Automated Web-based Analysis and Visualization of Spatiotemporal Data

Dylan Keon

Ph.D. Thesis Defense
Geography

November 16, 2012

Major Advisor: Dr. Dawn Wright

Introduction

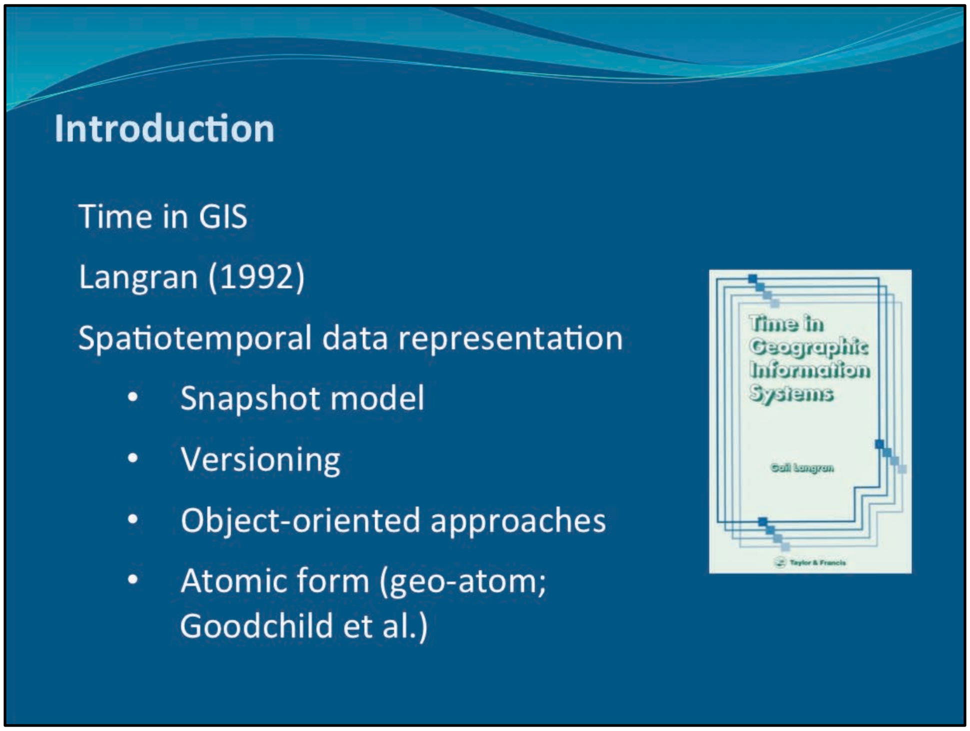
Spatial data increasingly common

Temporal component

- Dynamic phenomena
- Change over time
- Visualization

Spatiotemporal data

GIS software – great for spatial data

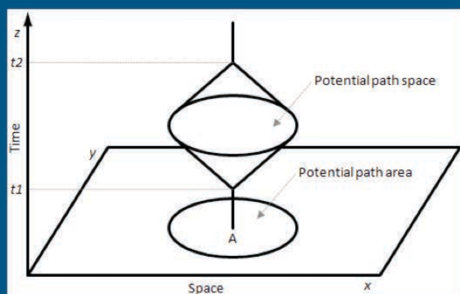


Time in GIS has been studied for a long time.
Key to the implementation of temporal capabilities in GIS is the idea of spatiotemporal data representation, or how to best represent the data digitally.
Gail Langran addressed this in her seminal 1992 book, Time in Geo Info Sys, addressing all of the techniques considered at that time.
One of the simplest techniques, but still useful, is the snapshot model, in which a new snapshot of the dataset is taken at each time step.
Another approach is versioning, where only data that have changed are represented at each time step.
Object-oriented approaches have also been used, with entities represented as objects with inheritable properties.
Another approach is the atomic form, or the geo-atom, as proposed by Michael Goodchild and others. The geo-atom consists of a point at a particular space and time, with a particular property.

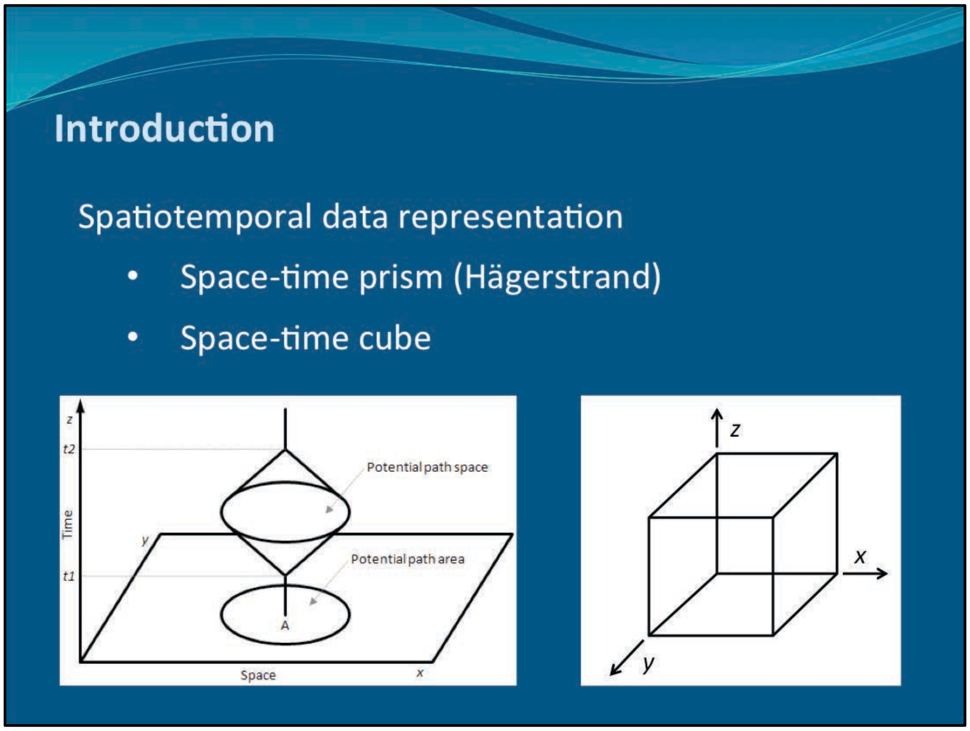
Introduction

Spatiotemporal data representation

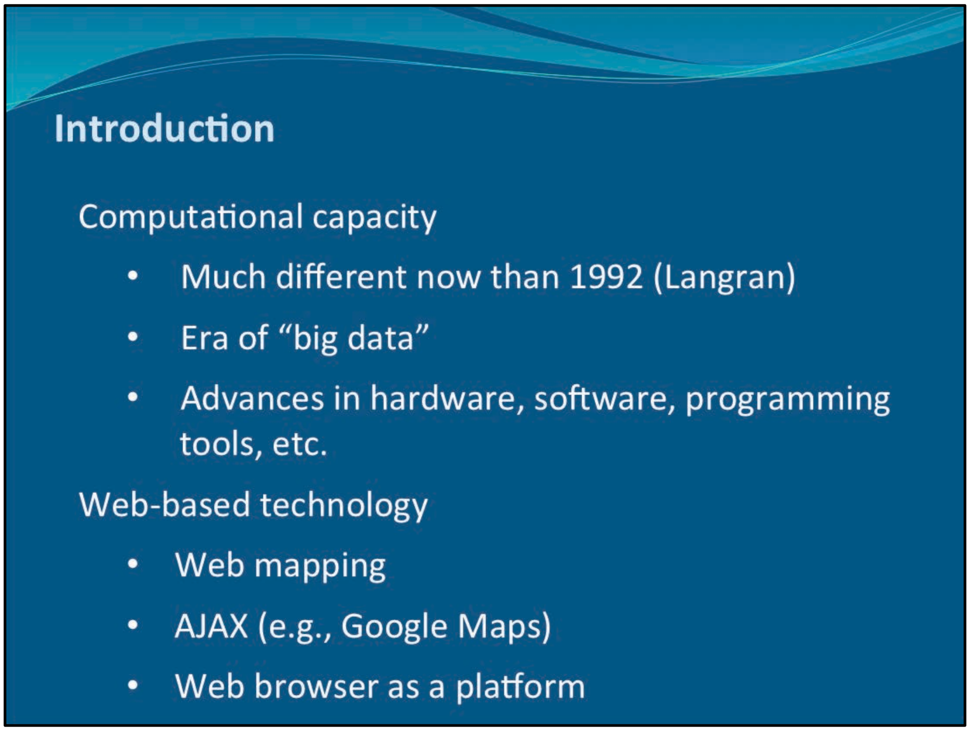
- Space-time prism (Hägerstrand)



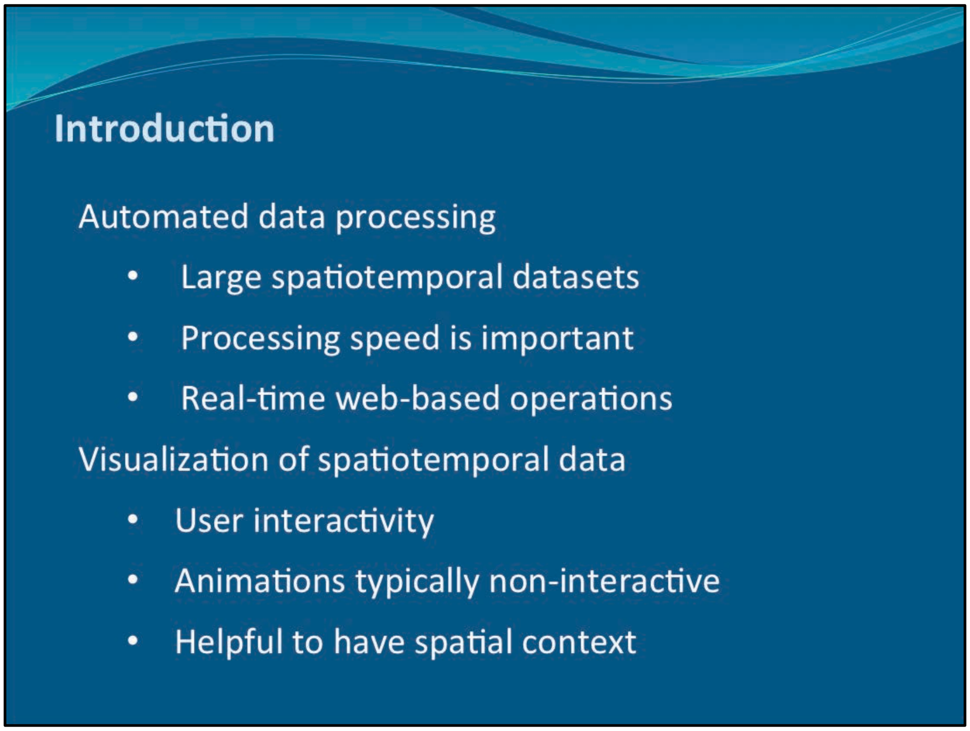
But sort of underpinning all of these approaches is the concept of the space-time prism, put forth by Hagerstrand in 1970.
The space-time prism is a representation of all the points an individual can possibly reach in a given time interval, represented by the potential path space.



