High Fidelity Computational Model for Fluidized Bed Experiments

Vinod Kumar, Ph.D.
Associate Professor, Mech. Engg. & Core faculty, Computational Science,
University of Texas, El Paso (**UTEP**)

William (**Bill**) Spotz, Ph.D.
Senior Staff Scientist, **Sandia** National Labs, Albuquerque

Outline

- Technical goal
- Objectives
- Background
- Technical approach
- Tasks and subtasks
- Team description and assignments
- Project milestones, budget and schedule
- Project risks and risk management plan
- Project status
- Concluding remarks-

Technical goal

The technical goal of this project is to develop, validate and implement **advanced linear solvers** to replace the existing linear solvers that are used by the National Energy Technology Laboratory's (**NETL**) open source software package Multiphase Flow with Interphase eXchanges (**MFIX**). This goal will be achieved by integrating **Trilinos**, a publicly available open-source linear equation solver library developed by **Sandia** National Laboratory, with MFIX. The project will demonstrate scalability of the Trilinos- MFIX interface on various high-performance computing (HPC) facilities including the ones funded by the Department of Energy (DOE).

The expected results of the project will be **reduction of computational time** when solving complex gas-solid flow and reaction problems in MFIX, and reduction in time and cost of adding new algorithms and physics based models into MFIX

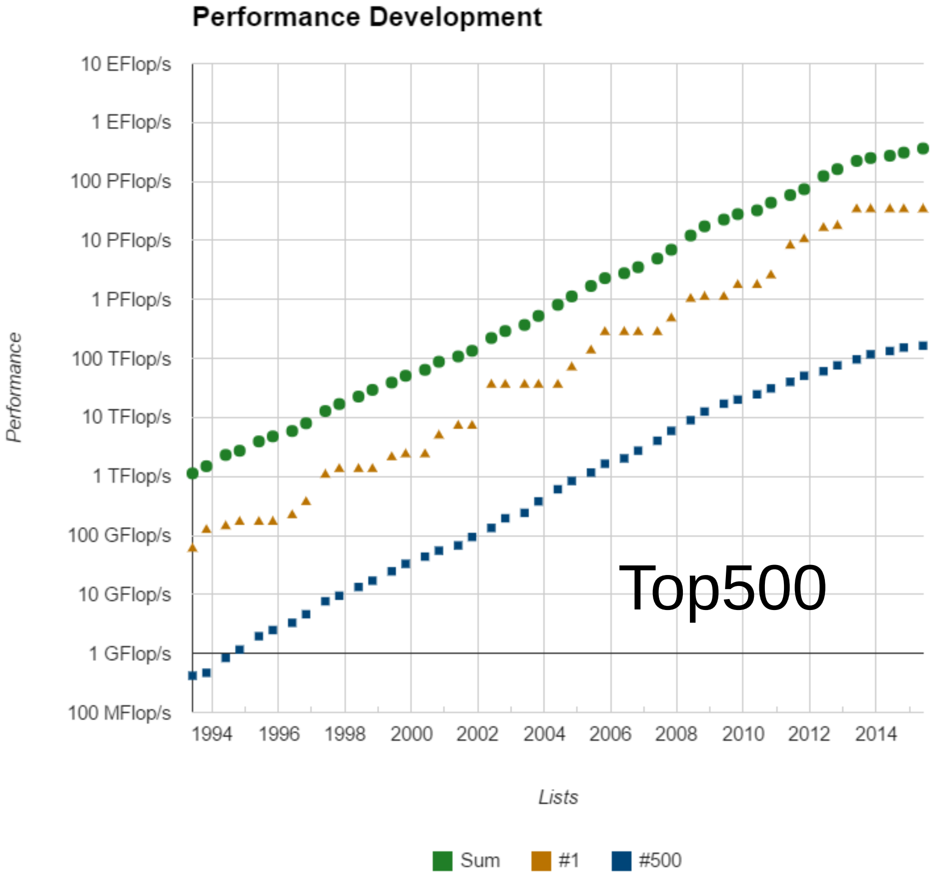
Objectives

- Create a framework to integrate the existing MFIX linear solver with Trilinos linear solver packages,
- Evaluate the performance of the state-of-the-art preconditions and linear solver libraries in Trilinos with MFIX, and
- Test three dimensional (3D) MFIX suites of problems on massively parallel computers with and without GPU acceleration.

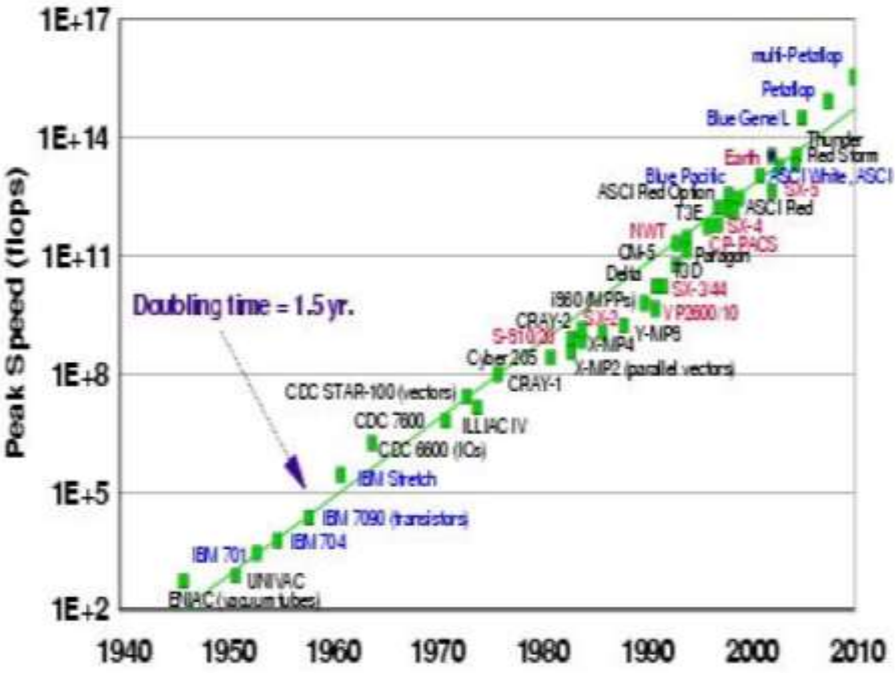
Background

The Multiphase Flow with Interphase eXchanges (MFIX) software package, a multiphase Computational Fluid Dynamics (CFD) software developed by NETL, is a **widely used by the fossil fuel reactor communities to model and understand the multiphase physics in a circulating fluidized bed**. In MFIX, gas-solids are addressed by solving coupled continuity and momentum conservation and parameterizing many effects such as drag force, buoyancy, virtual mass effect, lift force, Magnus force, Basset force, Faxen force, etc.

Supercomputers



Top500



Google Compute Engine (Cloud computer):
16core, 104 GB, $1.184/hr

MFiX (From MFiX reports)

Rogers, Syamlal, O'Brien

Gas continuity:

$$\frac{\partial}{\partial t}(\epsilon_g \rho_g) + \nabla \cdot (\epsilon_g \rho_g \vec{v}_g) = \sum_{n=1}^{N_g} R_{gn}$$

Solids continuity:

$$\frac{\partial}{\partial t}(\epsilon_{sm} \rho_{sm}) + \nabla \cdot (\epsilon_{sm} \rho_{sm} \vec{v}_{sm}) = \sum_{n=1}^{N_{sm}} R_{smn}$$

Gas momentum balance:

$$\frac{\partial}{\partial t}(\epsilon_g \rho_g \vec{v}_g) + \nabla \cdot (\epsilon_g \rho_g \vec{v}_g \vec{v}_g) = -\epsilon_g \nabla P_g + \nabla \cdot \overline{\overline{\tau}}_g + \sum_{m=1}^{M} F_{gm}(\vec{v}_{sm} - \vec{v}_g) + \vec{f}_g$$

$$+ \epsilon_g \rho_g \vec{g} - \sum_{m=1}^{M} R_{0m}\left[\xi_{0m} \vec{v}_{sm} + \overline{\xi}_{0m} \vec{v}_g\right]$$

Convection-diffusion/Transport of species

- First-order schemes/Second-order/High-order schemes

The use of higher order methods may result in a violation of Patankar's Rule 2 in some regions!

