compilers: Intel, GNU, PGI
  - MPI: Open MPI, mvapich2, mpich2, PGI, Intel MPI
- Software environment configuration
  - Environment Modules
  - Adaptive software module loading/unloading
  - Demo

Case Study: Parallel PySAL

Serge Rey, Jay Laura

Concluding Discussion

- The first set of selected software components has been integrated into the CyberGIS Toolkit 0.5-alpha
- A prototype integration framework has been established to allow sophisticated multi-level integration tests
- Scalability analysis has been effective to identify computational bottlenecks and improve the performance of several components
- CyberGIS Toolkit has been deployed on XSEDE for community access
  - Through GISolve Open Service API
- Community evaluation and feedback is critical for building high-quality and scientifically sound software

Acknowledgements

- NSF Software Infrastructure for Sustained Innovation (SI2) Program
- This material is based in part upon work supported by NSF under Grant Number OCI-1047916. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation

We need your feedback!

Contact: dev@cybergis.org

Thanks!