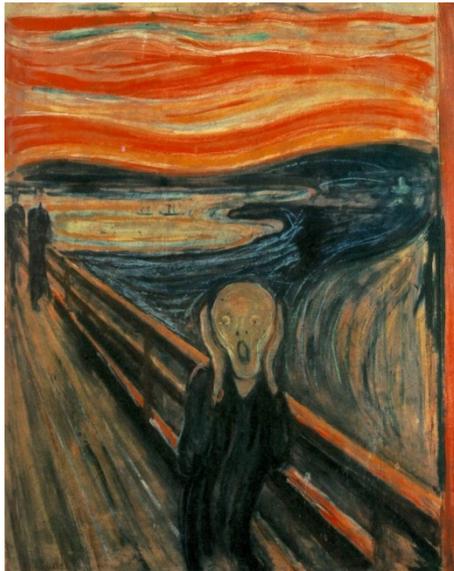


Working with Tony
is everything
it's cracked up to be



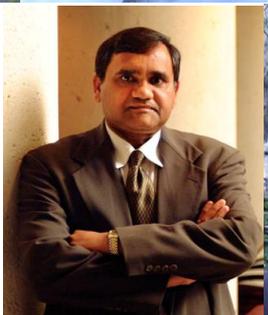
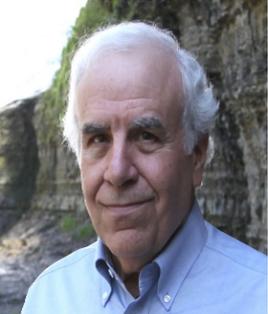
Keshav Pingali
University of Texas, Austin

How I met Tony (circa 1997)



Adaptive Software Project (ASP)

- NSF-funded large ITR project
 - Adaptive Software for Field-driven Simulation
 - 2000-2005
 - \$2 million/yr
- Project partners:
 - Cornell University
 - Mississippi State University
 - University of Alabama, Birmingham
 - College of William and Mary
 - Ohio State University
 - Clark-Atlanta University





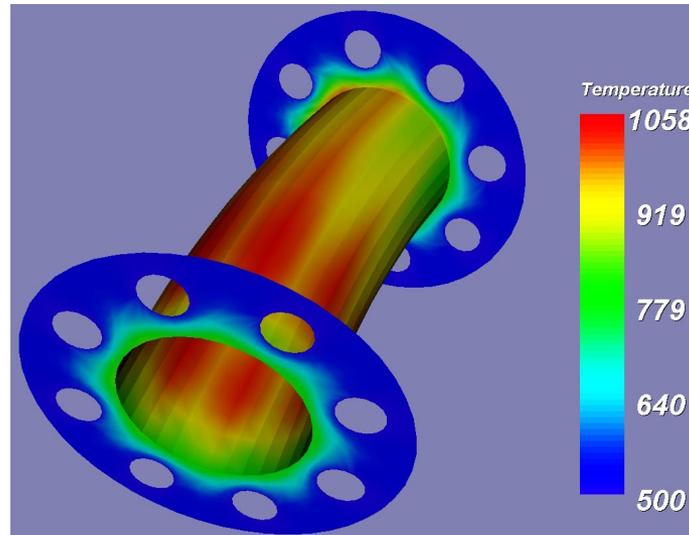
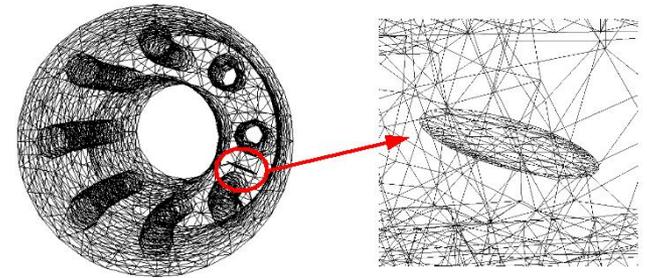
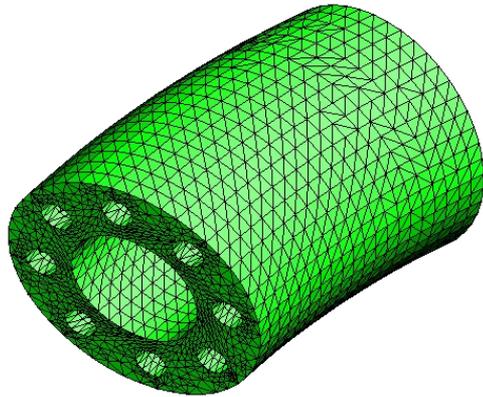
Goal of ASP