- Downwind factors

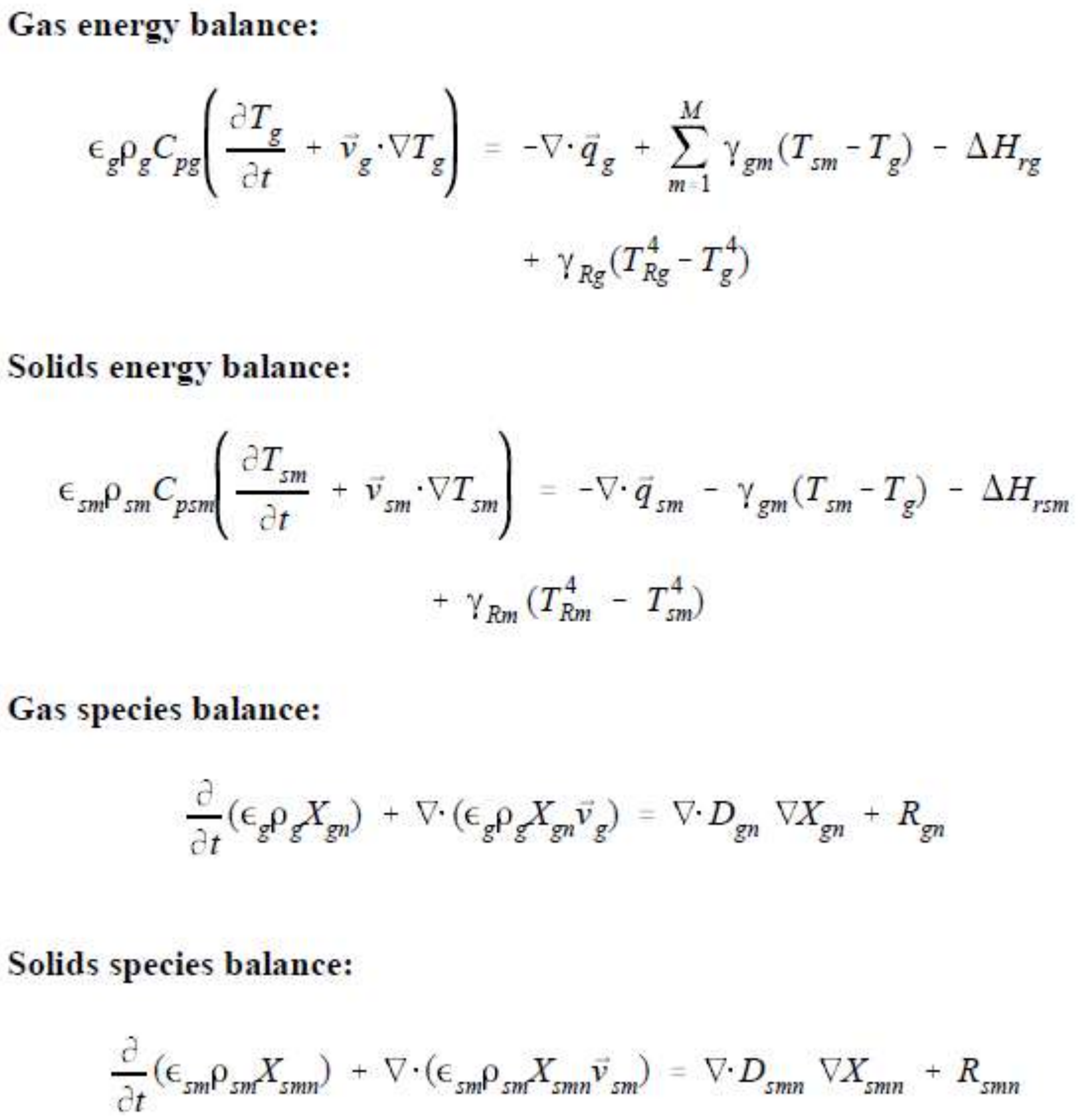
$$\frac{\partial}{\partial t}(\epsilon_m \rho_m \phi) + \frac{\partial}{\partial x_i}(\epsilon_m \rho_m v_{mi} \phi) = \frac{\partial}{\partial x_i}\left(\Gamma_\phi \frac{\partial \phi}{\partial x_i}\right) + R_\phi$$

$$\rho u \frac{\partial \phi}{\partial x} - \frac{\partial}{\partial x}\left(\Gamma \frac{\partial \phi}{\partial x}\right)$$

- Numerical diffusion
- Consistency

MFiX Eqs & Schemes (From MFiX reports)

Rogers, Syamlal, O'Brien



Gas energy balance:

$$\epsilon_g \rho_g C_{pg} \left(\frac{\partial T_g}{\partial t} + \vec{v}_g \cdot \nabla T_g \right) = -\nabla \cdot \vec{q}_g + \sum_{m=1}^{M} \gamma_{gm} (T_{sm} - T_g) - \Delta H_{rg} + \gamma_{Rg} (T_{Rg}^4 - T_g^4)$$

Solids energy balance:

$$\epsilon_{sm} \rho_{sm} C_{psm} \left(\frac{\partial T_{sm}}{\partial t} + \vec{v}_{sm} \cdot \nabla T_{sm} \right) = -\nabla \cdot \vec{q}_{sm} - \gamma_{gm} (T_{sm} - T_g) - \Delta H_{rsm} + \gamma_{Rm} (T_{Rm}^4 - T_{sm}^4)$$

Gas species balance:

$$\frac{\partial}{\partial t} (\epsilon_g \rho_g X_{gn}) + \nabla \cdot (\epsilon_g \rho_g X_{gn} \vec{v}_g) = \nabla \cdot D_{gn} \nabla X_{gn} + R_{gn}$$

Solids species balance:

$$\frac{\partial}{\partial t} (\epsilon_{sm} \rho_{sm} X_{smn}) + \nabla \cdot (\epsilon_{sm} \rho_{sm} X_{smn} \vec{v}_{sm}) = \nabla \cdot D_{smn} \nabla X_{smn} + R_{smn}$$

MFiX Eqs & Schemes (From MFiX reports)

Rogers, Syamlal, O'Brien

Gas-solids drag:

$$F_{gm} = \frac{3\epsilon_{sm}\epsilon_g\rho_g}{4V_{rm}^2 d_{pm}}\left(0.63 + 4.8\sqrt{V_{rm}/Re_m}\right)^2 \left|\vec{v}_{sm} - \vec{v}_g\right|$$

Solids-solids drag:

$$F_{slm} = \frac{3(1+e_{lm})\left(\frac{\pi}{2} + \frac{C_{flm}\pi^2}{8}\right)\epsilon_{sl}\rho_{sl}\epsilon_{sm}\rho_{sm}(d_{pl}+d_{pm})^2 g_{0_{lm}} \left|\vec{v}_{sl} - \vec{v}_{sm}\right|}{2\pi(\rho_{sl}d_{pl}^3 + \rho_{sm}d_{pm}^3)}$$

$$g_{0_{lm}} = \frac{1}{\epsilon_g} + \frac{3\left(\sum_{\lambda=1}^{M} \epsilon_{s\lambda}/d_{p\lambda}\right) d_{pl}d_{pm}}{\epsilon_g^2(d_{pl}+d_{pm})}$$

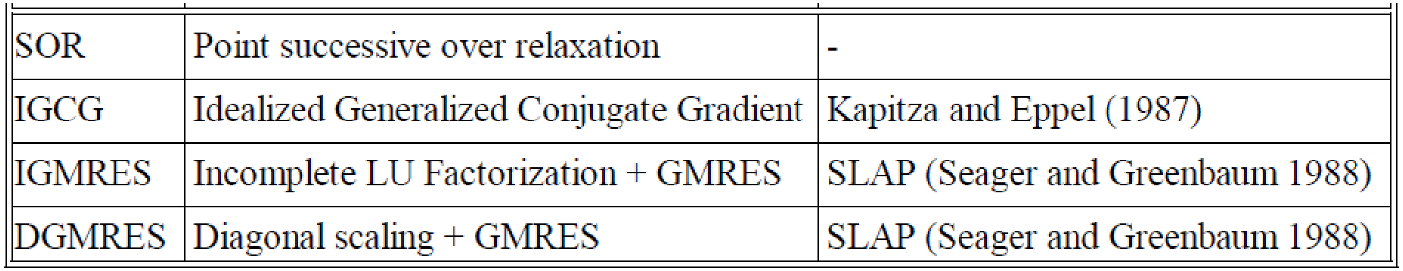
And, Even More Equations…

MFiX

Solution algorithms

- SIMPLE (Patankar 1980)
  - More variables than single phase (slows computations)
  - The multiphase momentum equations are strongly coupled through the momentum exchange term
- Handling of close-packed regions
- Fluid-pressure correction
- Boundary conditions
- **Numerical schemes**
  - **Linear/Non-linear system of equations solver**
  - **Parallel/Distributed computing**
    - **Partitioning/Domain decomposition**
    - **I/O , Data management/Cloud**
    - **Number crunching – HPC/GPU?**
- **Graphics**
  - **Data-storage**
  - **Rendering**

MFIX



SOR	Point successive over relaxation	-
IGCG	Idealized Generalized Conjugate Gradient	Kapitza and Eppel (1987)
IGMRES	Incomplete LU Factorization + GMRES	SLAP (Seager and Greenbaum 1988)
DGMRES	Diagonal scaling + GMRES	SLAP (Seager and Greenbaum 1988)

- has basic linear equations solvers (such as Point successive over relaxation or SOR, Idealized Generalized Conjugate Gradient (IGCG), Incomplete LU Factorization + Generalized Minimal RESidual (IGMRES), and Diagonal scaling + GMRES or DGMRES) that results from the discretization of transport equations.
- poor convergence in the linear equation solver can increase the number of iterations and lead to nonconvergence of the iterations.
- An optimum degree of convergence has been determined from experience and is controlled by a specified number of iterations inside the linear solver.
- The current capabilities in MFIX however **lack the advanced solvers**(such as multi-level, segregated/block, algebraic preconditions, planned development time integration methods, modular and latest linear/non-linear solvers).

MFIX Challenges

In nutshell, although **MFIX** is **increasingly being used** to design & scale-up of fossil fuel reactors, overall utility of the multiphase models remains **limited** due to the **computational expense** of large scale simulations. The time-to-solution however can be reduced by **leveraging** state-of-the-art **preconditions and linear solver libraries** where majority of processor-level time is spent in solving large systems of linearized equations.

Technical approach

One of the main challenges for any software development is keeping the computer code up-to-date with the advancement in applied mathematics, software and hardware development in computational science and engineering. Realizing the challenge, the CSRI group at Sandia has developed and continues to develop scalable solver algorithms and software through next-gen (exa-scale, peta-scale, exteme-scale, etc.) computing investment. The project is called **Trilinos** project.

Funded by various DOE entities mainly NNSA - Advanced Simulation and Computing (**ASC**)/*DOE Office of Science (***SciDAC**), Advanced Scientific Computing Research (**ASCR**)

Note: Slides in this topic mostly borrowed from M.Heroux & other trilinos members

Trilinos

*The Trilinos Project is an effort to develop and implement **robust algorithms** and **enabling technologies** using modern object-oriented software design, while still leveraging the value of established libraries such as **PETSc, Metis/ParMetis, SuperLU, Aztec, the BLAS and LAPACK**. It emphasizes **abstract interfaces** for maximum **flexibility** of component interchanging, and provides a full-featured set of concrete classes that implement all abstract interfaces. Research efforts in **advanced solution algorithms** and **parallel solver** libraries have historically had a large impact on engineering and scientific computing. Algorithmic advances increase the range of tractable problems and reduce the cost of solving existing problems. Well-designed solver libraries provide a mechanism for leveraging solver development across a broad set of applications and minimize the cost of solver integration. Emphasis is required in both new algorithms and new software (Heroux et.al., http://trilinos.sandia.gov/).*

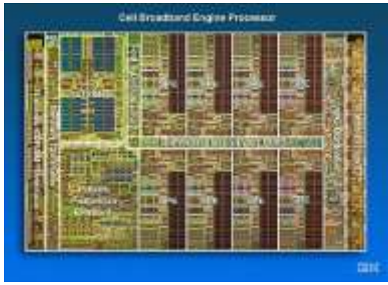
What is Trilinos?

