O'SOAP - A Web Services Framework for DDDAS Applications

Keshav Pingali and Paul Stodghill
Department of Computer Science
Cornell University
Ithaca, NY 14853, USA

Funding by NSF grant ACIR-0085969

CORNELL

ISS

Outline

- The Adaptive Software Project
- Infrastructure requirements
- Introduce O'SOAP
  - Features
  - Example ASP application
  - Performance
- Summary and future work

CORNELL

ISS

The Adaptive Software Project (ASP)

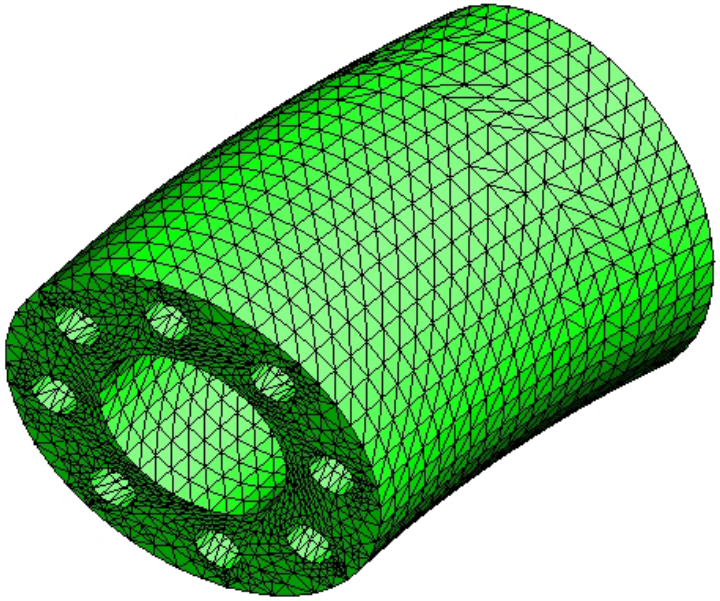
- Multi-institutional
  - Cornell
  - Mississippi State University
  - University of Alabama at Birmingham
  - College of William and Mary
  - Ohio State University
  - Smith College
  - Clark Atlanta University
- Multi-disciplinary
  - Physics
  - Civil Eng.
  - Fluid Mech.
  - Fracture Mech.
  - Bio. Mech.
  - Computer Science
    - Numerical Analysis
    - Comp Geo.
    - Graphics
    - Compilers
    - Distributed Systems

