

A New Paradigm for Scientific Computing

April 11, 2001

Introduction & Historical Notes

The last twenty years have seen a profound change in the model of scientific publication that has held sway since the founding of the Royal Society in the mid-17th century. The rise of computational research in the second half of the 20th century has burdened authors with the need both to present results and to make those results (and the algorithms that produced them) accessible to readers and colleagues. With the advent of PC-workstation computing in the 1980's, the central super-computer model of serious computing was phased out in favor of the more accessible local desktop computing. The development of the World Wide Web furthered this de-centralized model by allowing not only local research at a workstation, but the wholesale importation of software, algorithms, and results over Web. The result is a model of scientific publication now so common in many scientific disciplines as to qualify as a paradigm: the *methodology-diffusion paradigm*.

New complications in the structure of collected data and the development of more computationally intensive procedures such as 3-D imaging have recently forced the single-workstation to the upper limits of computational capacity. Without easily manageable experimentation for new research techniques, true diffusion of ideas and extension of recently published work is restricted to those with unusual computational capacity. Furthermore, the diversification of computing platforms and software environments used for serious research implies that the transfer of code alone may not be sufficient to enable experimentation even if the computing power is available locally.

Unfortunately, the development of processor power in an individual processor unit does not keep pace with the intensification of data and computing requirements. However, instead of returning to the central super-computing model, high throughput computing is beginning to turn to processor farms (distributed computing models) that stack multiple CPU's into a single system where jobs are split and distributed to processors. With distributed computing

systems, it is now possible to build relatively cheap and easily expandable processor farms to accomplish intensive computational tasks efficiently.

Recent trends in information technology make a new model possible, where research consumers upload data to a Web portal that offers research methodology as a service so that processing does not take place on the user's own computer. We propose that the combination of a cross-platform web interface with a centralized processor farm unit allows a unique opportunity to offer a widely available interactive library of scientific publication computing. We envision a processor farm hosted by an institute or funding organization where research time on the processors is available in return for the permanent installation of algorithms (an interface for automatically installing is provided).

In our prototype, to appear (later) at www.beamlab.org, we interface established code modules for experiments in harmonic analysis (using Wavelab/Curvelab type algorithms) with associated web forms. By stacking modules, the user may upload data and perform new experiments, retrieving results from the web server. Similarly, authors may organize code into modules, and establish pre-set job 'stacks' in our interface that allow users to automatically re-create published figures.

Such a standardized system requires both a sponsor organization (to maintain the hardware and interface) and the dedication of authors to add their research to the library. However, it has the benefit of establishing a clear reproducibility standard and allowing authors to automatically generate formatted documentation. Universally readable documentation and permanent installation of software on the interface also prevents the lag time in re-opening closed research cases, and the openness of the interface ensure the figure parameters and sequences can be automatically and transparently saved for future use.

“The actual scholarship [in a scientific publication] is the complete software development environment and the complete set of instructions which generated the figures.”

-Wavelab and Reproducible Research Technical Report, 1995.

■ Reproducibility as Portable Software

- (1) Portable software has a target computing platform (Unix, PC, Mac)
- (2) Requires local processing
- (3) May require specific installed software (such as Matlab) or specific types of computing environments

Methodology Diffusion Paradigm

Existing Examples

Wavelab: David Donoho and others, Stanford Statistics Department.
Downloadable code package for Matlab with basic harmonic analysis functionality. Includes documentation, several reproducible book excerpts, and tutorials. Extensions available for Curvelab and Beamlab. www-stat.stanford.edu

SEPLib: Jon Claerbout and the Stanford Exploration Project, Geophysics.
Fortran90 Library + local GUI/Graphics programs for 8-bit color. Uses Makerules to construct and re-construct published ‘Easily Reproducible Figures.’ Available on SEP by request or in CD-ROM format. sepwww.stanford.edu

Why Go Beyond Portable Software?

- **Increasing Data Size and Structural Complexity**

Many jobs are too big for most convenient local computing, limiting the amount of experimenting researchers can do.

Examples: *3D medical imaging, Internet Traffic Data, Geophysical Estimation, Image database problems, Curvelets & Ridgelets on large images*

- **Diversifying Software Base**

Experimentation with portable code requires a serious investment of time, brain power, and/or tech support.

Existing packages (random selection): *Fortran (SEPlib), C (Rainbow) , Splus (CART, MART, etc.), Matlab (Wavelab),...*

- **Increasing Reliance on PC computing**

Basic Unix packages are easy, but no longer standard; new software packages must be prepared for PC platforms, Linux, or Unix.