Hagerstrand space-time prism concept also gave rise to the idea of the space-time cube.
X and y represent the spatial dimensions, while z represents the temporal dimension.
An individual's movement through space and time can be represented by points and lines within this cube.
If you can imagine data at each time slice along the z axis, this can be considered a representation of a multidimensional spatiotemporal dataset.



Time in GIS, and the representation and processing of spatiotemporal data, have all taken on increased focus recently.
There is a renewed interest in techniques for dealing with the temporal dimension.
Although many of the conceptual challenges mentioned by Langran in 1992 persist, computational resources today are wildly different.
This is truly the era of big data, requiring a lot of storage capacity and processing power.
Spatial databases are one means of effectively storing and representing spatiotemporal data, and there are some interesting software tools available for this today.



Also central to this approach is the automation of data processing tasks.
Particularly with large multidimensional spatiotemporal datasets.
Processing speed is important – faster is always better.
For real-time processing operations, we need automated techniques that can perform quickly when kicked off on the server-side.

Research Questions

- Can complex systems that process and visualize multidimensional spatiotemporal data be developed entirely in web-based platforms?

Research Questions

- Can complex systems that process and visualize multidimensional spatiotemporal data be developed entirely in web-based platforms?
- Do existing open source software packages provide enough functionality and flexibility to support the development of these systems?

Research Questions

- Can complex systems that process and visualize multidimensional spatiotemporal data be developed entirely in web-based platforms?
- Do existing open source software packages provide enough functionality and flexibility to support the development of these systems?
- Can user interactivity be extended to both the input side (i.e., defining simulation parameters) and output side (i.e., manipulating visualizations, performing queries) in these systems?

Research Goals

- Automate processing of spatiotemporal data

Research Goals

- Automate processing of spatiotemporal data
- Automate generation of visualizations based on user input

Research Goals

- Automate processing of spatiotemporal data
- Automate generation of visualizations based on user input
- Create server-side systems that enable fast querying and processing of spatiotemporal data

Research Goals

- Automate processing of spatiotemporal data
- Automate generation of visualizations based on user input
- Create server-side systems that enable fast querying and processing of spatiotemporal data
- Animate time-series data in output visualizations, while retaining user interactivity

Research Goals

- Automate processing of spatiotemporal data
- Automate generation of visualizations based on user input
- Create server-side systems that enable fast querying and processing of spatiotemporal data
- Animate time-series data in output visualizations, while retaining user interactivity
- Develop systems using open source software tools

Research Goals

- Automate processing of spatiotemporal data
- Automate generation of visualizations based on user input
- Create server-side systems that enable fast querying and processing of spatiotemporal data
- Animate time-series data in output visualizations, while retaining user interactivity
- Develop systems using open source software tools
- Develop generalizable methods that can be re-used in other systems

Project 1

Spatial data configuration, storage, and analysis for web-based tsunami computational modeling and visualization

Dylan B. Keon, Cherri M. Pancake, Harry H. Yeh, Dawn J. Wright

For submission to: *Transactions in GIS*

Oregon State University OSU · NACSE · Arctic Region Supercomputing Center · NOAA · DOD HPC Department of Defense High Performance Computing Modernization Program

Project 1: Introduction

Tsunamis are infrequent but can be devastating

US Pacific Coast

Thailand

New Zealand

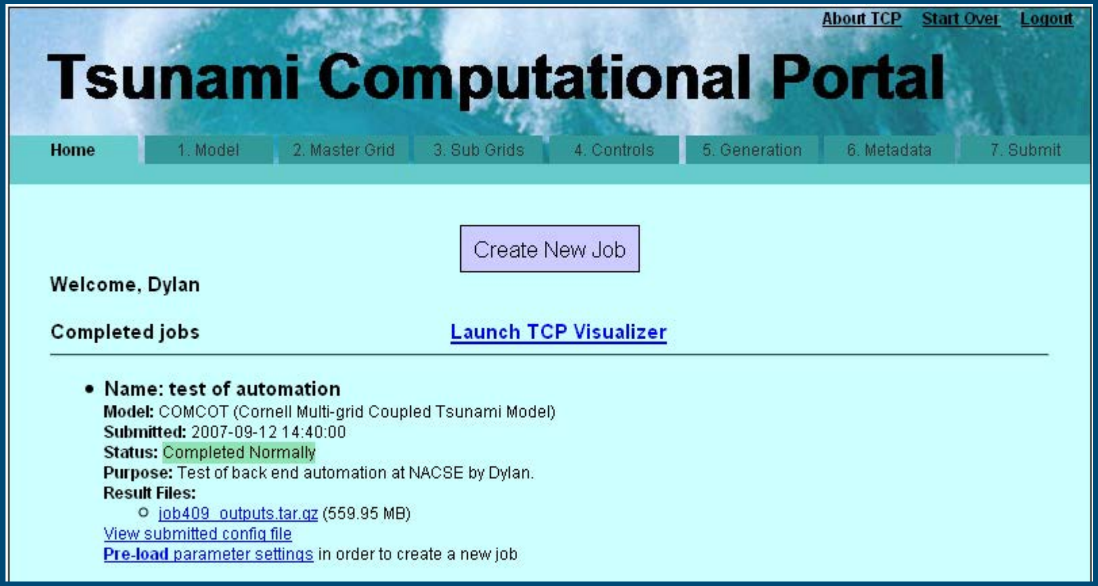
Project 1: Introduction

- Tsunami simulation modeling is important
- Multiple tsunami computational models exist, but they are isolated, difficult to implement elsewhere
- Unique model parameterizations complicate comparisons
- Limited access to "big iron" – supercomputing resources needed to run models efficiently

Project 1: Introduction

Tsunami Computational Portal

- Web-based portal that enables configuration and submission of tsunami simulation runs
- Collaborative effort (OSU, NACSE, ARSC)



Project 1: Introduction

Tsunami Computational Portal

- Three models (COMCOT, UAF, Tsunami CLAW)
- Common parameterization scheme
- Facilitiates identical runs across models, or tweaking of one parameter
- Easy to use

Project 1: Introduction

Data-intensive operations (both input and output)

Multiple data challenges:

- Bathy/topo data storage and representation
- Dynamic preparation of input data for modeling
- Multiple resolutions of input data grids
- Data alignment across grids
- Post-processing of time-series output data

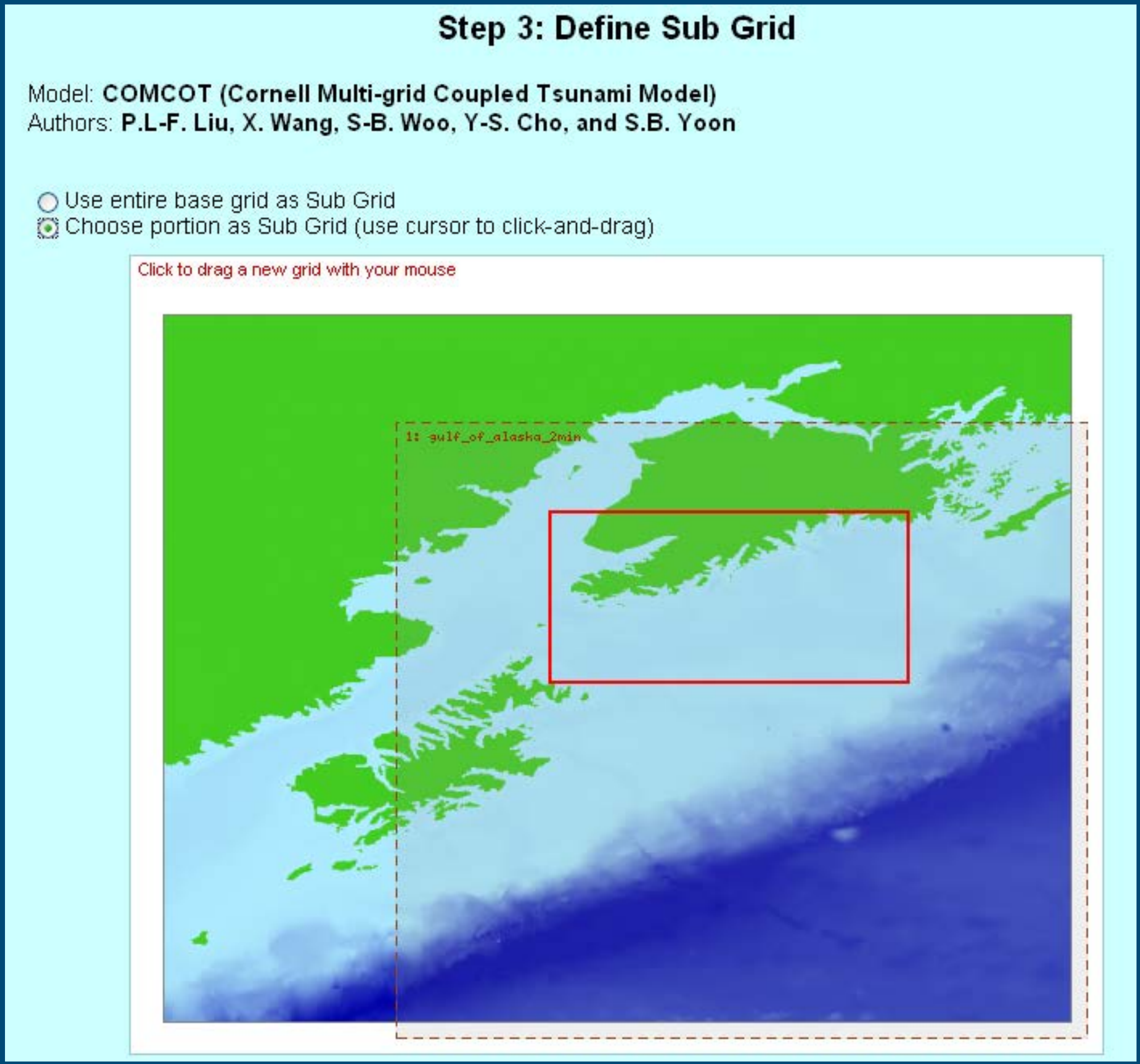
Project 1: Research Goals

- Effective methods for fast spatial data preparation and processing in support of TCP
- Automation of spatial data input and spatiotemporal data output processing tasks
- Spatial database optimization
- Alignment of input grids of multiple resolutions, and validation of alignment in portal