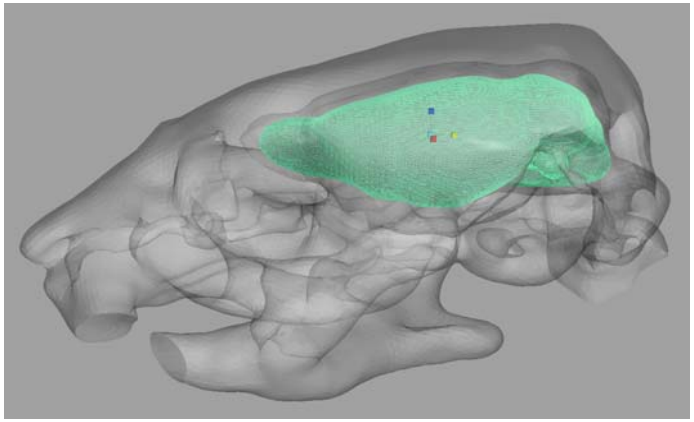
http://www.asp.cornell.edu/

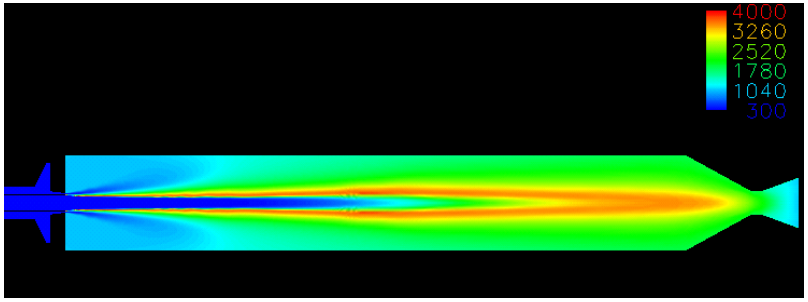
CORNELL

ISS

ASP is Multi-disciplinary



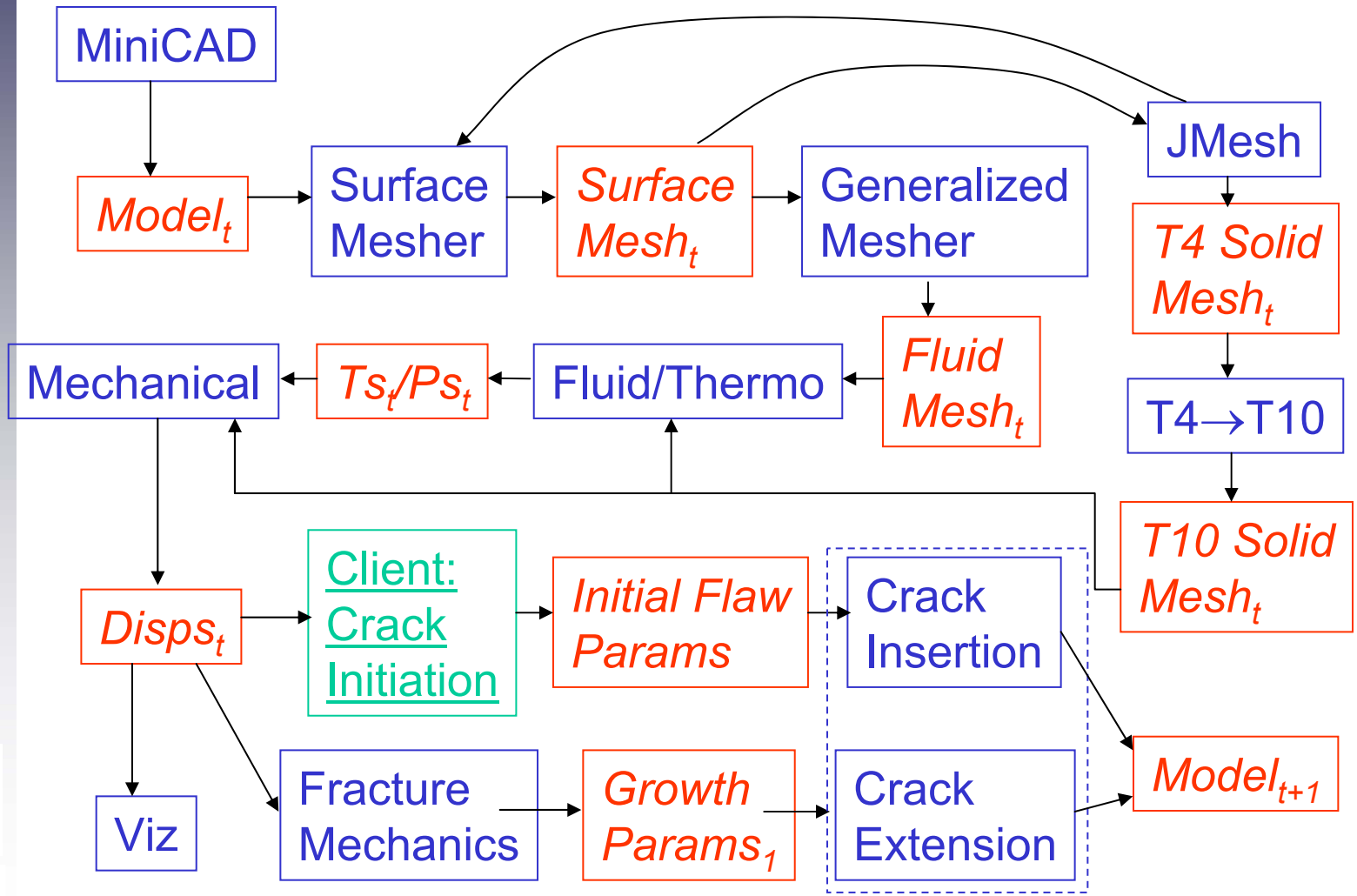




CORNELL

ISS

Sample ASP Workflow



CORNELL

ISS

A Lesson for DDDAS

- ASP is necessarily a distributed simulation system
  - Multi-institutional, multi-disciplinary, multi-platform
- Other examples: geographically distributed instruments
  - VLA radio telescopes
  - Sensor nets

- ***Adaptive and DDDAS applications can be (usually are?) distributed systems***

CORNELL

ISS

Building distributed systems is hard!

- Do you need to be a computer scientist as well as a computational scientist in order to build a DDDAS application?
- Computer scientists need to make distributed systems simpler in order to enable computational scientists to build DDDAS's.
- Needed: Infrastructure for Distributed DDDAS that will lower the entry point for application programmers.

CORNELL

ISS

Requirements of a DDDAS Infrastructure

- Open, Extensible
  - Uses standard protocols
- Suitable for distributed DDDAS's
  - Supports loosely coupled distributed component
- Simple to use
  - Able to incorporate existing components
  - General enough to handle different domains
- High performance
  - Large data-sets