- Object-oriented software framework for…
- Solving big complex science & engineering problems
- More like LEGO™ bricks than Matlab™

Trilinos provides the state-of-the-art in preconditions and **linear solver** libraries

- demonstrate **scalability** on **current HPC systems**
- illustrate plans for **continued maintenance**
- include **support for new hardware technologies**

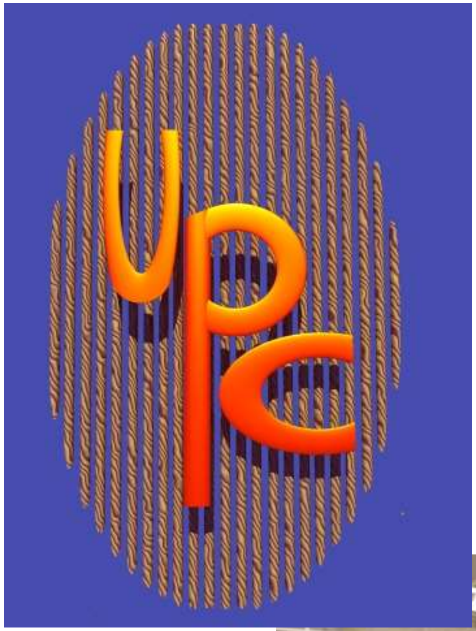
Target Platforms



Desktop: Development and more…

Capability machines:

- Redstorm (XT3), Clusters
- Roadrunner (Cell-based).
- **Multicore nodes**.

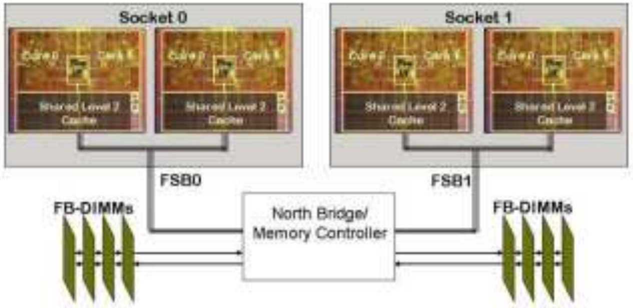


Parallel software environments:

- MPI
- UPC, CAF, threads, vectors,…
- Combinations of the above.

User "skins":

- C++/C, Python
- Fortran.
- **Web**, CCA.



Unique features of Trilinos

- Huge library of algorithms
  - Linear & nonlinear solvers, preconditioners, …
  - Optimization, transients, sensitivities, uncertainty, …
  - Discretizations, mesh tools, automatic differentiation, …
- Package-based architecture
- Support for huge (> 2B unknowns) problems
- Support for mixed & arbitrary precisions
- Growing support for hybrid (MPI+X) parallelism
  - X: Threads (CPU, Intel Xeon Phi, CUDA on GPU)
  - Built on a unified shared-memory parallel programming model: Kokkos (see Session 2 & later this week)
  - Support currently limited, but growing

Evolving Trilinos Solution

Trilinos[1] is an evolving framework to address these challenges:

- Fundamental atomic unit is a *package*.
- Includes core set of vector, graph and matrix classes (Epetra/**Tpetra** packages).
- Provides a common abstract solver API (Thyra package).
- Provides a ready-made package infrastructure (new_package package):
  - Source code management (cvs, git, bonsai).
  - Build tools (Cmake).
  - Automated regression testing.
  - Communication tools (mailman mail lists).
- Specifies requirements and suggested practices for package SQA.

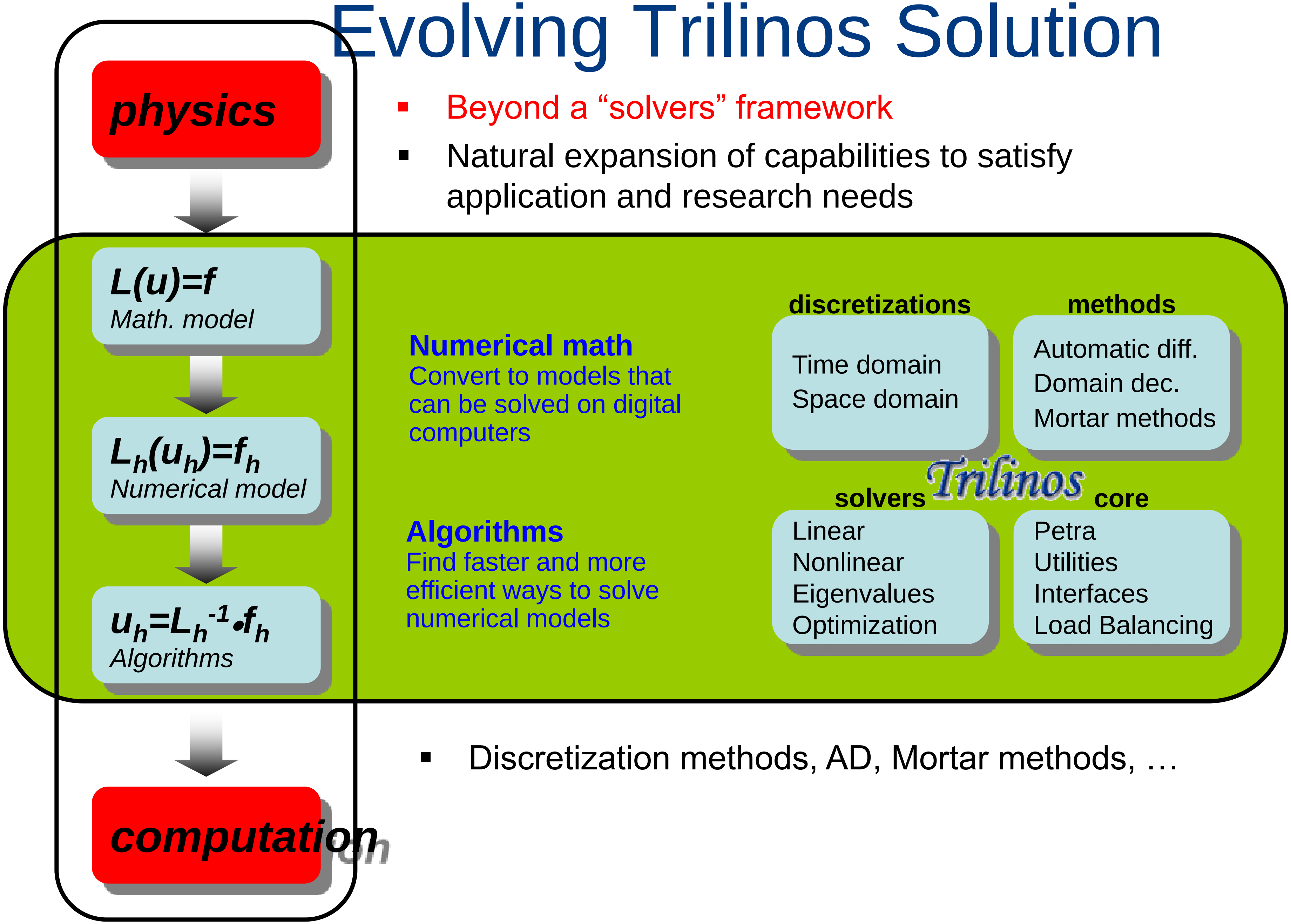
In general allows us to categorize efforts:

- Efforts best done at the Trilinos level (useful to most or all packages).
- Efforts best done at a package level (peculiar or important to a package).

Allows package developers to focus only on things that are unique to their package.

Evolving Trilinos Solution

- Beyond a “solvers” framework
- Natural expansion of capabilities to satisfy application and research needs



physics

$L(u)=f$
Math. model

$L_h(u_h)=f_h$
Numerical model

$u_h=L_h^{-1}\cdot f_h$
Algorithms

computation

Numerical math
Convert to models that can be solved on digital computers

Algorithms
Find faster and more efficient ways to solve numerical models

discretizations
Time domain
Space domain

methods
Automatic diff.
Domain dec.
Mortar methods

Trilinos

solvers
Linear
Nonlinear
Eigenvalues
Optimization

core
Petra
Utilities
Interfaces
Load Balancing

- Discretization methods, AD, Mortar methods, …

Trilinos Strategic Goals

Scalable Computations: As problem size and processor counts increase, the cost of the computation will remain nearly fixed.

Hardened Computations: Never fail unless problem essentially intractable, in which case we diagnose and inform the user why the problem fails and provide a reliable measure of error.

Full Vertical Coverage: Provide leading edge enabling technologies through the entire technical application software stack: from problem construction, solution, analysis to optimization.

Algorithmic Goals

Grand Universal Interoperability: All Trilinos packages will be interoperable, so that any combination of packages that makes sense algorithmically will be possible within Trilinos and with compatible external software.

Universal Accessibility: All Trilinos capabilities will be available to users of major computing environments: C++, Fortran, Python and the Web, and from the desktop to the latest scalable systems.

Universal Capabilities RAS: Trilinos will be:

- Integrated into every major application at Sandia (**Availability**).
- The leading edge hardened, efficient, scalable solution for each of these applications (**Reliability**).
- Easy to maintain and upgrade within the application environment (**Serviceability**).

Software Goals

Trilinos Packages

Trilinos is a collection of *Packages*.