New Paradigm

(1) Accessible Computing

- Any user should have access both to the code and to the computing environment in which it functions efficiently - including the necessary computing power.

(2) Transparent Documentation

- Documentation should be universally readable (a daunting task), and the implementation of a published algorithm should be easily accomplished.

- The system should be 'stable' across platforms and software libraries.

Precedent-setting projects:

- NOVA analysis framework for physicists at Brookhaven National Lab. Web interface for relational database and basic analysis to facilitate research and ensure uniformity.

- WITS web navigator for Mars Polar Lander to allow both simulations and remote joint scientific control of experiments.

Proposed Web Portal RSP (prototype)

<http://www.beamlab.net> & <http://www.beamlab.org>

- **Distributed Computing System**

Processor farm (installed at Stanford) with web interface access to install and use research algorithms. Installed basic harmonic analysis functionality

- **Three Access Modes**

(1) Reproduce a published figure

(2) Run an existing algorithm with specified data and parameters

(3) Upload data and perform basic transforms & functions

- **Invariant Web Interface**

Stable on all machines, with automatically generated module forms for function interactions. Web Form submission of data and jobs, and retrieval of results.

Example of a Web-Reproducible Figure

↓ Original article, with PDF hot-link or URL reference.

56 Journal of Make-believe Examples

The last twenty years have seen a profound change in the model of scientific publication that has held sway since the founding of the Royal Society in the mid-17th century. The rise of computational research in the second half of the 20th century has burdened authors with the need both to present results and to make those results (and the algorithms that produced them) accessible to readers and colleagues. With the advent of PC-workstation computing in the 1980's, the central super-computer model of serious computing was phased out in favor of the more accessible local desktop computing. The development of the World Wide Web furthered this decentralized model by allowing not only local research at a workstation, but the wholesale importation of software, algorithms, and results over Web. The result is a model of scientific publication now so common in many scientific disciplines as to qualify as a paradigm: the *methodology-diffusion paradigm*.
New complications in the structure of collected data and the




Fig 2a: Example of estimated student count by major using the priors from section 1. www.beamlab.org/carrie/example/Fig02a

development of more computationally intensive procedures such as 3-D imaging have recently forced the single-workstation to the upper limits of computational capacity. Without easily manageable experimentation for new research techniques, true diffusion of ideas and extension of recently published work is restricted to those with unusual computational capacity. Furthermore, the diversification of computing platforms and software environments used for serious research implies that the transfer of code alone may not be sufficient to enable experimentation even if the computing power is available locally.
Unfortunately, the development of processor power in an individual




Fig 2a: Example of estimated student count by major using the priors from section 1. www.beamlab.org/carrie/example/Fig02a

Fig 2a: View Script Documentation

Fig 2a: Edit Figure

Download PDF

↑ Interface generated by RSP.
Options include reviewing script documentation, altering stack or parameters, or downloading a PDF of the figure.




Fig 2a: Example of estimated student count by major using the priors from section 1. www.beamlab.org/carrie/example/Fig02a

Functions	Arguments
load_data	edudata .

↑ Web form of the Fig2a computation with editable parameters. Jobs can also be assembled independently.
Interface generated by RSP.