  - Low overhead
  - Scalable performance

CORNELL

ISS

O'SOAP – An Infrastructure for Adaptive and DDDAS Applications

- Web-Services based
- Developed for use in the ASP Project
- Implemented using O'Caml
- Design goals:
  - Make it simple to use
  - Make it fast
- Implements core Web Services protocols
  - eXtensible Markup Language (XML)
  - Simple Object Access Protocol (SOAP)
  - Web Service Definition Language (WSDL)

CORNELL

ISS

Outline

- The Adaptive Software Project
- Infrastructure requirements
- Introduce O'SOAP
  - Features
  - Example ASP application
  - Performance
- Summary and future work

CORNELL

ISS

O'Soap Clients

- osoap_tool: stand-alone command line interface

```
$ osoap_tool \
    http://somewhere.com/test.cgi?WSDL \
    call x=2 y=3 out:output.txt
```

- wsdl_tool: WSDL → O'Caml stub generator
  - WSDL → C/C++ in development
  - WSDL → Perl, Java, Python, … using SWIG

CORNELL

ISS

O'Soap Servers

- Legacy command line applications into Web Services
  - Servers implemented as CGI scripts
  - "oids_server": sample command line → SOAP server

- sample.cgi,

```
#! /bin/bash

oids_server \
    -n arithmetic-test \
    -N "Test Server"
    -U urn:test -- \
    ./add.sh \
        '[in x:int]' \
        '[in y : int]' '>' \
        '[out file result:int]'
```

