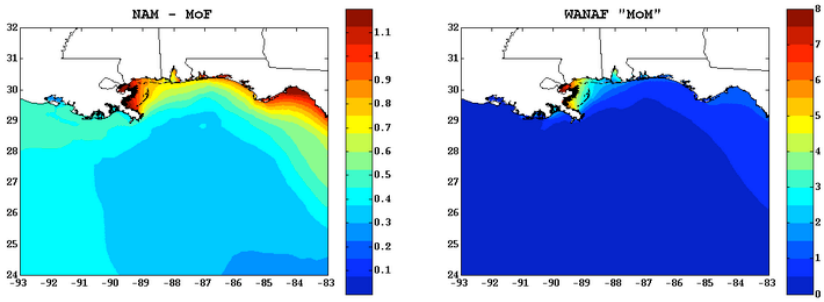
Application Name	SCOOP ADCIRC
Application Area	Ocean Modeling on the Grid
Keywords	
Project/Dept. Affiliation	NC SCOOP Team – UNC Marine Science, Renaissance Computing Institute, MCNC
Value of grids to this application	To increase potential forecast accuracy, it is useful to run storm-surge models with different forcing conditions (i.e., wind fields), all of which constitute an ensemble run. Grid environments are ideal for such simulations.
Originating institution	UNC/MCNC
Contact (s)	Lavanya Ramakrishnan lavanya@renci.org Steve Thorpe thorpe@mcnc.org Howard Lander howard@renci.org
Participating sites	U. Kentucky, Texas A&M U., USC, TACC…
General description	SCOOP stands for the **S**outheastern Universities Research Association's (SURA) **C**oastal **O**cean **O**bserving and **P**rediction program. SCOOP goals include improving predictions of coastal phenomena such as storm-surge driven by extra-tropical cyclones and hurricanes. ADCIRC is a finite element method shallow water model for computing tidal and storm surge water level and depth averaged currents associated with these phenomena. The application consists of a group of ADCIRC simulations driven by an ensemble of wind fields that represent a forecast of a current North Atlantic basin tropical storm or hurricane. The wind fields are being generated by other SCOOP partners (UF, VIMS), and distributed via LDM as "*grib*" files that contain the hurricane pressure and winds. Other wind sources include the National Centers for Environmental Prediction North American Mesoscale forecast run and their hurricane-specific forecast run. When the wind fields arrive through LDM, the ADCIRC workflow is triggered. The workflow consists of an ensemble of ADCIRC runs, with one simulation per wind field input. Note that the workflow is triggered not by human input, but by incoming wind fields for a current North Atlantic storm or hurricane as supplied by our SCOOP partners. Thus in our standard mode of operation, we are unable to predict exactly when or whether the ADCIRC simulations will be pushed onto the grid. We have developed a simple Java API using standard grid tools (such as Globus 3.2.1 and Java Cog) to locate the best resources for the storm-surge simulations, based on availability and current load of the machines. This resource allocation is independent of the actual model and is used to locate resources on the grid (that have the Globus components - gatekeeper, GridFTP and information services available) for forecast and hindcast model runs. Once the appropriate resources are located, the files for each ADCIRC simulation (wind, grid definition, initial conditions, etc.) are moved to the appropriate machine and the run started. The initial conditions for the ADCIRC simulation are extracted from the mass storage archive of historical ADCIRC simulations. When all the ensemble runs are complete, the separate ADCIRC storm surge simulation results may be gathered and a synthesized ensemble solution created (the details of this are not yet determined). The final solution may be pushed for