Project 1: Spatial Database

- PostgreSQL / PostGIS on Linux
- Fast db server, SCSI disk array
- Multiple gridded bathy/topo datasets stored as spatial objects
  - Global: ETOPO1, GEBCO_08
  - High-resolution grids contributed by UAF, others
- Largest dataset (GEBCO_08) occupies nearly a billion rows
- Grids processed into spatial objects via Perl

Project 1: Interface



- Grid area definition
- Selected master grid displayed as reference
- Dynamic snapping of subgrid to master grid's point alignment

Project 1: Input Data Processing

```
################################################
# COMCOT input file
################################################

########## Global Information ##########
Job_Id=708
User_Id=keon@nacse.org
Model_Name=COMCOT
Number_of_Time_Steps=87630
Length_of_Time_Step=0.2
Output_Frequency=10
Coef_of_Bottom_Friction=0.0033
Watch_Frequency=100
Initial_Wave_Source=FAULT

########## FAULT Generation ##########
Focal_Depth=5000
Fault_Length=350000
Fault_Width=250000
Dislocation=6.4
Strike_Direction=234
Dip_Angle=5
Slip_Angle=90
Domain_Latitude=53.6166667
Domain_Longitude=-153.6833335
Epicenter_Latitude=56.5
Epicenter_Longitude=-150.933

########## Grid Information ##########
Grid_Number=1
Parent=0
Name=gulf_of_alaska_2min
Output_Name=gulf_of_alaska_2min
Latitude_SW_Corner=53.6166667
Longitude_SW_Corner=-153.6833335
East_West_Points=377
North_South_Points=220
Spacing=120
Parent_Child_Spacing_Ratio=0
X_Start=0
X_End=0
Y_Start=0
Y_End=0
Coordinate_Switch=0
Nonlinear_Switch=0
Friction_Switch=0
Depth_File='gulf_of_alaska_2min.dep'
```

- Dynamically prepare input data upon job submission
- Automated process clips aligned, gridded bathy/topo data
- Custom C++ PostgreSQL functions extract data in binary format required by ARSC
- Job configuration parameters + data submitted to ARSC

Project 1: Database Optimization

- Spatial indexes are important!
- Example (2 deg. longitude x 1 deg. latitude):

QUERY #1 (spatial index disabled)

```
SELECT id, x, y, ST_x(geom) AS lon, ST_y(geom) AS lat
FROM gebco30
WHERE _ST_Within(geom, GeometryFromText('POLYGON((-125 44, -125 45,
                  -123 45, -123 44, -125 44))', 4326));
```

QUERY #2 (spatial index enabled)

```
SELECT id, x, y, ST_x(geom) AS lon, ST_y(geom) AS lat
FROM gebco30
WHERE ST_Within(geom, GeometryFromText('POLYGON((-125 44, -125 45,
                 -123 45, -123 44, -125 44))', 4326));
```

Project 1: Database Optimization

- Spatial indexes are important!
- Example (2 deg. longitude x 1 deg. latitude):

QUERY #1 (spatial index disabled) **Query time: 489180 ms**

```
SELECT id, x, y, ST_x(geom) AS lon, ST_y(geom) AS lat
FROM gebco30
WHERE _ST_Within(geom, GeometryFromText('POLYGON((-125 44, -125 45,
                 -123 45, -123 44, -125 44))', 4326));
```

QUERY #2 (spatial index enabled) **Query time: 342 ms**

```
SELECT id, x, y, ST_x(geom) AS lon, ST_y(geom) AS lat
FROM gebco30
WHERE ST_Within(geom, GeometryFromText('POLYGON((-125 44, -125 45,
                -123 45, -123 44, -125 44))', 4326));
```

Project 1: Output Data Products

- Time series
  - Binary format
  - File contains water depth at each grid point across all simulated time steps
- Sea level max
- U & V velocity vectors
- All output data are useful for further analyses

Project 1: Output Data Processing

Time series visualization

Project 1: Results

- Data optimization + automation + portal system + supercomputers = fast results
- Over 500 modeling jobs completed since inception
- Currently over 120 portal users
- Publications (Marine Geol, Bull Seis Soc Am), presentations (AAG, AGU)
- Useful computational approach for multiple disciplines (climate modeling, ecological modeling)

Project 2

Web-based simulation modeling and visualization of tsunami inundation and potential human response

Dylan B. Keon, Benjamin M. Steinberg, Harry H. Yeh, Cherri M. Pancake, Dawn J. Wright

For submission to: *International Journal of Geographical Information Science*

Project 2: Introduction

Tsunami inundation a concern in coastal communities

- 2011 Tōhoku tsunami
- Potential along U.S. Pacific coast

Tsunami awareness programs

- Evacuation/inundation maps
- Some county plans lacking (Lindell & Prater 2010)

Project 2: Introduction

Simulating potential human response

- Can help researchers, planners understand risk, possible behavior
- Must incorporate informed casualty determination

Simulation frameworks often challenging to build

- Multiple factors, variables to consider
- Can be computationally intensive
- Effective display and representation of output

Project 2: Introduction

User interactivity is important

- Input side: Ability to influence simulation
- Output side: Ability to interactively view results and control visualizations to help interpret data

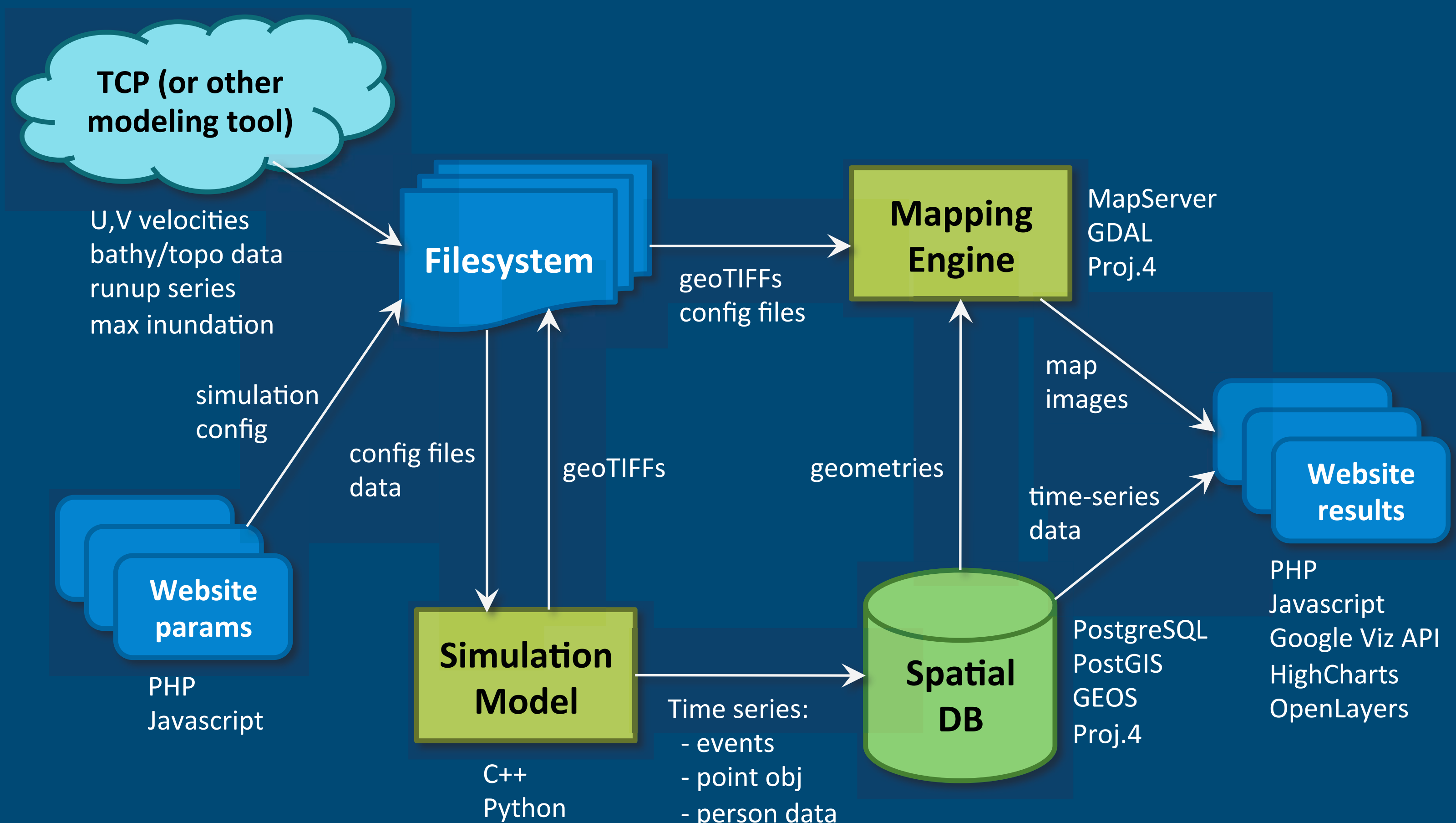
Automation and real-time processing are ideal