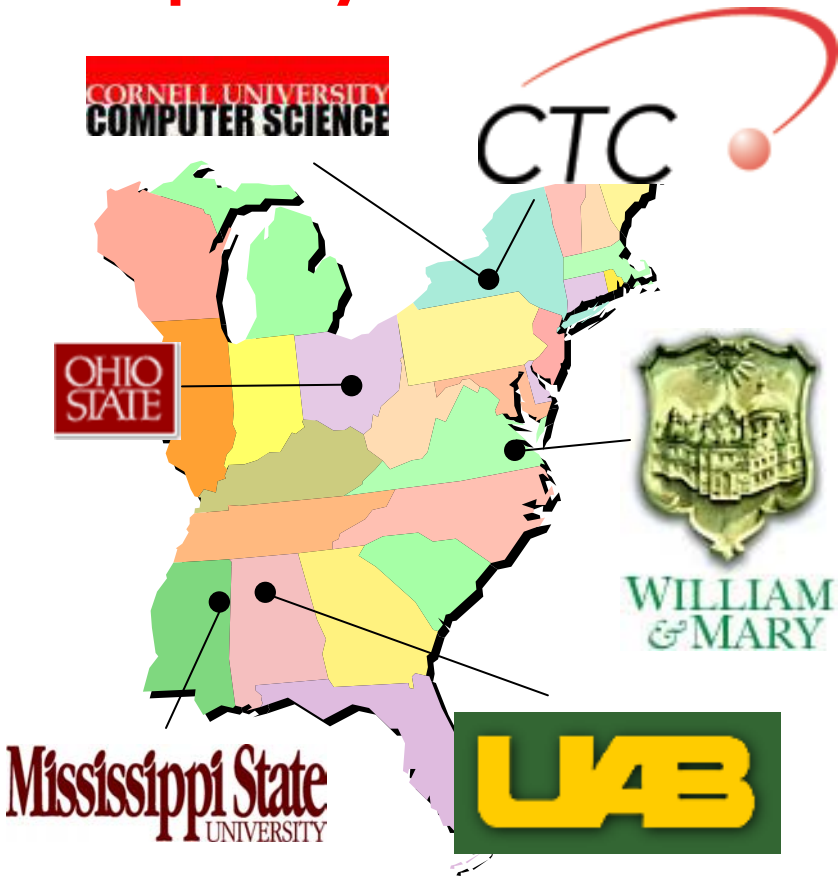
- http://host.com/sample.cgi?wsdl

CORNELL

ISS

Web Services Deployment

Component	CUCS	CTC	MSU	CW&M
Surface Mesher	***Yes***			Yes
JMesh	***Yes***			Yes
T4→T10	***Yes***			
Generalized Meshers	Yes		***Yes***	
Fluid/Thermal Solver	***Yes***		Yes	
Mechanical Solver		***Yes***		
Crack Insertion	***Yes***			
Fracture Mechanics	***Yes***			
Crack Growth	***Yes***			



Yes = component exists as Web Service
Yes = denotes components used for experiments

CORNELL

ISS

General enough to handle different domains

- Aerospace
  - Hypersonic, Chemically Reacting, Fluid Flow
  - Thermal Dynamics
  - Structural and Fracture Dynamics
- Medicine
  - Traumatic Brain Injury (TBI)
- Program Transformation System
  - C^3 Compiler: Automatic, Application-level Checkpointing for Sequential, MPI, and OpenMP programs