- Each package is:
  - Focused on important, state-of-the-art algorithms in its problem regime.
  - Developed by a small team of domain experts.
  - Self-contained: No explicit dependencies on any other software packages (with some special exceptions).
  - Configurable/buildable/documented on its own.
- Sample packages: NOX, AztecOO, ML, IFPACK, Meros.
- Special package collections:
  - Petra (Epetra, **Tpetra**, Jpetra): Concrete Data Objects
  - **Thyra**: Abstract Conceptual Interfaces
  - Teuchos: Common Tools.
  - New_package: Jumpstart prototype.
- **Multi-scale/physics** simulation– Panzer
  - **User Physics Kernels + Problem Description** = Thyra::ModelEvaluator

Trilinos Package Summary

	Objective	Package(s)
Discretizations	Meshing & Discretizations	Intrepid, Pamgen, Sundance, Mesquite, STKMesh
	Time Integration	Rythmos
Methods	Automatic Differentiation	Sacado
	Mortar Methods	Moertel
Services	Linear algebra objects	Epetra, Tpetra
	Interfaces	Xpetra, Thyra, Stratimikos, Piro, …
	Load Balancing	Zoltan, Isorropia, Zoltan2
	"Skins"	PyTrilinos, WebTrilinos, ForTrilinos, CTrilinos
	Utilities, I/O, thread API	Teuchos, EpetraExt, Kokkos, Phalanx, Trios, …
Solvers	Iterative linear solvers	AztecOO, Belos, Komplex
	Direct sparse linear solvers	Amesos, Amesos2, ShyLU
	Direct dense linear solvers	Epetra, Teuchos, Pliris
	Iterative eigenvalue solvers	Anasazi
	Incomplete factorizations	AztecOO, Ifpack, Ifpack2
	Multilevel preconditioners	ML, CLAPS, MueLu
	Block preconditioners	Meros, Teko
	Nonlinear solvers	NOX, LOCA
	Optimization	MOOCHO, Aristos, TriKota, GlobiPack, OptiPack
	Stochastic PDEs	Stokhos

Full Vertical Solver Coverage

Trilinos

Optimization Unconstrained: Constrained:	Find $u \in \Re^n$ that minimizes $g(u)$ Find $x \in \Re^m$ and $u \in \Re^n$ that minimizes $g(x,u)$ s.t. $f(x,u)=0$	Sensitivities (Automatic Differentiation: Sacado)	MOOCHO
Bifurcation Analysis	Given nonlinear operator $F(x,u) \in \Re^{n+m}$ For $F(x,u)=0$ find space $u \in U \ni \frac{\partial F}{\partial x}$		LOCA
Transient Problems DAEs/ODEs:	Solve $f(\dot{x}(t), x(t), t) = 0$ $t \in [0,T], x(0) = x_0, \dot{x}(0) = x_0'$ for $x(t) \in \Re^n, t \in [0,T]$		Rythmos
Nonlinear Problems	Given nonlinear operator $F(x) \in \Re^m \to \Re$ Solve $F(x)=0$ $x \in \Re^n$		NOX
Linear Problems Linear Equations: Eigen Problems:	Given Linear Ops (Matrices) $A, B \in \Re^{m \times n}$ Solve $Ax=b$ for $x \in \Re^n$ Solve $A\nu = \lambda B \nu$ for (all) $\nu \in \Re^n$, $\lambda \in$		AztecOO Belos Ifpack, ML, etc... Anasazi
Distributed Linear Algebra Matrix/Graph Equations: Vector Problems:	Compute $y = Ax; A = A(G); A \in \Re^{m \times n}, G \in \Im^{m \times n}$ Compute $y = \alpha x + \beta w; \alpha = \langle x, y \rangle; x, y \in \Re^n$		Epetra Tpetra Kokkos

Proposed Tasks and subtasks

Task1

Task 1.0 – Project Management and Planning

The Recipient will develop and maintain a project management plan to foster team interaction, track deliverables, maintain and implement a project risk management plan, interface with DOE, and report progress and financials in accordance with the requirements set forth in the award document. Any proposed revisions to deliverables, milestones, project schedule, or budget shall be reported to DOE in accordance with the terms and conditions of the award.

Task2

Task 2.0 – Assembly of Optimum Trilinos Linear Equation Package for Integration with MFIX

Setup a GIT/version-control repository for development of the Trilinos-MFIX interface, choose the most suitable Trilinos software package for this project, and develop a ForTrilinos based Fortran program interface for MFIX

Subtasks

Subtask 2.1: Setup a GIT/version-control repository
Establish a GIT repository so that changes to files can be recorded and recalled, and to accelerate creation, merging and deletion of computer code in the Trilinos-MFIX interface

Subtask 2.2: Select the optimum Trilinos software package
Select the Trilinos software package for MFIX that best supports distributed data structures and provides access to performance-portable algorithms for Graphical Processing Units (GPU) and other HPC architectures

Subtask 2.3: Develop a ForTrilinos based Fortran interface for MFIX
Develop a software interface to link the Fortran based MFIX software with the C/C++ based Trilinos software by developing a complete link between the object oriented Fortran interface package, ForTrilinos, with the Stratimokos package in Trilinos.

EPetra/TPetra Software Stacks

Challenges: Many different third-party solvers but no clear winner for all problems. Different, changing interfaces & data formats, serial & parallel

Targeted solver package

Amesos: Interface to sparse direct solvers

- Accepts Epetra & Tpetra sparse matrices & dense vectors

AztecOO

- Iterative linear solvers: CG, GMRES, BiCGSTAB,…
- Incomplete factorization preconditioners

Belos Next-generation linear iterative solvers

- Block & pseudoblock solvers: GMRES & CG
- Recycling solvers: GCRODR (GMRES) & CG
- “Seed” solvers (hybrid GMRES)
- Block orthogonalizations (TSQR)

Developers: M. Heroux, et.al.

Task3

Task 3.0 – Performance Evaluation of Preconditions and Linear Solver Libraries

Evaluate the performance of the preconditions and linear solver libraries, perform a scalability analysis for use of the selected libraries on large parallel computing systems, and improve computing performance of the Trilinos-MFIX interface.

Subtasks

Subtask 3.1: Test and compare the linear equation solver packages in Trilinos
Run test cases of fluidized bed simulations in MFIX using the various Trilinos linear equations preconditioner and solver packages. Compare the performance (computing time and accuracy) of the MFIX- Trilinos package with solutions that use MFIX and its existing linear solvers.

Subtask 3.2: Perform a scalability analysis of Trilinos-MFIX
Conduct scalability tests of the Trilinos-MFIX package on single node and multi-core computer clusters using distributed/shared or in hybrid environment HPC systems. Test multi-core clusters containing 4, 16, 64, 128, 512 and 1024 cores.

Subtask 3.3: Improve performance of Trilinos-MFIX
Use the profiling and debugging tools in Trilinos to determine the bottlenecks in the Trilinos-MFIX package and then improve the performance (computing time and accuracy of solution) of the Trilinos linear solvers when they are used in MFIX.

Task4

Task 4.0 –Performance Evaluation of MFIX with the Trilinos Linear Solver on Massively Parallel Computers

Obtain computing time on one or more massively parallel HPC systems, compile and test the Trilinos-MFIX package on those systems, and compare its performance with the existing MFIX-linear solvers using selected gas-solid fluidized bed problems that have been previously solved using MFIX.

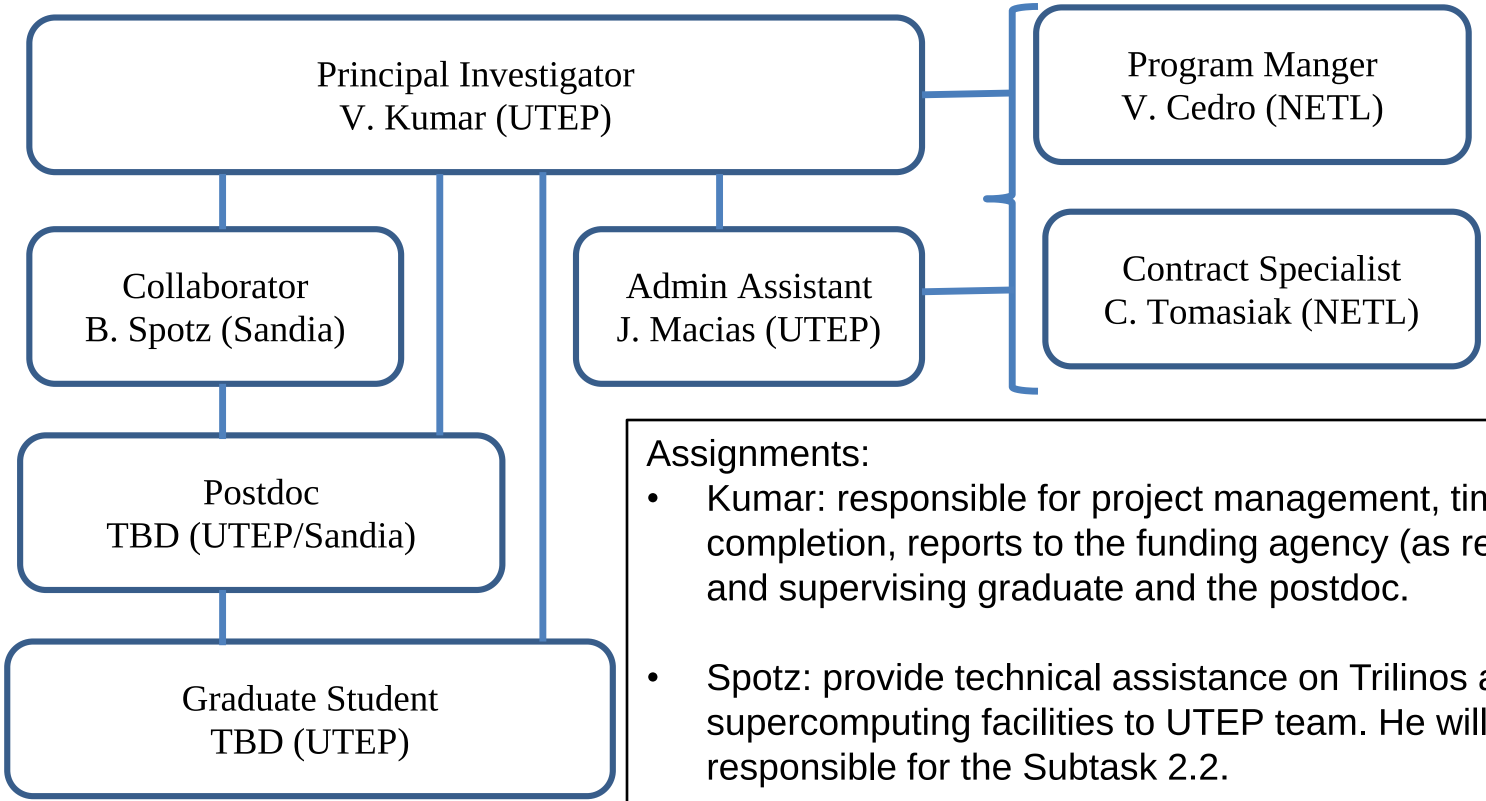
Subtask 4.1: Secure computational time on massively parallel computers

Subtask 4.2: Compile Trilinos-MFIX on the selected massively parallel computer(s)

Subtask 4.3: Run simulations of a fluidized bed test problems with various particle sizes and shapes

Subtask 4.4: Analyze Trilinos-MFIX performance for various computer architectures and fluidized bed test problems

Team description, assignments & organization



Assignments:

- Kumar: responsible for project management, timely task completion, reports to the funding agency (as required), and supervising graduate and the postdoc.
- Spotz: provide technical assistance on Trilinos and DOE supercomputing facilities to UTEP team. He will be responsible for the Subtask 2.2.
- Postdoc: responsible for Task 2 & 3 & assisting the graduate student to perform the tests.
- Graduate student: responsible for the Task3.