- **Implement a**
 - geographically distributed system for multi-physics multi-scale adaptive CSE simulations
 - computational fracture mechanics
 - chemically-reacting flow simulation
- **Understand principles of implementing**
 - high-performance distributed computing systems
 - adaptive software systems

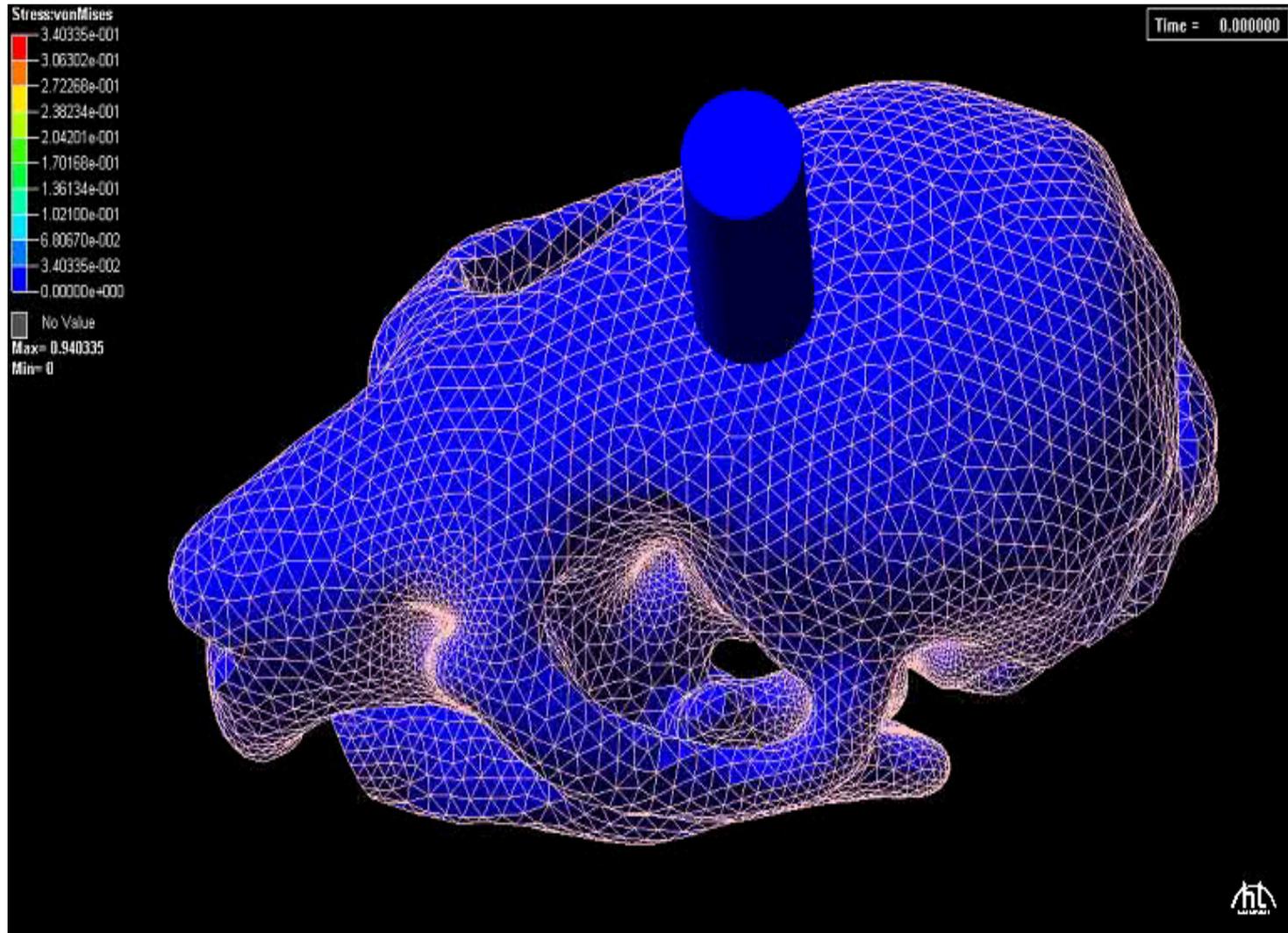
Grid computing

- Two models:
 - Utility computing:
 - Metaphor of electrical power grid
 - Program execute wherever there are computational resources
 - Programs are mobile to take advantage of changing resource availability
 - Distributed collaboration:
 - Large system is built from software components and instruments residing at different sites on Internet
 - High-performance distributed computation