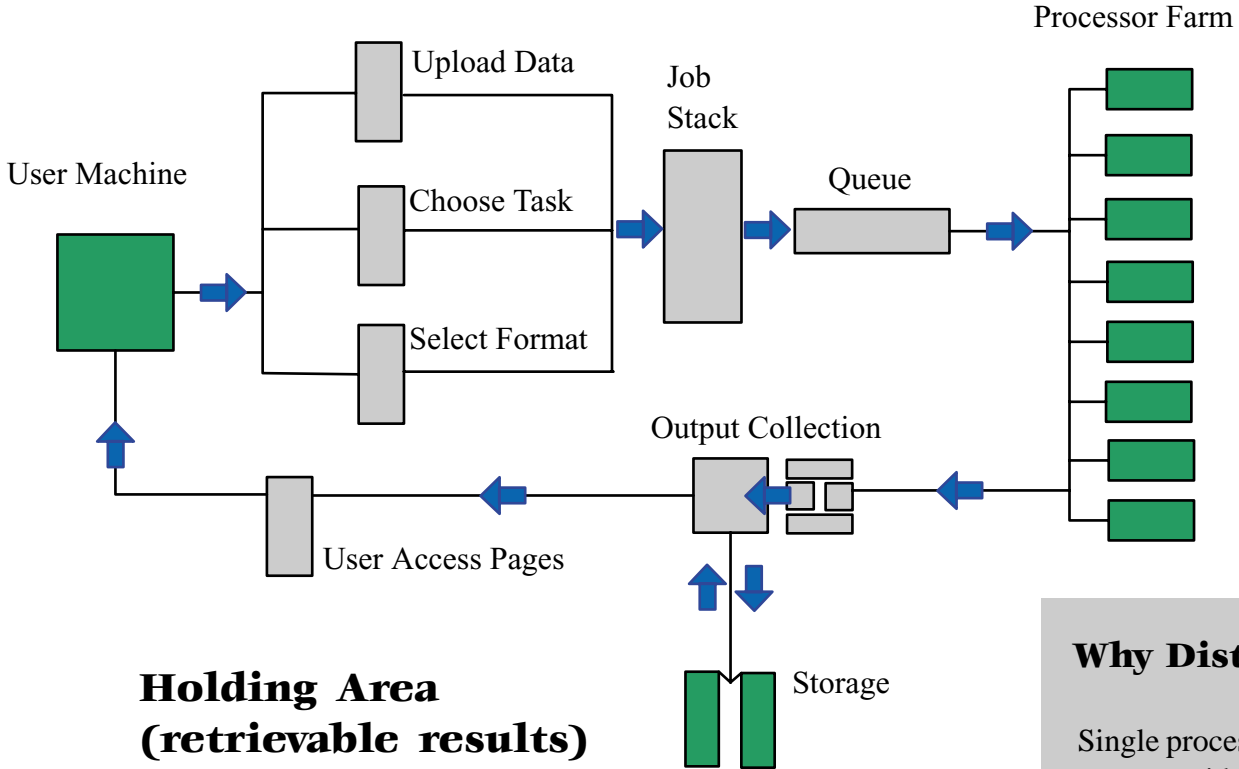
Proposed Portal System Diagram

Web Server Interface (job construction)

Processing (actual computing)

User at
Web Portal

[http://
www.beamlab.org](http://www.beamlab.org)

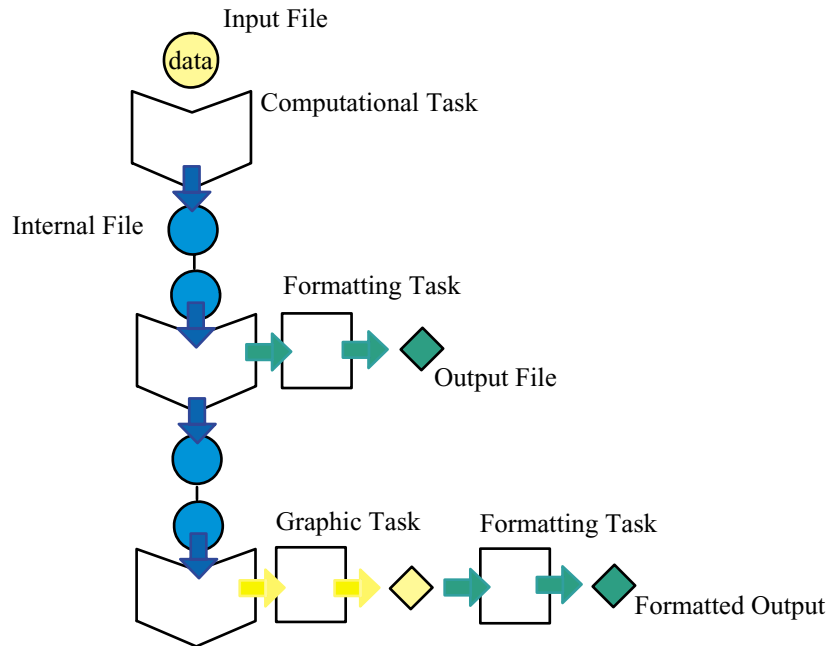


**Holding Area
(retrievable results)**

Why Distributed Computing?

Single processor farms mirror the benefits seen on wide-area networks of PC's such as the SETI@home project. SETI@home uses volunteer CPU's to compute during screensaver time. For \$500,000, SETI has 15 TeraFLOPS, while IBM ascii white supercomputer cost \$110 million for 12 TeraFLOPS.

Prototype Job Structure



- Upload/Select dataset
- Select or Construct a Computation ‘Stack’ using the form/menu system
- Tap the Stack for Desired Output
- Format Output for Retrieval

Design Issues:

- Linear module stack: ‘fire brigade’ pattern of software. Optimal for Matlab-type routines, less optimal for hierarchical code.
- Reproducible figures generated by pre-assembled job stacks

Author/User Two-sided interface

Adding to the available methodology: Using cgi-scripts, we generate new interface pages using an author-interface. From individual code files, the script strips headers to create a function information file. From that file, user calls for the function generate a form entry for the web page.