Principal Investigator: Dr. V. Kumar, University of Texas at El Paso

UTEP	**Associate Professor**	2014 - Present
	Assistant Professor, Mechanical Engineering	2008-2014
GFDL(Princeton Univ./NOAA)	Research Scientist, Climate Modeling	2007-2008
Rice Univ,	PostDoc, Physic & Astronomy	2005-2007
Rice Univ,	Ph.D., Mechanical Engineering	1999-2005
IIT Kanpur	B.Tech., Aerospace Engineering	1993-1997

Sabbatical	**Sandia, Trilinos**/CSRI	**08/2015** – 2016
Visiting Professor	Sandia**,** CSP	09/2012 – 12/2014
Visiting Professor	Sandia, Climate Modeling	09/2010 – 08/2013
AFOSR Visiting Faculty	KAFB/Maui, Atmospheric turbulence	06/2012 –08/2012
DOE FaST Fellow	NREL, Thermal-fluid/CSP	06/2011 – 08/2011
Cons/Smr-faculty	Sandia, Climate Modeling	06/2010 – 08/2010
ORISE Visiting Faculty	**NETL**, Geological Sequestration	06/2010 – 08/2010
Visiting Faculty	ORNL, Climate Modeling	06/2009 – 08/2009
Development Engineer	**Fluent (now ANSYS)**	07/1997 – 05/1999
UG RA	IIT Kanpur	07/1995 – 07/1997

Team description, assignments & organization

Collaborator: Dr. William Spotz

Senior Member of Technical Staff, Sandia National Laboratories (SNL)

Education and Training

Postdoc, Adv Study Program, **Nat’l Center for Atmospheric Research** (NCAR)	1996-98
Ph.D., Aerospace Eng, **University of Texas at Austin** (Adv: Prof. Graham Carey)	1991-95
M.S., Aerospace Eng, **University of Texas at Austin**, (Adv: Prof. Graham Carey)	1989-91
B.S., Aerospace Eng, **University of Texas at Austin**	1985-89

Professional Experience

SNL	Senior Member, Tech Staff	Computing Research Center	11/01-Present
DOE	Program Manager	Adv Scientific Computing Rsrch	06/08 - 05/10
NCAR	Project Scientist	Scientific Computing Div	06/98-10/01

Project milestones, budget and schedule

	Title	Description	Related task or subtask	Expected Completion Date	Success Criteria
Budget Year 1:					
Milestone 1.1	GIT repository setup completed	Setup GIT/version-control repository for Trilinos MFIX	Subtask 2.1	Q1	A working repository tested by at least three researchers
Milestone 1.2	Best Trilinos linear solver package decided	Choose the best Trilinos linear solver package for MFIX	Subtask 2.2	Q2	Source code for the decided package uploaded to the GIT repository
Milestone 1.3	Fortran interface for Trilinos MFIX created	Develop Fortran interface for the Trilinos linear solver to communicate with MFIX	Subtask 2.3	Q3-4	A working version of the Fortran interface uploaded
Budget Year 2:					
Milestone 2.1	>30% Linear solved speedup achieved	Test the linear solvers for its performance	Subtask 3.1	Q5	20% or better linear solver speedup achieved
Milestone 2.2	Scalability issues identified	Perform scalability analysis of Trilinos-MFIX	Subtask 3.2	Q6	Scalability testing for up to 1024 cores performed
Milestone 2.3	Bottlenecks to scalability identified and removed	Perform code profiling and identify bottlenecks	Subtask 3.3	Q7-8	Code profiling on Trilinos profiler completed and bottlenecks addressed for one HPC system
Budget Year 3:					
Milestone 3.1	Trilinos MFIX compiled on various OS/architectures	Compile Trilinos-MFIX on various cloud/HPC computers	Subtask 4.3	Q9	Trilinos MFIX compiled on 3 HPC (UTEP, DOE-Sandia, and one more)
Milestone 3.2	MFIX tests suites completed	Run simulations with various particle sizes and shapes of fluidized bed riser test problems	Subtask 4.3	Q10	All 2D runs and one 3D tests validated
Milestone 3.2	Trilinos MFIX performance analysis completed	Analyze Trilinos-MFIX performance for various computing architectures and fluidzed-test problems	Subtask 4.3	Q11-12	Report submitted

Project budget

Year 1		**Year 2**		**Year 3**	
Month	Projected Budget	Quarter	Projected Budget	Quarter	Projected Budget
1	$5,540	1	$40,338	1	$17,420
2	$5,540	2	$40,338	2	$17,420
3	$5,540	3	$40,338	3	$17,420
4	$5,540	4	$52,338	4	$29,420
5	$13,851				
6	$13,851				
7	$13,851				
8	$13,851				
9	$13,851				
10	$25,851				
11	$13,851				
12	$13,851				
Total	**$144,969**		**$173,352**		**$81,678**
Total all Year	**$399,999**				

Project schedule

Task Title		Budget Period: 1st half						Budget Period: 2nd half					
		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12
Program Management		▒											
Software preparations		▓	▓	▓	▓								
Subtask 2.1		▒	▒	▒									
Subtask 2.2			▒	▒	▒								
Subtask 2.3				▒	▒								
Trilinos MFIX Linear Solver					▓	▓	▓	▓	▓				
Subtask 3.1					▒	▒	▒						
Subtask 3.2						▒	▒	▒	▒				
Subtask 3.3								▒	▒				
MFIX suite tests								▓	▓	▓	▓	▓	▓
Subtask 4.1								▒	▒	▒			
Subtask 4.2									▒	▒	▒	▒	▒
Subtask 4.3										▒	▒	▒	▒

Project risks & risk management plan

Risk Category	Description of the Risk	Probability of Occurring	Impact	Overall Level of the Risk	Risk Mitigation Strategy
Internal Data Storage Array	How will the internal data storage arrays (a setpadiagonal matrix) be reconciled with native Trilinos data structures	Med	Low	Low	1) Define a class that directly copy the data 2) Compressed sparse row matrix storage 3) Discussions with experts at Sandia to determine which approach would be best
ForTrilinos	Trilinos Fortran interfaces	High	Low	Low	1) Consider Python as a glue language since it provides automatic wrapper generators. 2) Leverag from collaborators' existing PyTrilinos project 3) Integrate the fortran interface fully with Stratimikos
External supercomputer access	Securing access to the external supercomputers	Low	Low	Low	Added as subtask to Secure external resources. Strategies are 1) Work with Sandia Collaborator to secure DOE's HPC for Trilinos MFIX performance testing 2) Request for more allocation on TACC HPC through UT System HPC initiatives 3) Write proposal to secure computing hours on XSEDE HPC resources 4) PI already has access to Linux and IBM clusters through UTEP's HPC facilities

Proje[redacted]

- Setup [redacted] repository
- Select optimum Trilinos software package
- Develop ForTrilinos based Fortran interface for MFIX
- Test and compare the linear solver packages in Trilinos
    - Run test cases of fluidized bed simulations in MFIX using the various Trilinos linear equations preconditioner. Compare the performance (computing time and accuracy) of the MFIX- Trilinos package with solutions that use MFIX and its existing linear solvers
- Perform scalability analysis of Trilinos-MFIX
    - Conduct scalability tests of the Trilinos-MFIX package on single node and multi-core computer clusters using distributed/shared or in hybrid environment HPC systems. Test multi-core clusters containing 4, 16, 64, 128, 512, 1024, 8192(?) cores.
- Address Trilinos-MFIX performance bottlenecks via profiling and debugging tools in Trilinos
- Run fluidized bed test problems (various particle sizes and shapes, 2D/3D, Small/Large Scale, etc.)