ASP Test Problem 1: The Pipe (Portion of Regenerative Nozzle)

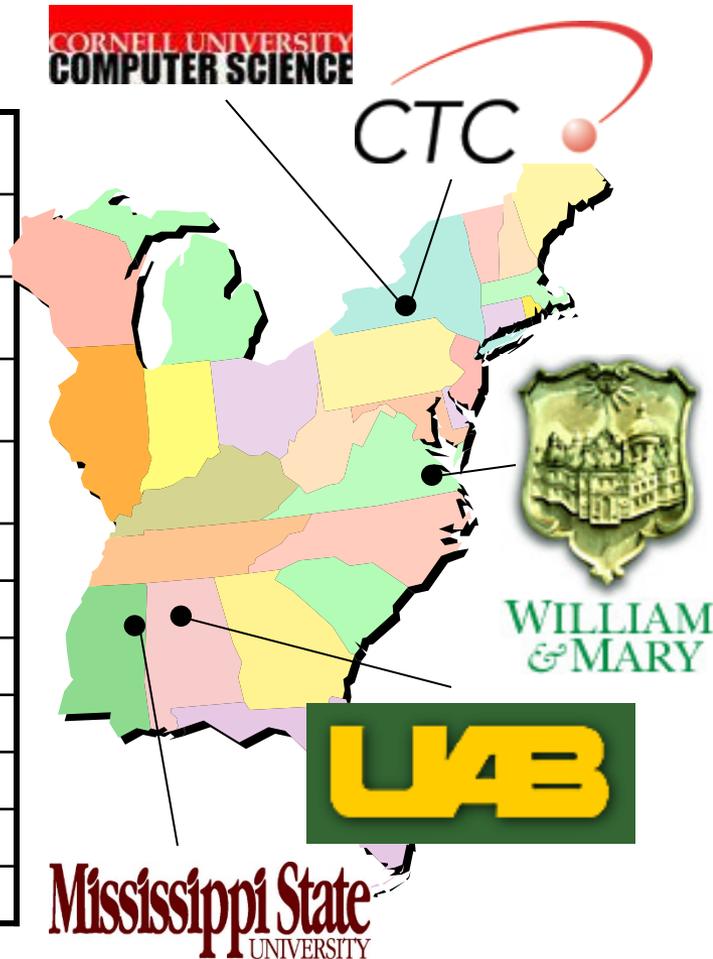


ASP Test Problem 2: Brain-trauma



Component Development

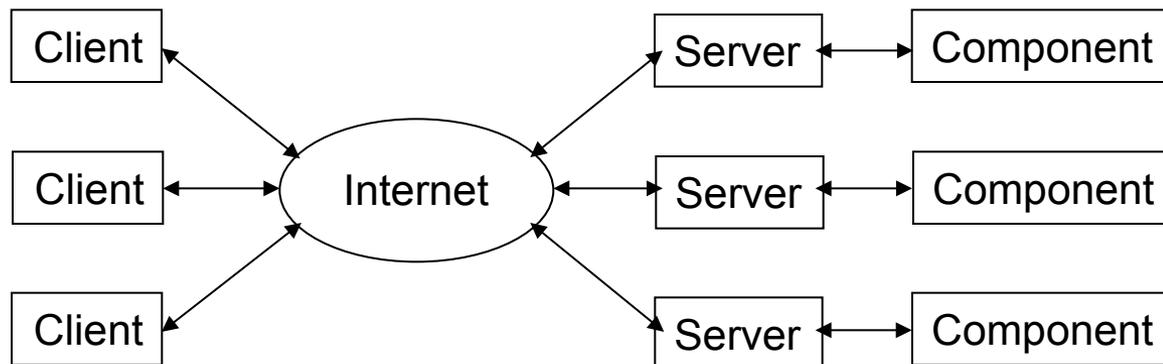
Component	CUCS	CTC	MSU	UAB	CW&M
Surface Mesher	X	*			X
JMesh	X	*			X
AFT				*	*
T4→T10	X				
Generalized Meshers	X		X		
Fluid/Thermal Solver	X		X		
Mechanical Solver		X			
Crack Insertion	X				
Fracture Mechanics	X				
Crack Growth	X				
Feature Detection			*		