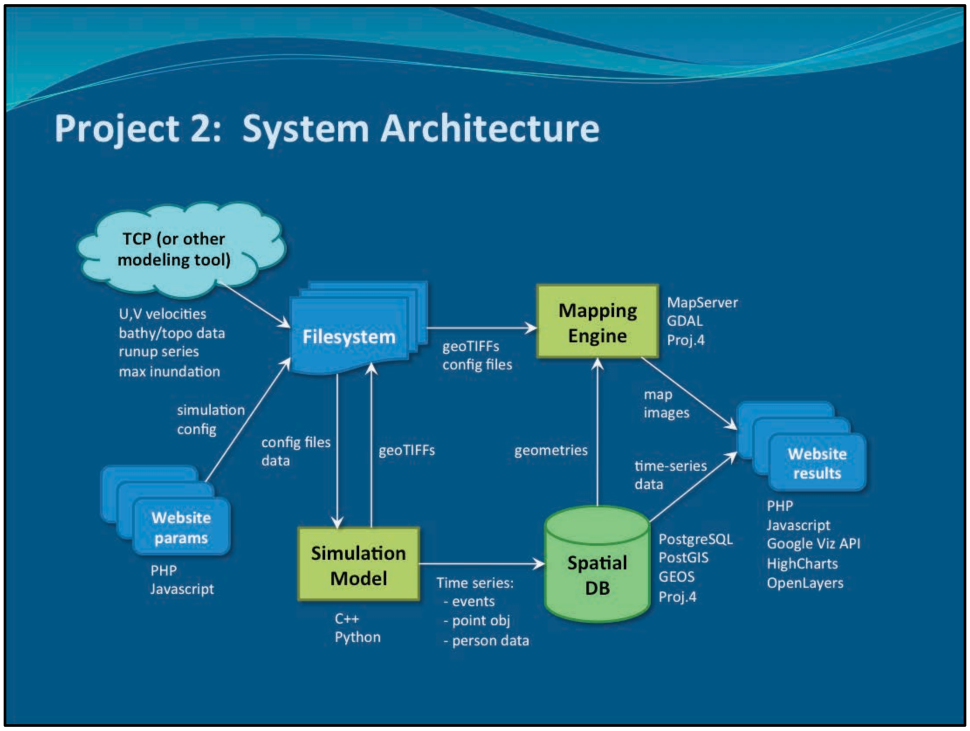
- Define parameters, click a button, get results
- Requires integrated systems

Project 2: Research Goals

- Model potential human response to tsunami events based on user input and a casualty model, and automatically generate simulation output
- Combine tsunami modeling output data with visualization of human response simulation data
- Create a web-based system for defining inputs and exploring output, using open source software
- Support interactive animation and querying within the visualization interface

Project 2: System Architecture





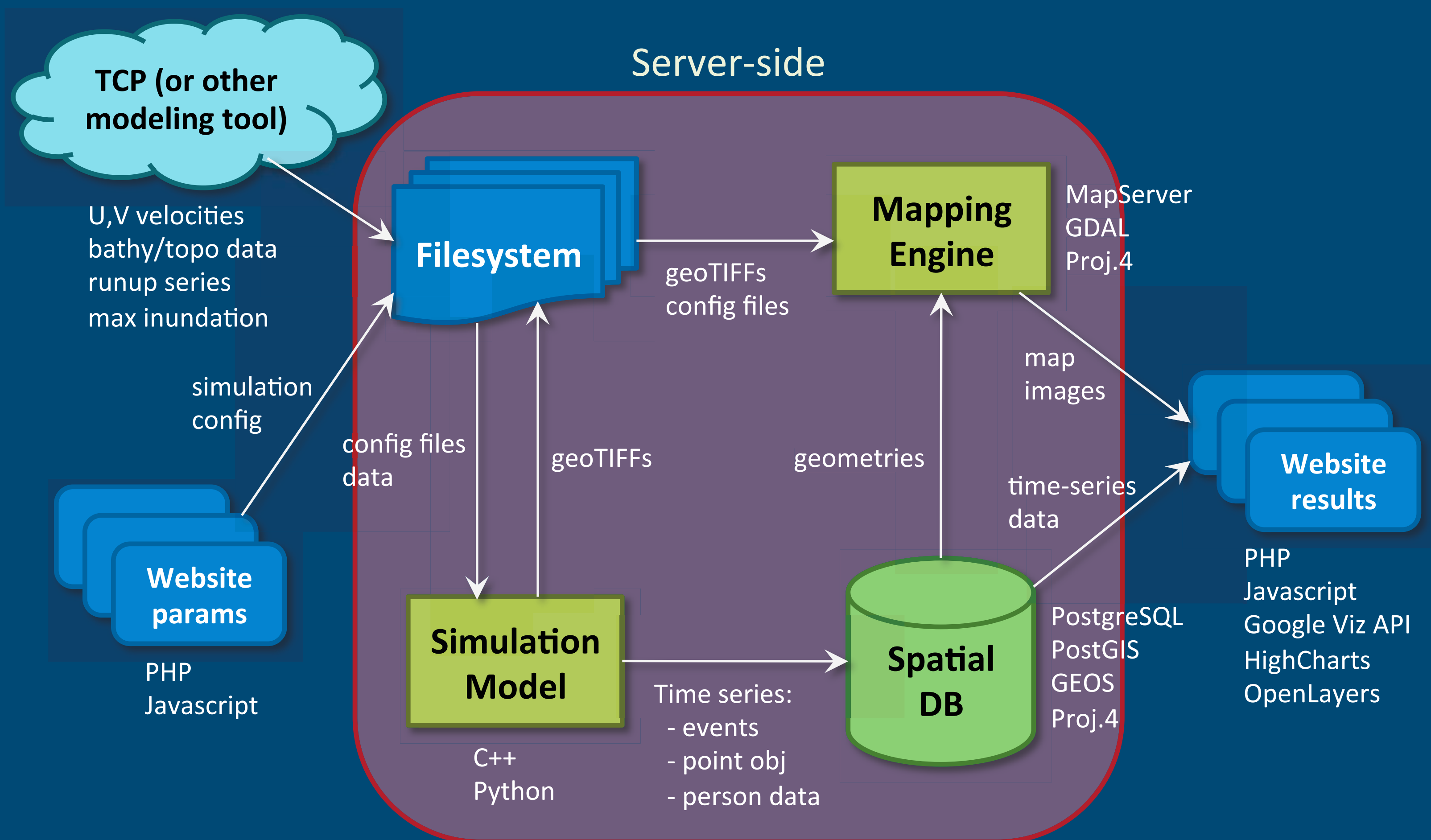
Development of prototype simulation framework

- Tsunami input data (TCP output, other)
- Community input data + seeded pop'l. distrib.
- Server-side simulation engine incorporating casualty model
- Client-side (web-based) interface for defining input parameters and interacting with output
- Utilities/code to "glue" it all together

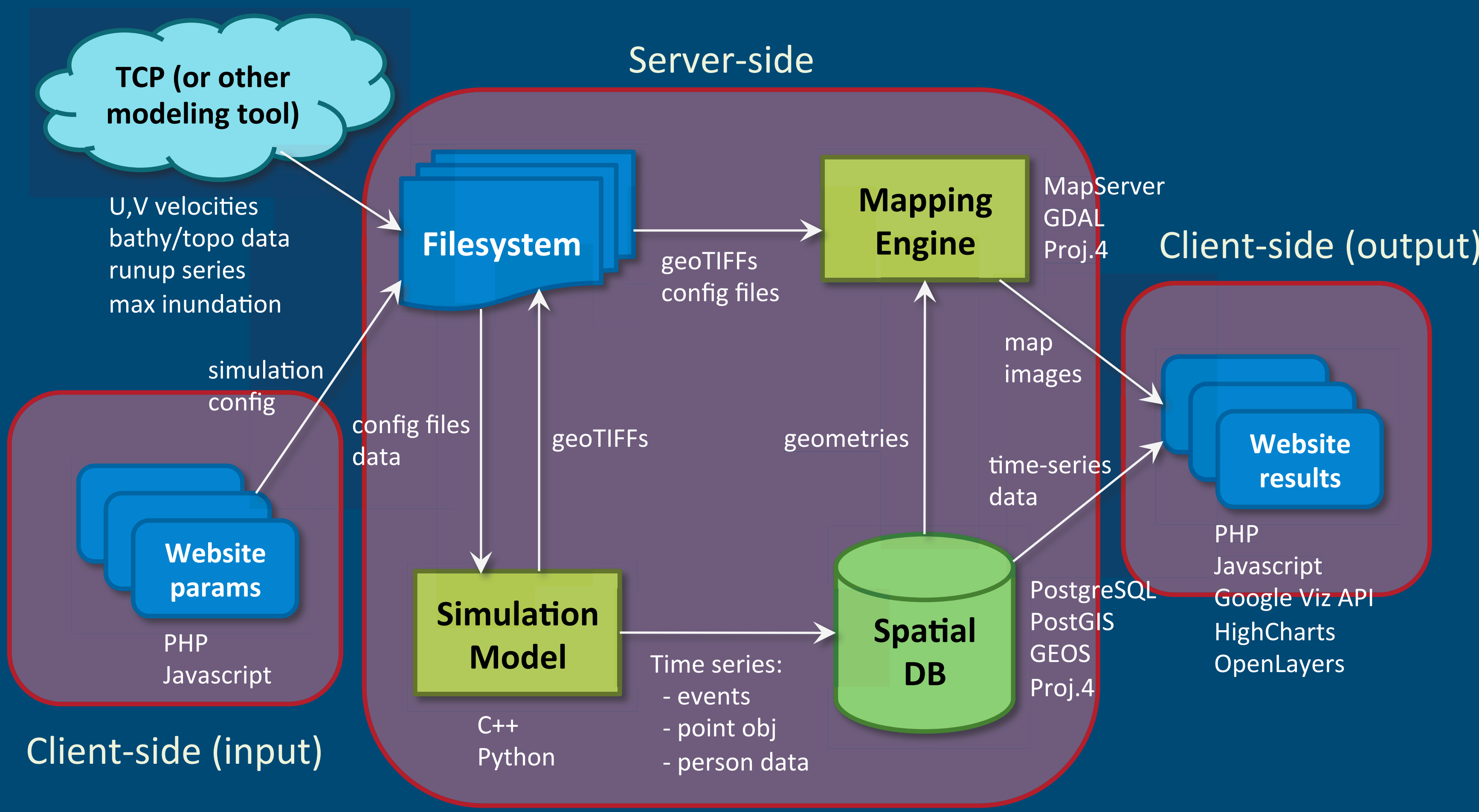
Input data sources

- Tsunami modeling output data (TCP, other)