Compilation Script – Trilinos 11.10.2

Install cmake28
Install Python Modules: Numpy, Scipy, Cython
Install matio libraries
Install swig 2.0.8
Install mpi4py
Install openGLM
Install Trilinos

```
#!/bin/bash
# Base on https://code.google.com/p/trilinos/wiki/BuildScript
#
# =============================================
#   Requirements:
#       Install python Numpy and Scipe
# =============================================
#

# Trilinos Source location
SOURCE_BASE="/root/trilinos-11.10.2-Source/"

# Path to build source
BUILD="$SOURCE_BASE/Build"

# Installation path:
PREFIX="/shared/trilinos/11.10.2/"

# Third party software installation paths
MPI_HOME="/shared/gcc/4.4.7/openmpi/1.8.1"
MPI_BIN="${MPI_HOME}/bin"
MKLROOT="/shared/lib/lib64"
NETCDF="/shared/netcdf"
```

```
cmake28 \
-D CMAKE_INSTALL_PREFIX="${PREFIX}" \
-D Trilinos_INSTALL_INCLUDE_DIR:PATH="${PREFIX}/include" \
-D Trilinos_INSTALL_LIB_DIR:PATH="${PREFIX}/lib" \
-D Trilinos_INSTALL_RUNTIME_DIR:PATH="${PREFIX}/bin" \
-D CMAKE_BUILD_TYPE:STRING=RELEASE \
-D TPL_ENABLE_MPI:BOOL=ON \
-D MPI_BASE_DIR:FILEPATH="${MPI_HOME}" \
-D MPI_EXEC:FILEPATH="${MPI_BIN}/mpiexec" \
-D MPI_Fortran_COMPILER:FILEPATH="${MPI_BIN}/mpif90" \
-D MPI_CXX_COMPILER:FILEPATH="${MPI_BIN}/mpicxx" \
-D MPI_C_COMPILER:FILEPATH="${MPI_BIN}/mpicc" \
-D CMAKE_CXX_FLAGS:STRING="-ansi -Wall" \
-D BLAS_LIBRARY_DIRS:FILEPATH="${MKLROOT}" \
-D BLAS_LIBRARY_NAMES:STRING="mkl_rt" \
-D LAPACK_LIBRARY_DIRS:FILEPATH="${MKLROOT}/" \
-D LAPACK_LIBRARY_NAMES:STRING="mkl_rt" \
-D Netcdf_INCLUDE_DIRS:FILEPATH="${NETCDF}/include"\
-D Netcdf_LIBRARY_DIRS:FILEPATH="${NETCDF}/lib"\
-D Netcdf_LIBRARY_NAMES:STRING="netcdf"\
-D Trilinos_ENABLE_OpenMP:BOOL=ON \
-D BUILD_SHARED_LIBS:BOOL=ON \
-D Trilinos_ENABLE_TESTS:BOOL=ON \
-D Trilinos_ENABLE_DEBUG:BOOL=ON \
-D Trilinos_SHOW_DEPRECATED_WARNINGS:BOOL=OFF\
-D Trilinos_ENABLE_ALL_PACKAGES:BOOL=ON \
-D Trilinos_ENABLE_SECONDARY_STABLE_CODE:BOOL=OFF\
-D Trilinos_ENABLE_PyTrilinos:BOOL=ON \
$EXTRA_ARGS \
${SOURCE_BASE}
```

Trilinos installation

- A web-based interface to Trilinos
- Most of Trilinos from the Web

  Teuchos, **Epetra,** EpetraExt, Galeri
  AztecOO, Amesos, IFPACK, ML,
  PyTrilinos

- XML, PHP, PyChart
- C++ module, which allows the programs
- MPI hack, web-socket

- HTML5 – SVG, WebGL

Install Pychart

```
Untar,  CD into folder
# python setup.py install
Testing
        # python ./demo/date.py > ../../date.pdf
        # evince ../../date.pdf
```

Install Apache/httpd

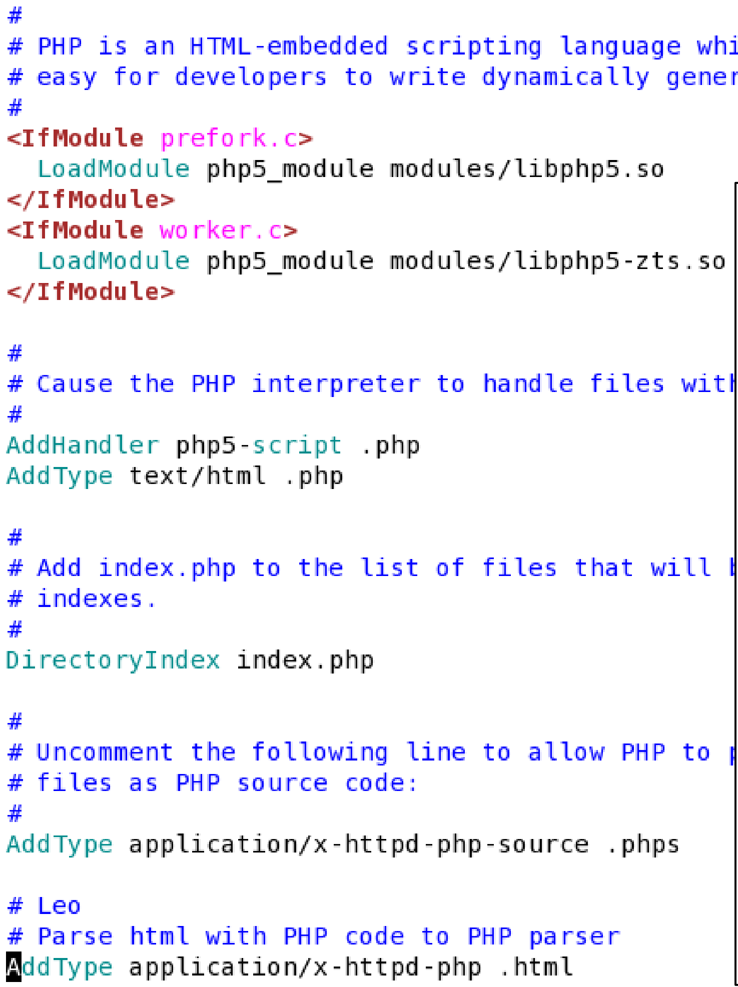
Install PHP

Installing the Web interface

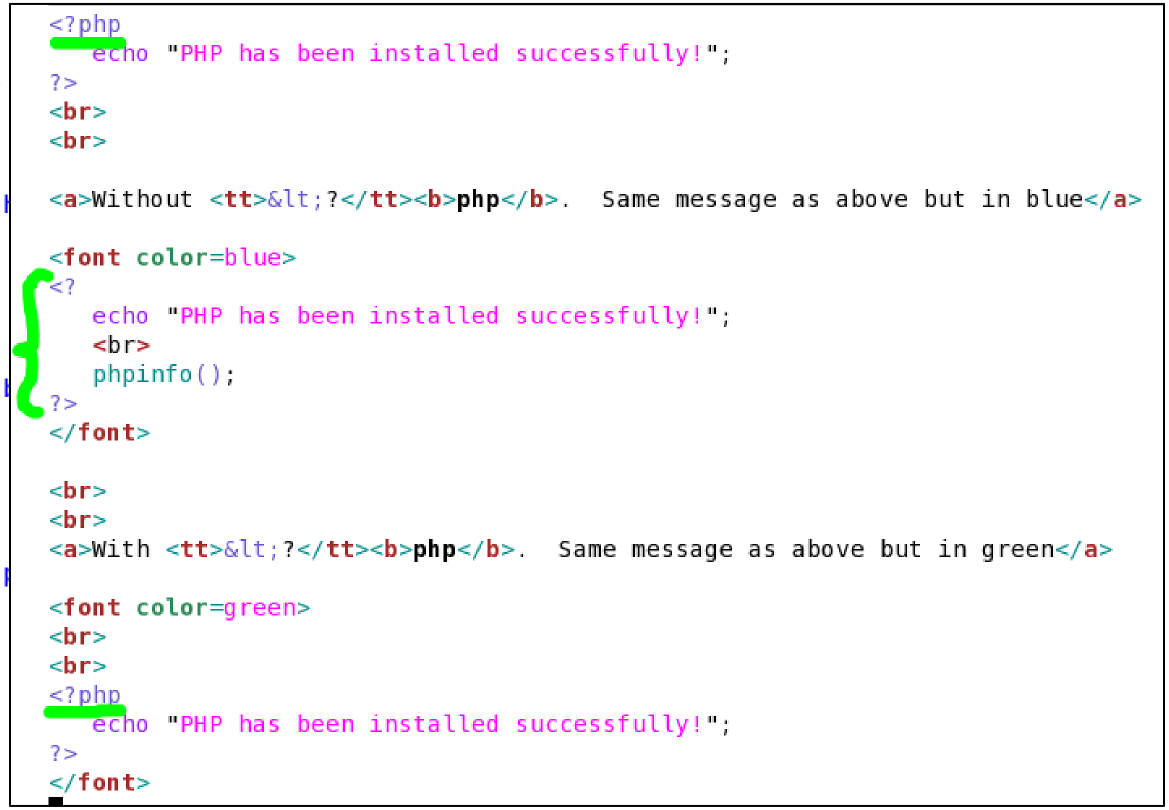
- Trilinos install by default the minimal WebTrilinos requirements

```
        # vi Build-Webtrilinos
        #!/bin/bash

        # Third party software installation paths
        MPI_HOME="/shared/gcc/4.4.7/openmpi/1.8.1"
        MPI_BIN="${MPI_HOME}/bin"
        MKLROOT="/shared/lib/lib64"
        NETCDF="/shared/netcdf/4.3.2"


./configure --prefix=/shared/WebTrilinos --enable-mpi --enable-tests --enable-webtrilinos-tests --enable-webtrilinos-examples --with-mpi-compilers="${MPI_BIN}"

# sh Build-Webtrilinos
```