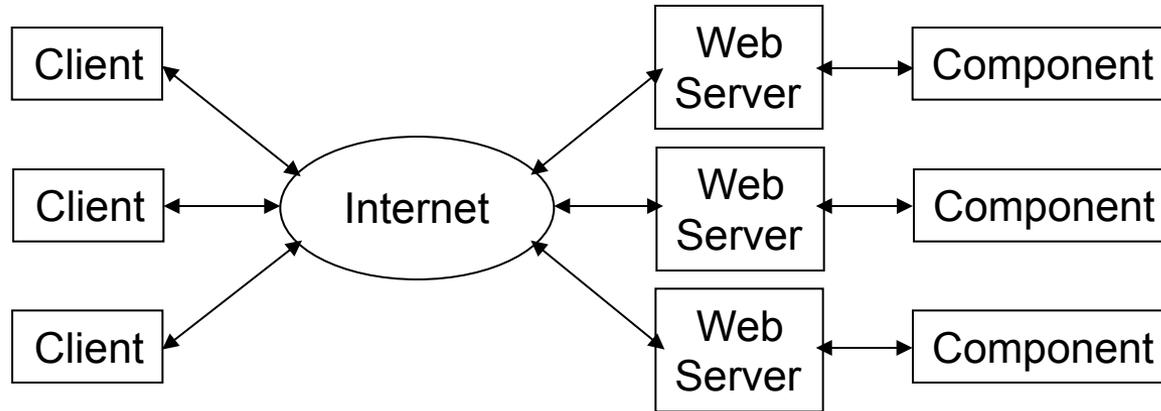
X = component exists
 * = planned

Component Integration

- **Development Realities**
 - Developers at 5 schools, many departments
 - Millions of lines of code
 - Codes used for many projects
 - Porting to a common development platform is not practical
- **Needed: a distributed component framework**