Project 2: System Architecture

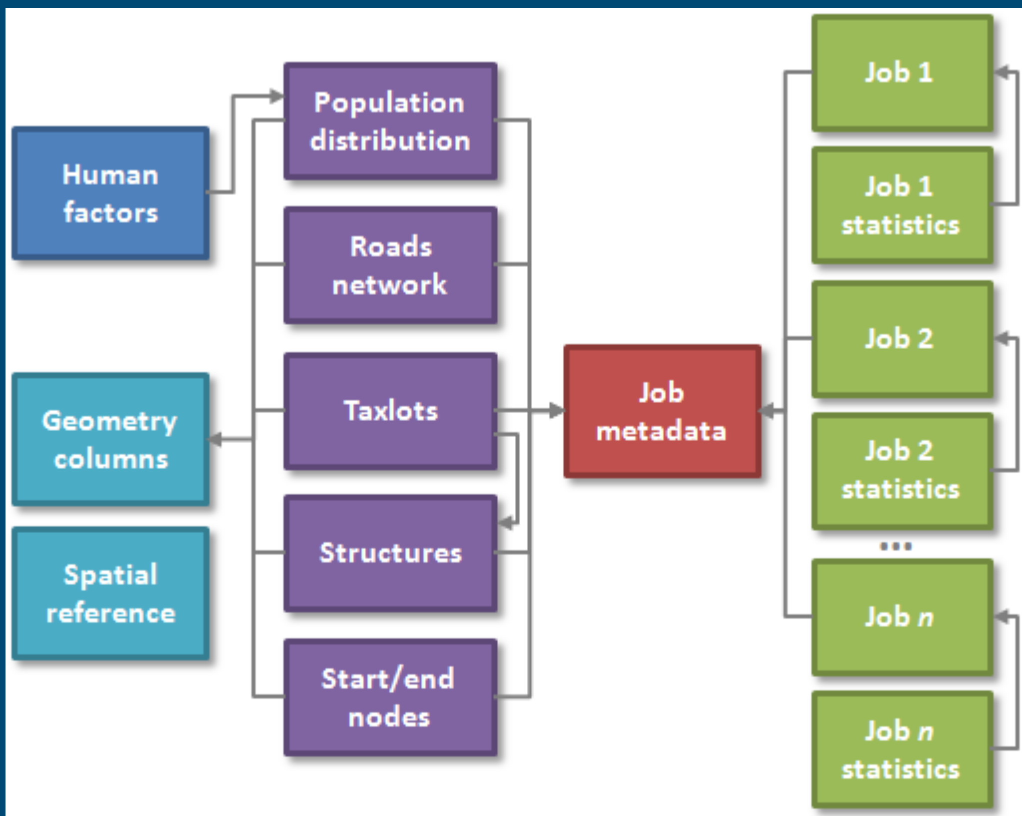


Project 2: System Architecture



Project 2: System Architecture

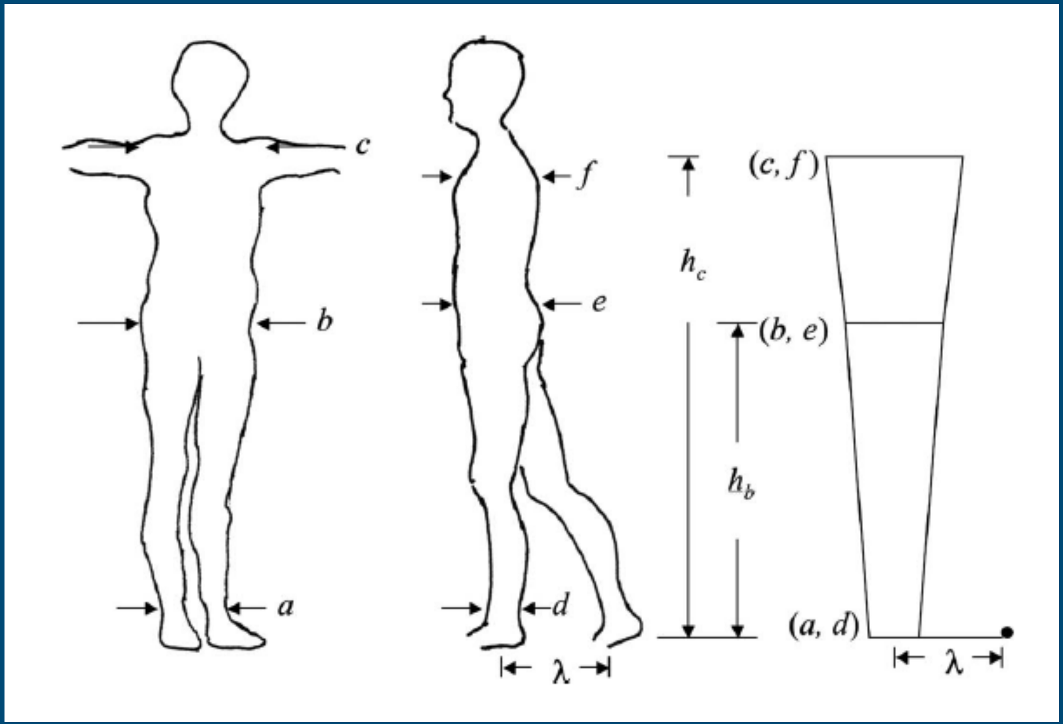
Generalized database schema



- Spatial database
- Time-stamped at record level
- Each person is a database object, and is assigned temporal, spatial, metadata info
- Appropriately indexed

Project 2: Tsunami Casualty Model

- Mean values of anthropometric data
- Generalized male, female body types
- Two potential failure modes (based on force balance, moment balance)



from Yeh (2010)

Project 2: Tsunami Casualty Model

Casualty determination

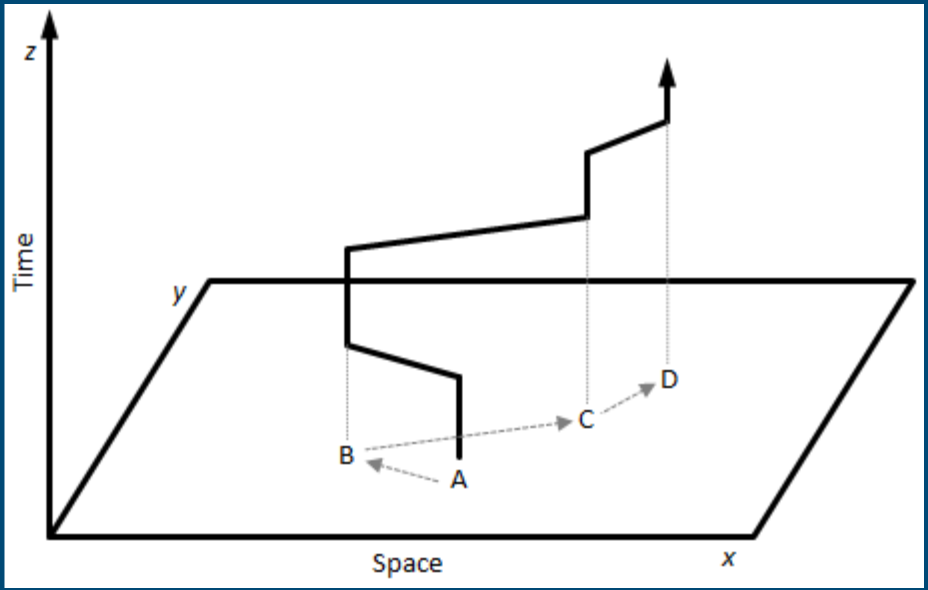
- Simulation model algorithm (C++) determines individual's casualty status at each time step
- Factors considered at each time step:
  - Routed location
  - Water depth in grid cell
  - Velocity and direction of tsunami flow
  - Individual's body type