CORNELL

ISS

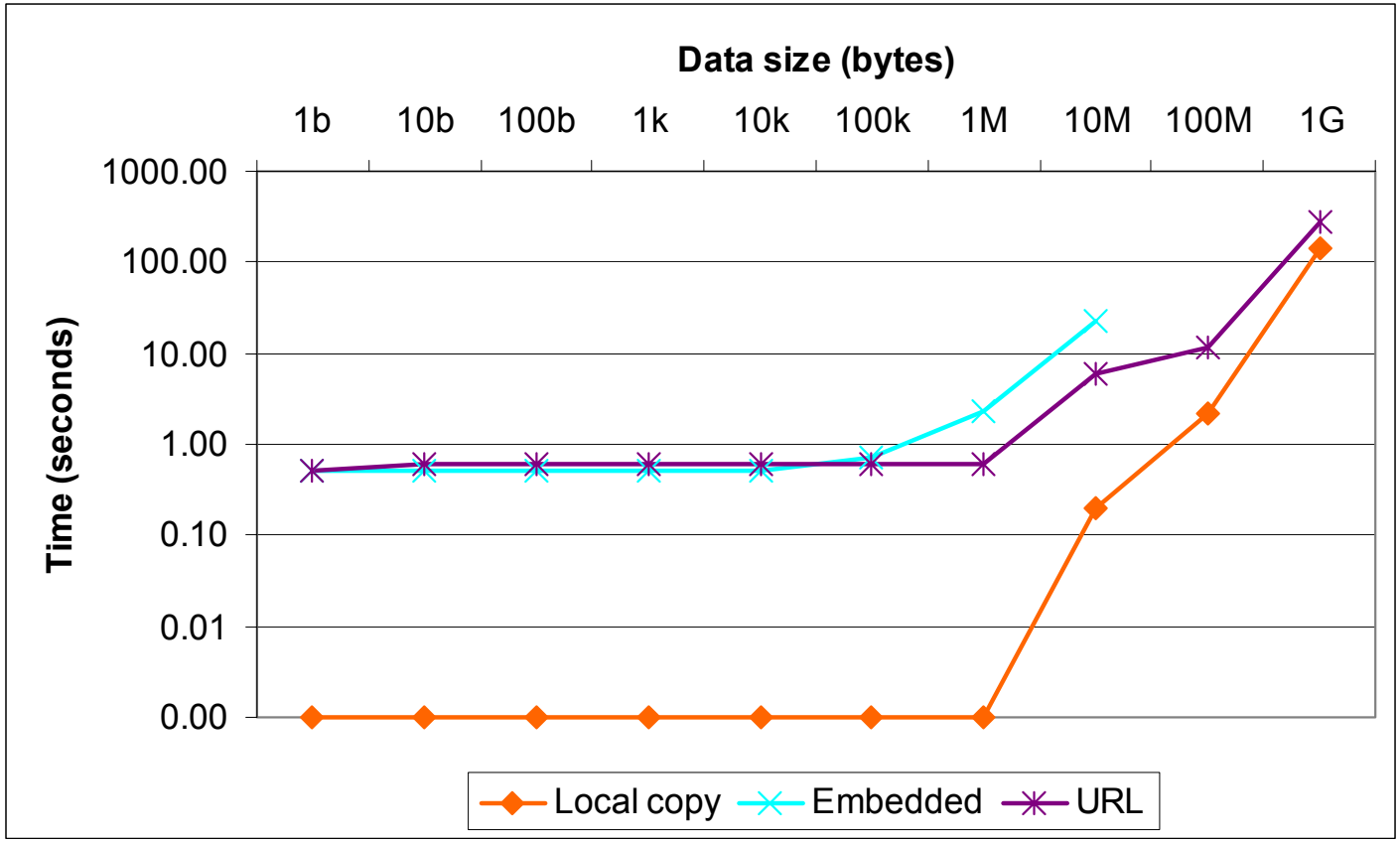
Handling Large Data-sets

- Classic SOAP usage
  - Data is embedded in messages
  - Does not scale
  - Overhead for processing XML
- O'SOAP
  - URL to data can be put in message
  - No additional overhead for processing
  - Data-sets can be let on server between component invocations
  - Data-sets can be transferred server-to-server
  - Automatic and transparent to the application developer

CORNELL

ISS

Ping-pong test



- Client and server both on CUCS
- Randomly generated binary files

CORNELL

ISS

Overheads and Scalability

Problem Size	Solid Mesh			Interior Mesh		
	Vertices	Triangles	tet's	Vertices	tri's/ quad's	tet's/ prisms
1	4,835	4,979	22,045	19,242	3,065	38,220
2	16,832	10,322	83,609	41,216	5,232	85,183
3	54,849	21,127	289,500	79,407	9,074	170,179

Problem Size	Local	CU Client		UAB Client	
	Runtime (secs.)	Runtime (secs.)	Overhead	Runtime (secs.)	Overhead
1	1630.89	1719.44	5.43%	1695.24	3.95%
2	5593.66	5776.55	3.27%	5745.78	2.72%
3	22202.73	22901.39	3.15%	22222.49	0.09%

CORNELL

ISS

Summary

- O'SOAP meets the DDDAS requirements
  - Uses standard protocols
  - Supports loosely coupled distributed component
  - Able to incorporate existing components
  - General enough to handle different domains
  - Large data-sets
  - Low overhead
  - Scalable performance
- Future work
  - Authentication, Authorization, Accouninting (WS-Security)
  - Interoperability with Grid systems (WSRF)
  - Numerous performance improvements

CORNELL

ISS

Thank you

O'SOAP's Home Page:
http://www.asp.cornell.edu/osoap/

My email: stodghil@cs.cornell.edu

CORNELL

ISS