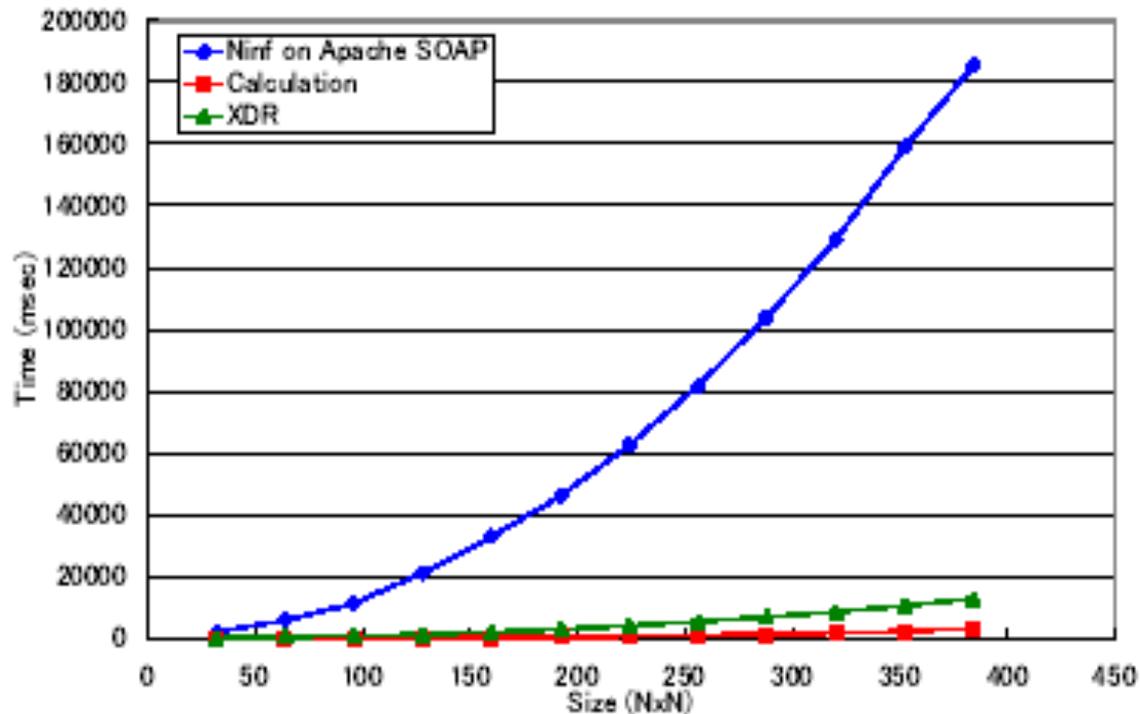
ASP solution: use Web Services



- Motto: "Write once, run from anywhere"
- Web services: SOAP/XML
 - Remote procedure call API and implementation for B2B applications
 - Supported by all vendors
- Components
 - expose **web services** on server located at appropriate developer site
- Communication
 - XML-based data formats for geometry/mesh/...: Vavasis et al
- Workflow defined by client

Study of Shirasuna et al

- Distributed matrix multiplication
- Communication used NINF API on top of SOAP/HTTP
- 100-fold slowdown!
- Conclusion: overheads make SOAP/XML impractical for high-performance distributed computing
- Global Grid Forum: special protocols and API's for HPC



Overheads

Sizes	Surface Mesh			Fluid Mesh	Solid Mesh
	#Vertices	#Edges	#Triangles	#Tets	#Tets
1	4835	639	4979	38220	22045
2	16832	966	10322	85183	83609
3	54849	1413	21127	170179	289500

<u>Size</u>	<u>Local</u> (sec.s)	<u>Cornell client</u> (sec.s)	<u>UAB client</u> (sec.s)
1	1631	1719 5%	1700 4%
2	5594	5777 3%	5746 2.7%
3	22203	22901 3%	22222 0.1%

Infrastructure Conclusions

- Key elements of ASP infrastructure
 - Web services-based simulation components
 - Standard formats for scientific data
- Works well for our applications
 - Low communication to computation ratios
 - Loosely-coupled components
- May not be suited for other applications
 - Matrix multiplication across Internet (but who wants to do this anyway??)
 - Lighter-weight components

Industry Reaction

This project demonstrates the potential of a new way to build applications and the potential for a new software industry structure based on delivering results.

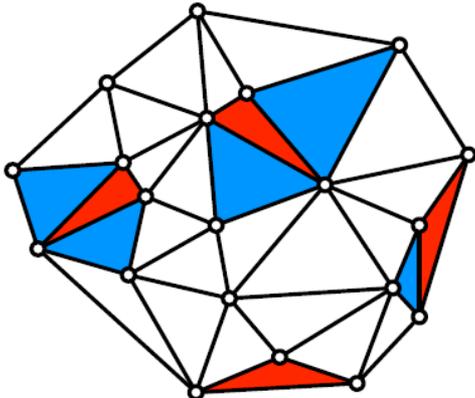
Users don't have to buy apps programs and maintain a more complex software environment; they simply call a program or database.

This is one of the few projects that I would call a Web Service, and it is well beyond what is running on today's experimental Grid.

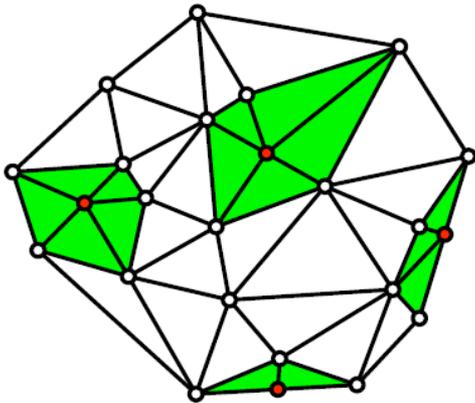
– Gordon Bell (GridToday,2005)

Galois project (2007-)

- Conventional wisdom at that time
 - To execute a program in parallel, build its dependence graph using compiler analysis or knowledge of algorithm
- Delaunay mesh refinement
 - Lots of parallelism
 - Cannot build dependence graph at compile-time
 - Requires runtime parallelization techniques
- Galois system (2007-)
 - Implementation of runtime parallelization techniques using library and runtime system
 - Important application area: graph analytics



Before



After

Delaunay mesh refinement

Red Triangle: badly shaped triangle

Blue triangles: cavity of bad triangle

Things I learnt from Tony

- Fly-fishing
- Running group meetings
- Motivating group members
- Wise sayings
 - You're never too old to learn something stupid.
 - If I agreed with you, we'd both be wrong.
 - You don't need a parachute to skydive; you only need it to skydive more than once.

Hog-hunting with Tony at Tinsley Oden's ranch