Project 2: Spatiotemporal Movement

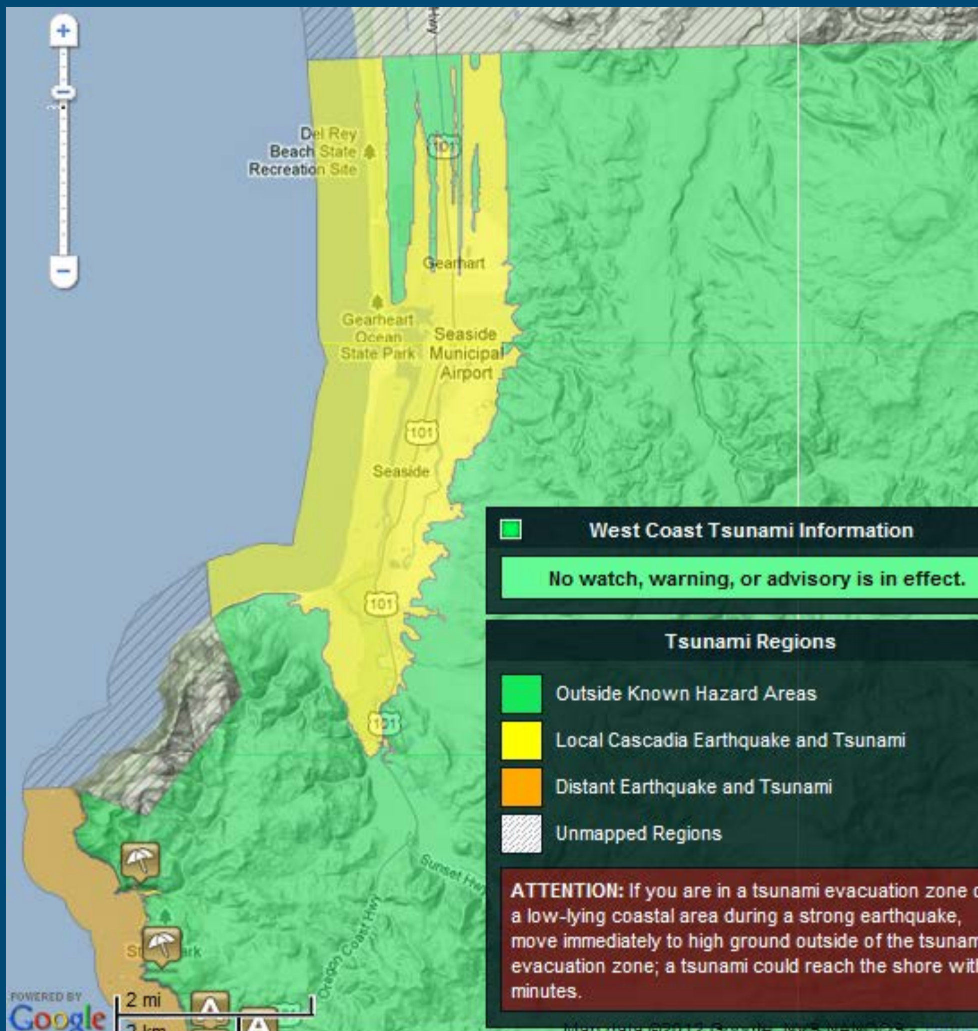
- Simulation model routes pedestrians along existing road network
- Routed toward existing evacuation points
- Routing is done using the Dijkstra shortest-path algorithm
- Simulation model produces spatiotemporal data for each simulated individual across all time steps

Project 2: Spatiotemporal Movement

- All simulation output data stored in spatial database
- Enables visual representation of space-time path



Project 2: Research Area for Prototype



- Seaside, Oregon
- Particularly susceptible to tsunami damage
- Flat for long distance inland
- GIS data

Project 2: Web Interface – Settings Page

Tsunami Inundation and Evacuation Simulation

Simulation Input Parameters

Information

This page is used for configuring tsunami evacuation input parameters for a particular tsunami inundation run. Tsunami inundation input data are produced in a separate interface, the Tsunami Computational Portal (TCP). Output simulation data from the TCP is used as a set of inputs for the evacuation simulation.

Tsunami casualty model parameters were calculated in collaboration with Dr. Harry Yeh. The casualty model is based on the population distribution of a community, including gender and age factors, and determines whether a person can remain standing within a tsunami flow. The method was published in Natural Hazards Review in 2010.

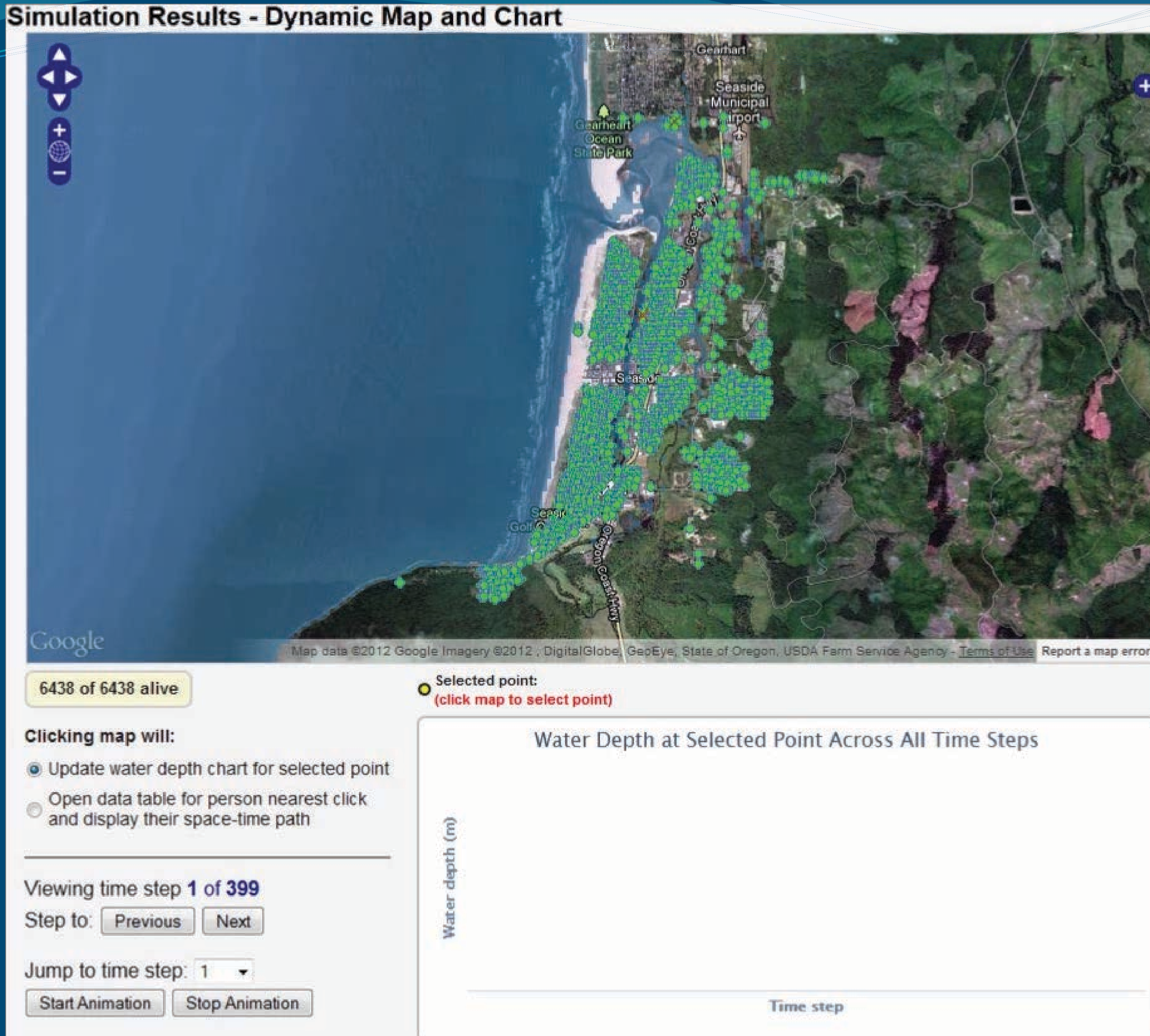
Available Parameters:

- **TCP simulation input:** Existing output data from a TCP simulation run
- **Population size:** Number of people to be simulated
- **Time of day:** General time of day (affects population distribution)
- **Evacuation start time step:** The time step at which to begin the evacuation simulation
- **Evacuation point(s):** Defined evacuation points that exist in the community of interest, or arbitrary user-defined points
- **Alternate road network:** An alternate shapefile of the road network can be provided (contact us for more information)

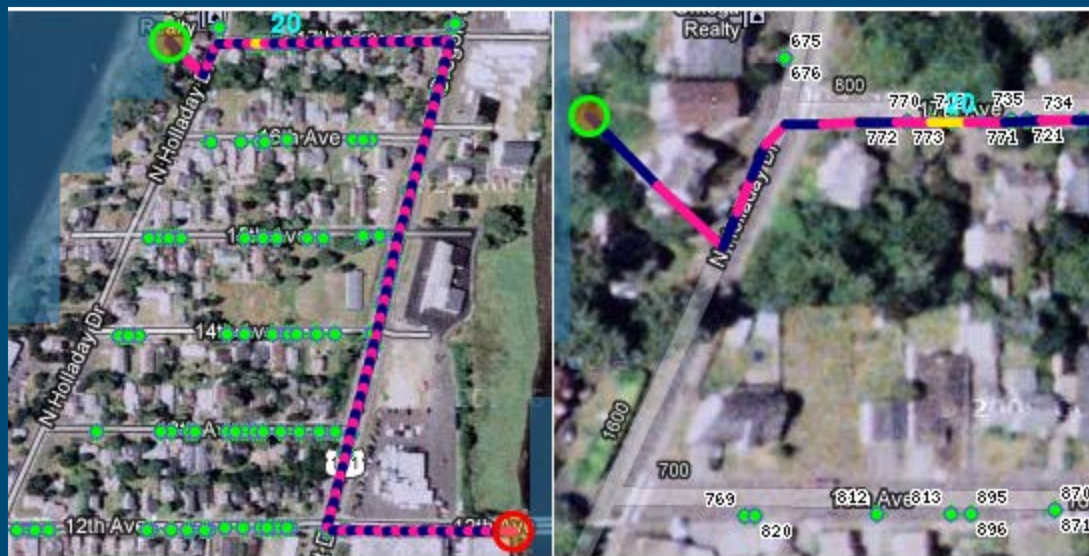
Input Parameters

TCP Simulation Input:	Seaside, OR - 10m *time step = 18s*
Simulation extent:	*determined by TCP Input*
U and V velocity vectors:	*determined by TCP Input (if UAF Tsunami model was used)*
Population size:	
Time of day:	Morning
Evacuation start time step:	(1-400)
Evacuation point(s):	○ Use existing evac point(s) for currently selected location ○ Define your own evac point(s) (up to 3) Lon: Lat: Lon: Lat: Lon: Lat:
Alternate road network:	*Contact us to use an alternate road network*
Bathymetry & topography:	*determined by TCP Input*
Your email address:	