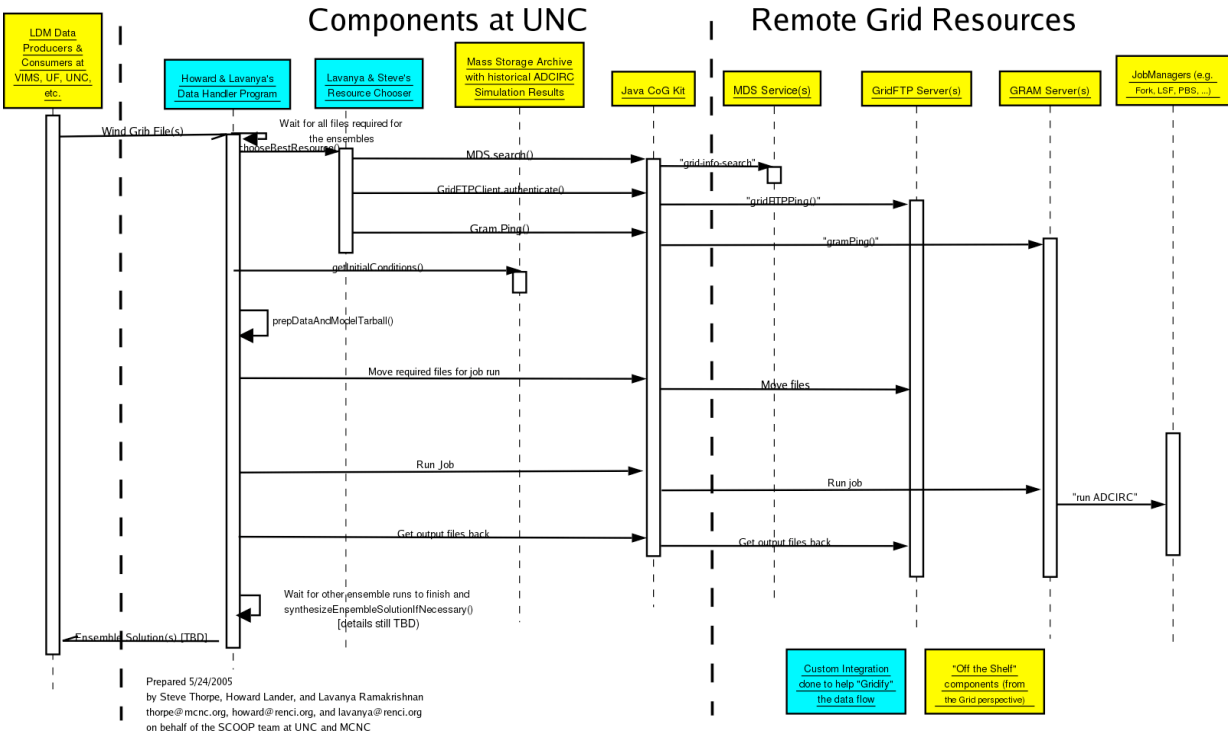
	distribution via LDM to be ingested by the SCOOP Visualization system – see below for example visualization output. Note these images are created by a visualization process that is out of scope of the simulation that is run by this application.  *Left panel: The ADCIRC maximum water level over the 72 h forecast that starts 29 Aug, 2005, driven by the "usual, always-available" ETA winds.* *Right panel: The ADCIRC maximum water level over ALL of the UFL ensemble wind fields and over the 72 h forecast that starts 29 Aug, 2005, driven by the UFL always-available ETA winds.* *POINT: note that the color scales are very different. The ETA-driven (left) max water levels are very small (order 1 m) compared to the ensemble wind-driven water levels (order 8 meters). This improved forecast could be considered one major benefit of the SCOOP project.*

Anticipated system requirements for participation	**Hardware requirements**: X86-based Linux systems, preferably a cluster but would technically work with a single node, albeit much ***much*** slower. Prefer >= 2.0 Ghz CPU and >= 2.0 GB of RAM per node at a minimum, and >= 1 GB of free disk space. **Software requirements**: From the Globus Toolkit, these three "pre-Web Services" pillars that are legacy since GT 2.4 are required: 1. Globus gatekeeper for job management, ideally with a back-end scheduler (see below). 2. GridFTP for file transfer. 3. MDS (desired but optional) for publishing resource information such as CPU characteristics and back end scheduler queue status. Initially, ssh access will be required for testing purposes. It is strongly recommended to use a backend scheduler behind the globus gatekeeper, such as OpenPBS, LSF, SGE, etc., but this is optional. If a backend scheduler is used, it should be MPI enabled (we have used mpich-1.2.5.2 but other versions would very likely work). A FORTRAN compiler will not be required, as the ADCIRC binaries are compiled at UNC prior to being shipped out onto the grid. Java will not be required on the globus resource side, as it is only being used client-side by the components that launch the job onto the grid.

Anticipated non-system requirements for participation	System administrator to be able to test grid connectivity and work through firewall issues
Grid focus (data sharing, computation, access to unique resources, collaboration)	Computation
Network dependencies (bandwidth, latency, multicast, other)	

Sequence Diagram for Grid Use Case #1: The UNC Scenario

As described at http://scoop.sura.org/bin/view/Main/UseCases#Grid_Use_Case_1_The_UNC_Scenario



Submitted SEPTEMBER 9, 2005