Apache+PHP

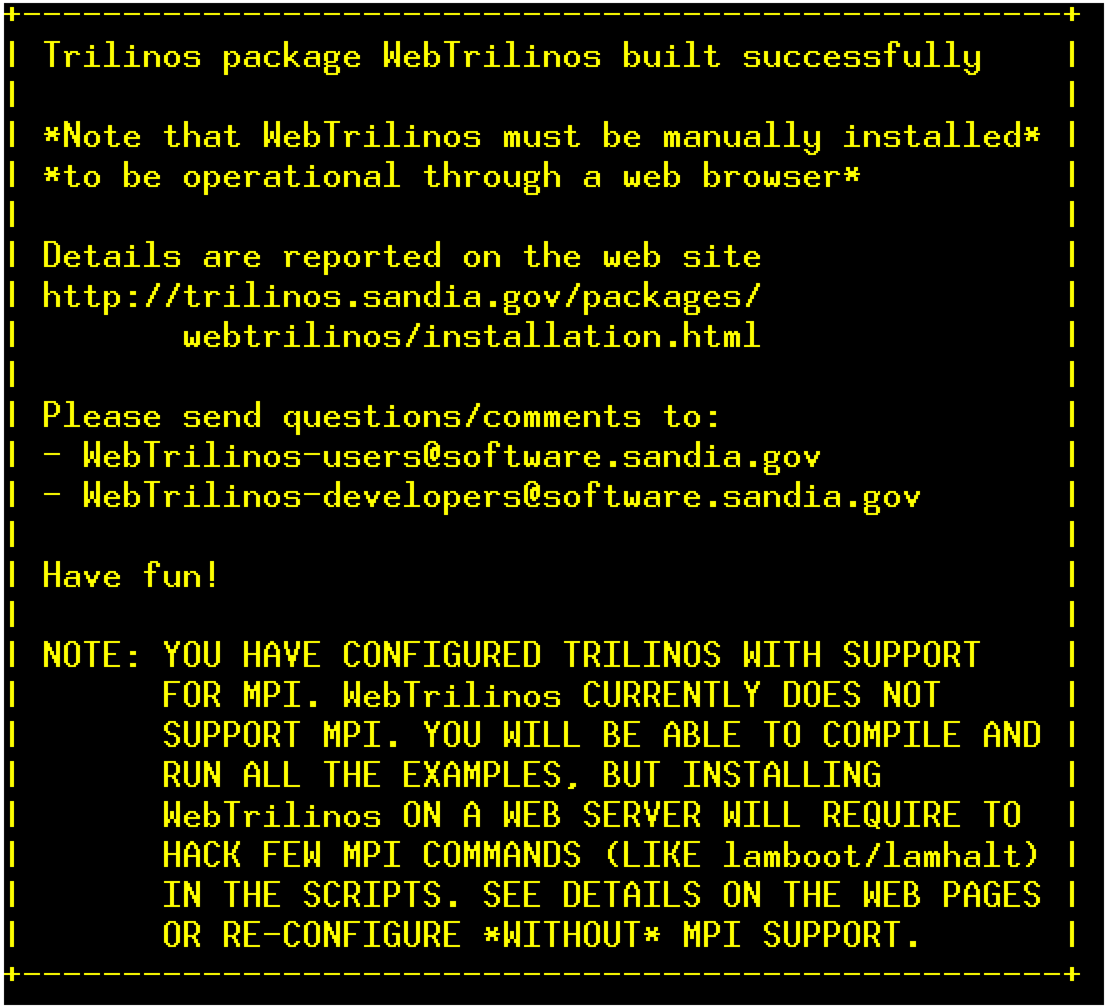


```
#
# PHP is an HTML-embedded scripting language which attempts to make it
# easy for developers to write dynamically generated webpages.
#
<IfModule prefork.c>
  LoadModule php5_module modules/libphp5.so
</IfModule>
<IfModule worker.c>
  LoadModule php5_module modules/libphp5-zts.so
</IfModule>

#
# Cause the PHP interpreter to handle files wit
#
AddHandler php5-script .php
AddType text/html .php

#
# Add index.php to the list of files that will b
# indexes.
#
DirectoryIndex index.php

#
# Uncomment the following line to allow PHP to p
# files as PHP source code:
#
AddType application/x-httpd-php-source .phps

# Leo
# Parse html with PHP code to PHP parser
AddType application/x-httpd-php .html
```



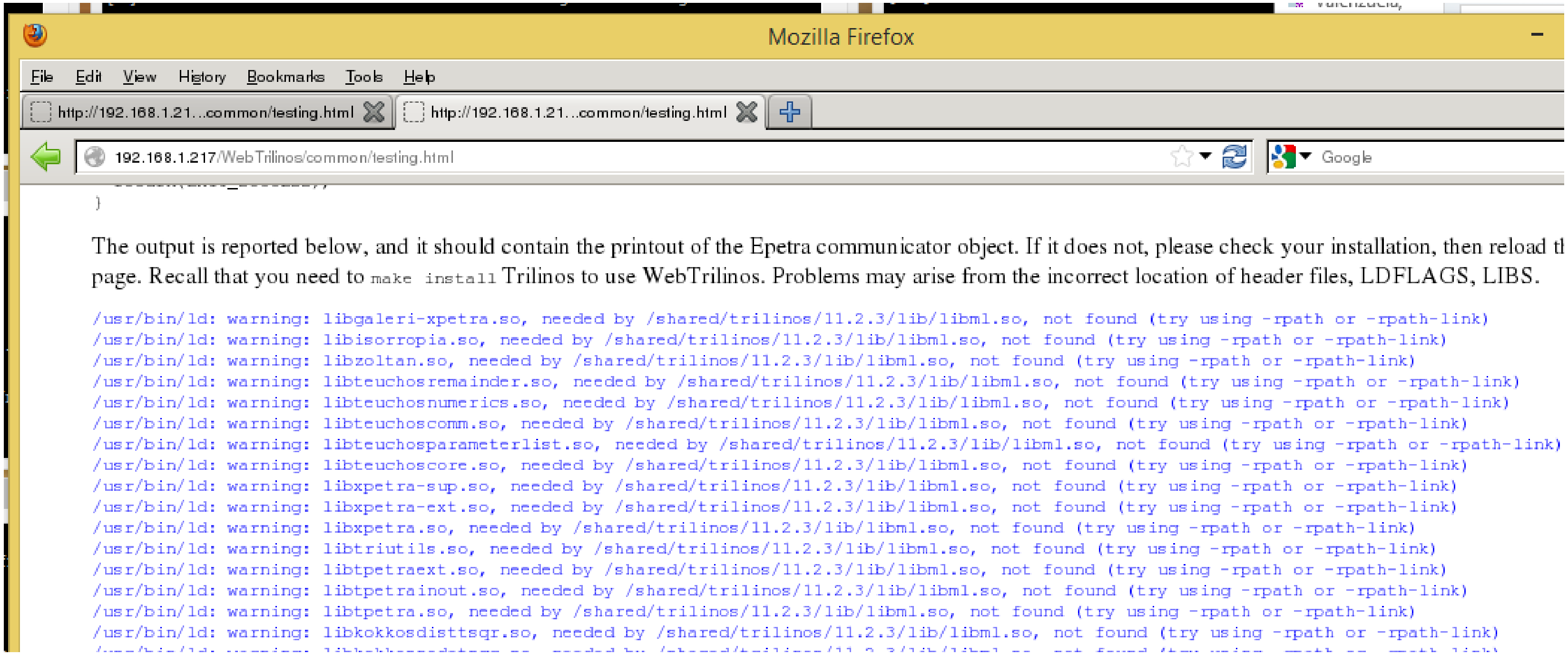
```
<?php
   echo "PHP has been installed successfully!";
?>
<br>
<br>

<a>Without <tt>&lt;?</tt><b>php</b>.  Same message as above but in blue</a>

<font color=blue>
<?
   echo "PHP has been installed successfully!";
   <br>
   phpinfo();
?>
</font>

<br>
<br>
<a>With <tt>&lt;?</tt><b>php</b>.  Same message as above but in green</a>

<font color=green>
<br>
<br>
<?php
   echo "PHP has been installed successfully!";
?>
</font>
```

Web-Trilinos



```
+-----------------------------------------------------+
| Trilinos package WebTrilinos built successfully     |
|                                                     |
| *Note that WebTrilinos must be manually installed* |
| *to be operational through a web browser*          |
|                                                     |
| Details are reported on the web site                |
| http://trilinos.sandia.gov/packages/                |
|        webtrilinos/installation.html                |
|                                                     |
| Please send questions/comments to:                  |
| - WebTrilinos-users@software.sandia.gov             |
| - WebTrilinos-developers@software.sandia.gov        |
|                                                     |
| Have fun!                                           |
|                                                     |
| NOTE: YOU HAVE CONFIGURED TRILINOS WITH SUPPORT      |
|       FOR MPI. WebTrilinos CURRENTLY DOES NOT       |
|       SUPPORT MPI. YOU WILL BE ABLE TO COMPILE AND  |
|       RUN ALL THE EXAMPLES, BUT INSTALLING          |
|       WebTrilinos ON A WEB SERVER WILL REQUIRE TO   |
|       HACK FEW MPI COMMANDS (LIKE lamboot/lamhalt)  |
|       IN THE SCRIPTS. SEE DETAILS ON THE WEB PAGES  |
|       OR RE-CONFIGURE *WITHOUT* MPI SUPPORT.        |
+-----------------------------------------------------+
```

Web-Trilinos: Error!

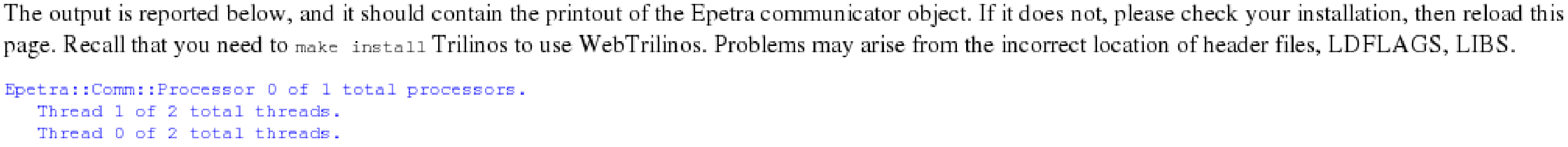

Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://192.168.1.21...common/testing.html
http://192.168.1.21...common/testing.html
192.168.1.217/WebTrilinos/common/testing.html
Google

```
}
```