Submit Parameters


Simulation Results - Dynamic Map and Chart
Gearhart
Seaside Municipal Airport
Gearhart Ocean State Park
Oregon Coast Hwy
Google
Map data ©2012 Google Imagery ©2012 , DigitalGlobe, GeoEye, State of Oregon, USDA Farm Service Agency - Terms of Use
Report a map error
6438 of 6438 alive
Selected point:
(click map to select point)
Clicking map will:
Update water depth chart for selected point
Open data table for person nearest click and display their space-time path
Viewing time step 1 of 399
Step to: Previous Next
Jump to time step: 1
Start Animation Stop Animation
Water Depth at Selected Point Across All Time Steps
Water depth (m)
Time step

Project 2: Simulation Results Interface



- Database query via OGR, GeoJSON return
- OpenLayers.Layer.PointTrack method used
- Wrapped in custom JavaScript code

Project 2: Discussion / Limitations

- Data – structures, roads, etc. changing
- Structure type – possible misattribution
- Resolution of tsunami simulation data
- Generalized body types – expansion of ages and associated running speeds, etc.
- Evac start times based on multiple warning types
- Introduction of random variation in start times and in body type parameters

Project 2: Conclusion

- Casualty simulation is an important method for estimating tsunami risk to coastal communities
- This prototype demonstrates a simulation framework that incorporates a published casualty determination method
- Web browser as a platform supports complex tools
- Possible to integrate animated representation of spatiotemporal data with interactive data viz/query interface

Project 3

GridStats: A web-based system for calculating climate grid statistics over varying spatial and temporal scales

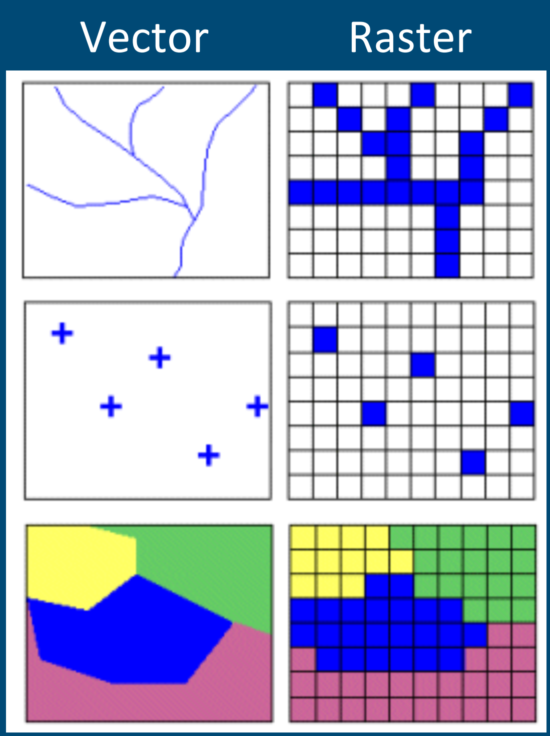
Dylan B. Keon, Christopher Daly, Adam Ryan

For submission to: *Computers & Geosciences*

Project 3: Introduction

Raster-based spatial datasets (grids)

- Simple yet powerful data structure
- Facilitates calculations across sets of aligned grids
- Each grid can represent a temporal snapshot
- Enables analysis of change over time



Project 3: Introduction

Raster-based datasets (grids)

- Grids occupy large file sizes
- Performing calculations across large sets of grids is computationally intensive
- Typically requires specialized desktop GIS software
- Web-based mapping and services have grown significantly, but web-based raster analysis capabilities remain limited

Project 3: Introduction

PRISM climate grids

- PRISM: Statistical-topographic model
- Weather station observations, elevation
- Modeled climate grids (precip, temp) are produced for the continental US (48 states)
- Monthly, annual time scales at 4 km since 1995
- Now producing data at 800 m, and on a daily time scale

Project 3: Introduction

PRISM climate grids

- Converted from ASCII GRID to BIL last year
  - Generalized binary raster data format
  - Well-supported in GDAL processing tools, as well as in ArcGIS, ENVI, Imagine
  - Fast programmatic read/write via Python
- Currently > 49,000 grids
- 7025 x 3105 cells (21.8 million) per grid

Project 3: Introduction

Data challenges

- Daily grids for each climate parameter (ppt, tmin, tmax, tmean) at 800 m = a lot of data
- Often need to make a quick assessment or perform a statistical operation over space and time (multiple grids)
- Can involve loading large sets of grids across network, manually clipping, etc.
- Need flexible way to process large sets of grids

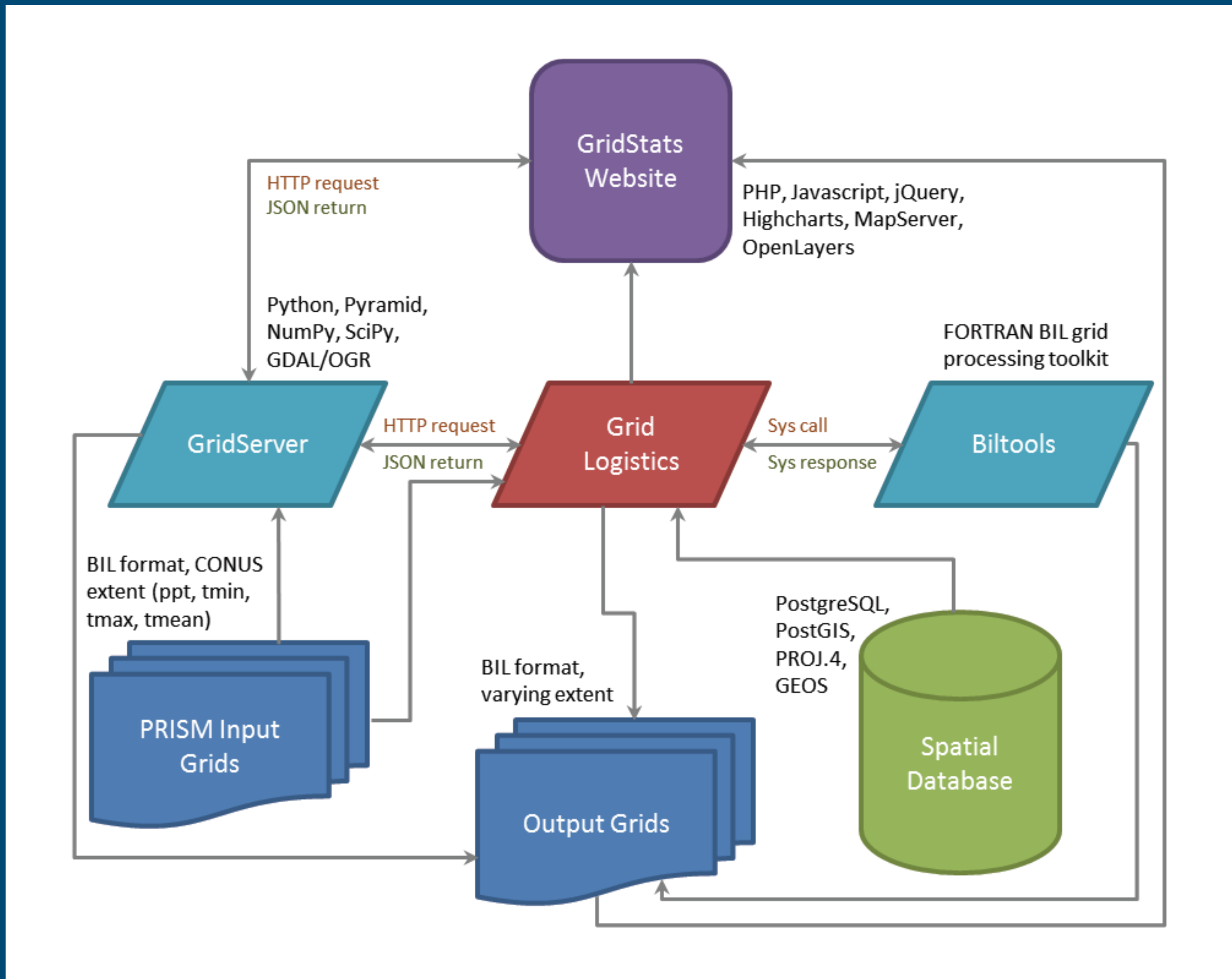
Project 3: Project Goals

- Enable web-based statistical analyses across multiple grids over user-defined spatial and temporal scales
- Devise optimization techniques to enhance server-side grid processing speed
- Design user interactivity features and aids to help properly define input parameters, and interpret the output data
- Develop interactive map- and chart-based tools for visualizing and exploring output data

Project 3: The GridStats System

- Quickly perform statistical operations across grids
- Varying spatial extent (anywhere in CONUS)
  - Point (single grid cell)
  - Area (bounding box, defined polygon boundary, or entire grid)
- Varying temporal extent
  - Any range within daily, monthly, annual time series (1895 to present for monthly/annual)
  - Also one month across range (all Januaries)

Project 3: GridStats System Architecture



Project 3: The GridServer

- Server-side grid processing system
- Fast processing of multiple grids, concurrent operations
- Python-based software, wrapped in a multi-threaded application (WSGI-based Pyramid, Paster)
- Runs on a dedicated server with local copy of grids
- Maintains file handles to all grids
- Enables real-time grid processing, gives JSON return + grid(s) as output

Project 3: The GridStats System

Statistics currently supported in GridStats

Statistic	Server-side Tool and Request Type		
	GridServer (point method)	**GridServer (area method)**	**Biltools (entire grid)**
Sum	X		
Range	X		
Min	X	X	X
Max	X	X	X
Mean	X	X	X
Median	X		
Sample Standard Deviation	X	X	X
Standard Error	X		
Principal Components Analysis (PCA)		X	

Parameter

- (•) Precip (ppt)
- ( ) Min Temp (tmin)
- ( ) Max Temp (tmax)
- ( ) Mean Temp (tmean)

Temporal Scale

- (•) All years in range: 1981 — 2010
- ( ) One month across range of years: - Month - — - Start Yr - to - End Yr -
- ( ) All months in range: 2000 Jan to 2009 Dec
- ( ) All days in range: 2009 May 1 to 2010 Apr 30