The output is reported below, and it should contain the printout of the Epetra communicator object. If it does not, please check your installation, then reload tl page. Recall that you need to `make install` Trilinos to use WebTrilinos. Problems may arise from the incorrect location of header files, LDFLAGS, LIBS.

```
/usr/bin/ld: warning: libgaleri-xpetra.so, needed by /shared/trilinos/11.2.3/lib/libml.so, not found (try using -rpath or -rpath-link)
/usr/bin/ld: warning: libisorropia.so, needed by /shared/trilinos/11.2.3/lib/libml.so, not found (try using -rpath or -rpath-link)
/usr/bin/ld: warning: libzoltan.so, needed by /shared/trilinos/11.2.3/lib/libml.so, not found (try using -rpath or -rpath-link)
/usr/bin/ld: warning: libteuchosremainder.so, needed by /shared/trilinos/11.2.3/lib/libml.so, not found (try using -rpath or -rpath-link)
/usr/bin/ld: warning: libteuchosnumerics.so, needed by /shared/trilinos/11.2.3/lib/libml.so, not found (try using -rpath or -rpath-link)
/usr/bin/ld: warning: libteuchoscomm.so, needed by /shared/trilinos/11.2.3/lib/libml.so, not found (try using -rpath or -rpath-link)
/usr/bin/ld: warning: libteuchosparameterlist.so, needed by /shared/trilinos/11.2.3/lib/libml.so, not found (try using -rpath or -rpath-link)
/usr/bin/ld: warning: libteuchoscore.so, needed by /shared/trilinos/11.2.3/lib/libml.so, not found (try using -rpath or -rpath-link)
/usr/bin/ld: warning: libxpetra-sup.so, needed by /shared/trilinos/11.2.3/lib/libml.so, not found (try using -rpath or -rpath-link)
/usr/bin/ld: warning: libxpetra-ext.so, needed by /shared/trilinos/11.2.3/lib/libml.so, not found (try using -rpath or -rpath-link)
/usr/bin/ld: warning: libxpetra.so, needed by /shared/trilinos/11.2.3/lib/libml.so, not found (try using -rpath or -rpath-link)
/usr/bin/ld: warning: libtriutils.so, needed by /shared/trilinos/11.2.3/lib/libml.so, not found (try using -rpath or -rpath-link)
/usr/bin/ld: warning: libtpetraext.so, needed by /shared/trilinos/11.2.3/lib/libml.so, not found (try using -rpath or -rpath-link)
/usr/bin/ld: warning: libtpetrainout.so, needed by /shared/trilinos/11.2.3/lib/libml.so, not found (try using -rpath or -rpath-link)
/usr/bin/ld: warning: libtpetra.so, needed by /shared/trilinos/11.2.3/lib/libml.so, not found (try using -rpath or -rpath-link)
/usr/bin/ld: warning: libkokkosdisttsqr.so, needed by /shared/trilinos/11.2.3/lib/libml.so, not found (try using -rpath or -rpath-link)
```

Adding the /shared/trilinos/11.2.3/lib to LD_LIBRARY_PATH does not work!
Fix adding ldconfig it does:
Touch /etc/ld.so.conf.d/trilinos-11.2.3.conf
          /shared/trilinos/11.2.3/lib
# ldconfig -v (update the library path)



The output is reported below, and it should contain the printout of the Epetra communicator object. If it does not, please check your installation, then reload this page. Recall that you need to `make install` Trilinos to use WebTrilinos. Problems may arise from the incorrect location of header files, LDFLAGS, LIBS.

```
Epetra::Comm::Processor 0 of 1 total processors.
   Thread 1 of 2 total threads.
   Thread 0 of 2 total threads.
```

Concluding remarks

- Desktop to Supercomputers
- Latest solver capability
- Extreme (Exa?) Scale computing
- Scalable Linear Algebra Themes:

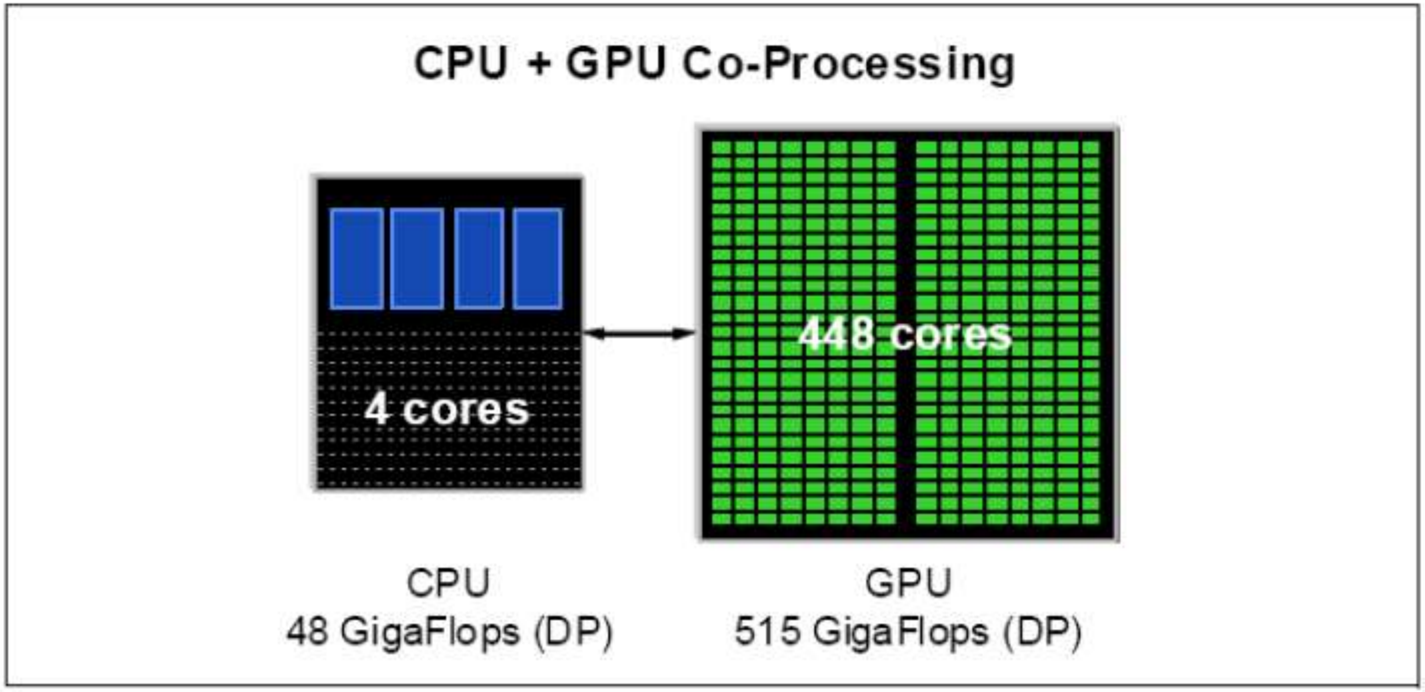
Multicore/GPUs: Pre-requisite for extreme scale.

Multi-precision algorithms.

Cloud based Supercomputing with Touchpad interface?

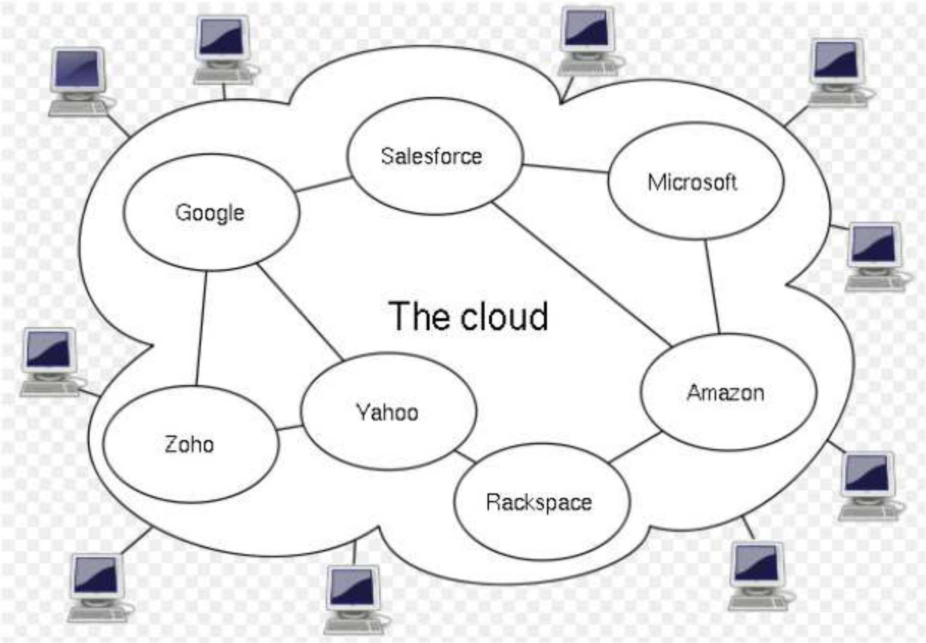
HPC/Cloud/GPU

Future!

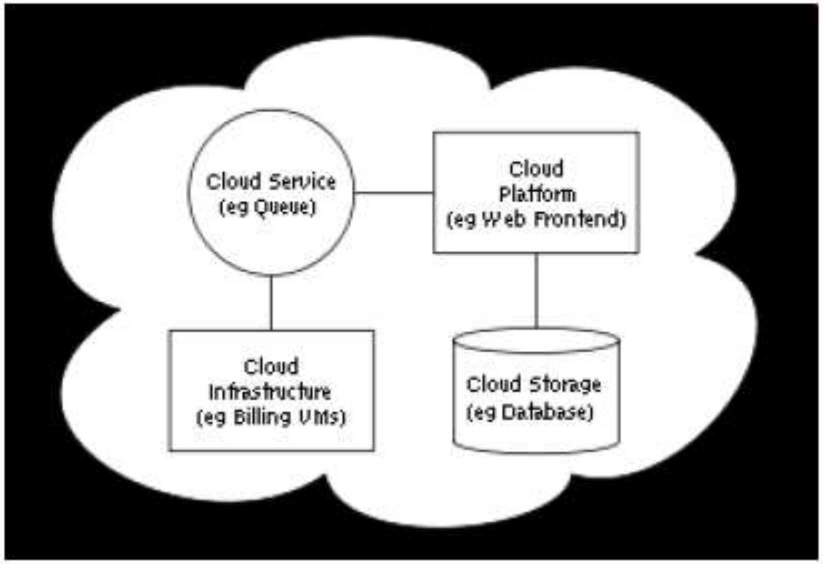


HPC/Cloud/GPU

Future!



Source: http://en.wikipedia.org/wiki/Cloud_computing



- Autonomic Computing
- Client - server model
- Grid Computing
- Mainframe Computer
- Utility Computing
- Peer-to-peer

Client

Application

Platform

Infrastructure

Server