Spatial Scale (matching cell range is extracted from each grid for analysis)

- (•) Point (one grid cell) Lon: -80.71777 Lat: 25.66349 (or click map to set point)
- ( ) Area
  - ( ) Draw box on map to define area (hold Shift key and click-drag mouse) [limited to 300,000 sq km]
  - ( ) State: - State -
  - ( ) County: - State - — - County -
  - ( ) Watershed (8-digit HUC): - State - — - HUC Name - — - HUC Code -
- ( ) Entire grid extent (48 states)

Statistic of Interest

Available via Point selection

- (•) Summary Stats

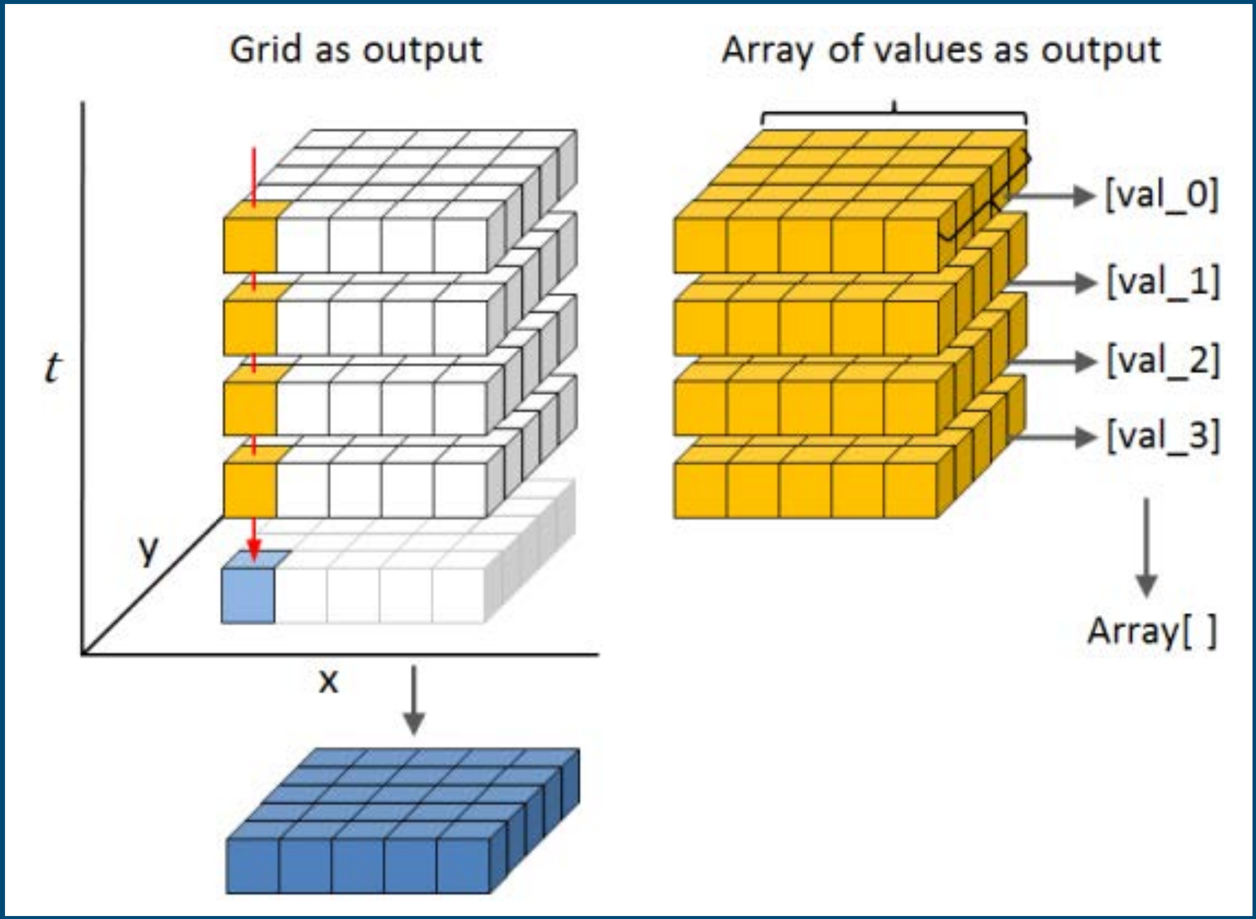
Available via Area selection

- ( ) Mean
- ( ) Median
- ( ) Min
- ( ) Max
- ( ) Standard Deviation
- ( ) PCA

Submit

Project 3: The GridStats System

Area methods: "Temporal first" vs. "Spatial first"



Project 3: The GridStats System

Python: NumPy module

- Numerical processing
- Support for multi-dimensional arrays, plus functions to operate on them
- Very fast processing speed
- Python + NumPy + GDAL = fast, flexible programmatic processing and analysis of raster data

Project 3: The GridStats System

"Extended" map algebra (Frank 2005)

"Multidimensional" map algebra (Mennis 2010)

- Calculations across spatiotemporal raster data
- Also works with NumPy ndarrays…
  - "Temporal first" and "Spatial first" operations
  - Calculations performed across grids loaded into in n-dimensional arrays

Project 3: Results

Point method: Performance
(ppt, daily, summary stats)

Requested date range (daily ppt data)	Number of grids in request	Initial response time (s)	Average subsequent response time (10 at same location) (s)
1980-01-01 – 1980-01-02	2	0.08	0.06
1980-02-01 – 1980-02-10	10	0.13	0.06
1980-03-01 – 1980-06-08	100	0.19	0.08
1980-07-01 – 1983-03-27	1,000	0.97	0.21
1983-04-01 – 1988-09-20	2,000	1.50	0.38
1988-10-01 – 2002-06-09	5,000	3.84	0.77
1980-01-01 – 2007-05-18	10,000	8.12	1.46

Project 3: Results

Area method: Performance (mean ppt, monthly)

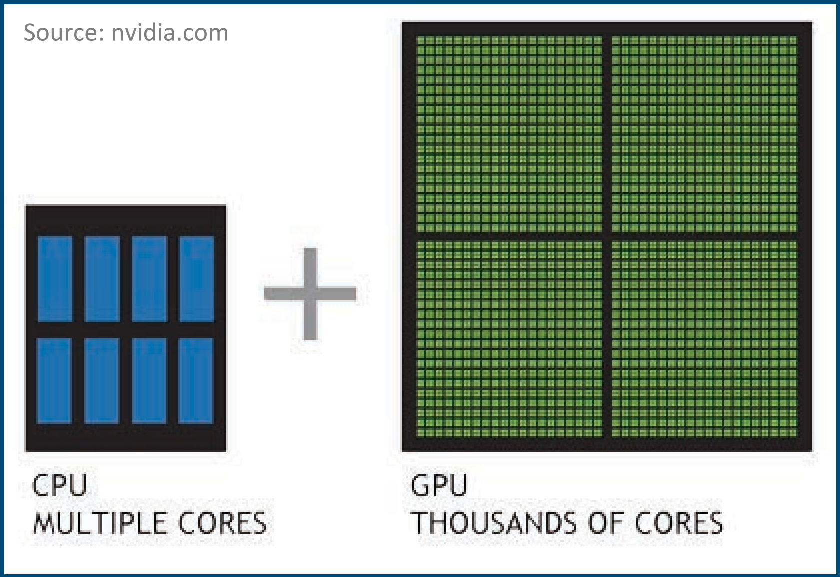
Requested date range (monthly ppt data)	Number of grids in request	Spatial extent	Number of cells processed per grid	Initial response time (s)	Average subsequent response time (10 at same location) (s)
Jan 1920 – Feb 1920	2	1-degree box (-120,43 to -119,44)	14,641	0.16	0.10
Jan 1922 – Oct 1922	10	1-degree box (-81,37 to -80,38)	14,641	0.46	0.19
Jan 1930 – Apr 1938	100	1-degree box (-94,41 to -93,42)	14,641	2.22	1.24
Jan 1940 – Apr 1948	100	3-degree box (-100,40 to -97,43)	130,321	6.78	5.72
Jan 1970 – Dec 1971	24	Wallowa County, Oregon	20,384	1.13	0.43
Jan 1990 – Dec 1992	36	Pennsylvania	214,900	2.97	2.69

Project 3: Results

- Already used internally as a data exploration tool, and for producing/exporting statistical output grids
- Core functionality will be used in production PRISM Climate Group projects
  - Area-based methods
  - Statistical summaries

Project 3: Discussion

- Effective server-side approach for fast automated statistical processing of large grid sets
- Could be even faster with parallel processing and/or GPU computing…
- Possible optimizations on software side (e.g., nginx) and hardware side (multiple servers, load balancer)



Project 3: Discussion

- Huge flexibility – perform statistical calculations over large (or small) spatial, temporal scales
- Server processes automatically make newest grids available to GridStats
- Web browser works well as a platform for configuring jobs and viewing/exploring results
- Client-side web tools offer great flexibility

Conclusion

- Multidimensional spatiotemporal data challenging to process, visualize – require specialized techniques
- Generalized methods possible to an extent
- Web browser is a viable platform for complex applications, and highly accessible
    - Rich internet applications will continue to expand in scope and complexity, and in possibilities
    - AJAX a key element of RIA

Conclusion

- Temporal snapshot data representation, time-stamping at record level both effective
- Open source software completely feasible for building complex RIA
  - Open source GIS utilities such as GDAL/OGR excel for fast server-side processing operations
- Automation is key to fast server-side processing and analysis of spatiotemporal data for web applications

Acknowledgments

Dawn Wright (CEOAS)

Anne Nolin (CEOAS)
Chris Daly (CBEE, PRISM)
Mike Bailey (EECS)
Rick Colwell (CEOAS)

Cherri Pancake (EECS, NACSE)
Harry Yeh (CCE)
Ben Steinberg (NACSE)
Adam Ryan (NACSE)

Colleagues at NACSE and the PRISM Climate Group

CEOAS

Collaborators at ARSC

Funding agencies:

DOD HPCMP
NOAA/NWS
NSF
NWACC
USDA

Family and friends

Karen, Julia, and Anna