Many Task Computing for Modeling the Fate of Oil Discharged from the Deep Water Horizon Well Blowout

A. Srinivasan
Center for Computational Science
and
Meteorology and Physical Oceanography Division
University of Miami
Miami FL, 33149
Email: asrinivasan@rsmas.miami.edu

J. Helgers
Center for Computational Science
University of Miami
Miami FL, 33149
Email: jhelgers@rsmas.miami.edu

C. B. Paris
Applied Marine Physics Division
University of Miami
Miami FL, 33149
Email: cparis@rsmas.miami.edu

M. LeHenaff
Meteorology and Physical Oceanography Division
University of Miami
Miami FL, 33149
Email: mlehenaff@rsmas.miami.edu

H. Kang
Meteorology and Physical Oceanography Division
University of Miami
Miami FL, 33149
Email: hkang@rsmas.miami.edu

V. H. Kourafalou
Meteorology and Physical Oceanography Division
University of Miami
Miami FL, 33149
Email: vkourafalou@rsmas.miami.edu

M. Iskandarani
Meteorology and Physical Oceanography Division
University of Miami
Miami FL, 33149
Email: miskandarani@rsmas.miami.edu

W. C. Thacker
Cooperative Institute of Marine and Atmospheric Science,
University of Miami
and
Atlantic Meteorological and Oceanographic Laboratory
Miami FL, 33149
Email: Carlisle.Thacker@noaa.gov

J. P. Zysman
Center for Computational Science
University of Miami
Miami FL, 33133
Email: jzysman@med.miami.edu

N. F. Tsinoremas
Center for Computational Science
and
Computer Science Department
University of Miami
Miami FL, 33133
Email: ntsinoremas@med.miami.edu

O. M. Knio
Dept of Mechanical Engineering
Johns Hopkins University
Baltimore, MD
knio@jhu.edu

***Abstract*—The Deep Water Horizon well blowout on April 20th 2010 discharged between 40,000 - 1.2 million tons of crude oil into the Gulf of Mexico. In order to understand the fate and impact of the discharged oil, particularly on the environmentally sensitive Florida Keys region, we have implemented a multi-component application which consists of many individual tasks that utilize a distributed set of computational and data management resources. The application consists of two 3D ocean circulation models of the Gulf and South Florida and a 3D oil spill model. The ocean models used here resolve the Gulf at 2 km and the South Florida region at 900 m. This high resolution information on the ocean state is then integrated with the oil model to track the fate of approximately 10 million oil particles. These individual components execute as MPI based parallel applications on a 576 core IBM Power 5 cluster and a 5040 core Linux cluster, both operated by the Center for Computational Science, University of Miami. The data and workflow between is handled by means**

978-1-4244-9705-8/10/$26.00 ©2010 IEEE

of a custom distributed software framework built around the Open Project for Networked Data Access Protocol (OPeNDAP). In this paper, we present this application as an example of Many Task Computing, report on the execution characteristics of this application, and discuss the challenges presented by the many task distributed workflow involving heterogeneous components. The application is a typical example from the ocean modeling and forecasting field and imposes soft timeliness and output quality constraints on top of the traditional performance requirements.

***Keywords*-many task computing, distributed computing, OPeNDAP, Ocean Modeling, Oil Spill Model**

I. Introduction

On April 20, 2010, the Deepwater Horizon (DWH) offshore drilling and extraction unit exploded, causing the riser pipe to rupture and crude oil to flow into the Gulf of Mexico from an approximate depth of 1500 m. Estimates of the discharge rate are highly uncertain, and range between 500 tons to 15000 tons per day, making it one of the largest documented oil pollution in history. The oil discharge continued until July 15 when it was temporarily closed by a cap. Relief wells to allow permanent termination of the flow were completed in mid-August 2010. A typical scenario in such a seabed blowout is that oil and gas are released under pressure which causes intense mixing at depths. Depending on the situation, oil and gas plumes may reach the surface or remain trapped in the water column. In the DWH case, some of the oil has reached the surface while an unknown amount is trapped in the water column. Given sufficient time, the combined actions of weathering and biodegradation can eliminate most of the discharged oil. However, nature does not always have enough time and in a situation where there is a possibility that substantial amount of the dispersed oil is trapped in the water column, oxygen levels might be altered with a potential to cause long-term damage to the ecosystem of the Gulf. Because of the seabed location of the discharge, large-scale numerical simulations are the only available tool for a rapid assessment of the fate and impact of the discharged oil. The computational analysis typically requires multiple models to address a set of interacting processes. Further, numerous realizations of the model solutions are necessary to adequately quantify uncertainty, and form a statistical picture of the fate of oil. Since the DWH incident, we have setup real-time ocean prediction systems for evolving oil fate assessments. This requires execution of a large number of independent tasks within a daily time frame. These tasks execute on distributed set of resources but communicate by file system operations and thus fall under the Many Task Computing (MTC) paradigm [1]. In this paper, we report on this integrated set of large-scale, compute and data intensive simulations to assess the fate of the oil from DWH as an example of Many Task Computing.

The transport and fate of the discharged oil is governed by physical, chemical, and biological processes that depend on oil properties, hydrodynamics, meteorological and environmental conditions. To simulate this process, most oil spill transport models use ocean currents and meteorological winds obtained from operational forecasting centers as inputs to an oil trajectory and fate prediction models. In these models, information on the time varying 3-dimensional ocean state and oil properties are combined to determine the evolution of the oil footprint in the ocean. Although, such simulations are now routine, the DWH incident due to its location and magnitude poses significantly bigger modeling and computational challenges. Early on, one major concern was the possibility of the Loop Current, the principal large scale current system in the Gulf, entraining the discharged oil and transporting it to coastal regions far away in its path such as the Florida Keys. To accurately assess this aspect, the circulation model providing oceanic state information to the oil transport model should be able to represent adequately the interactions between processes at the scale of the entire Gulf and at local scales of Florida Keys. Further, for such studies a circulation model with resolution less than 1 km in the horizontal and 10 m in the vertical is desirable. However, current state-of-the-art ocean forecasting systems provide near-real-time 3D state forecasts at 4 km in the horizontal and 50 m in the vertical for ocean basins like the Gulf of Mexico. Therefore, we decided to increase the resolution to 2 km in our Gulf of Mexico and embed a existing an existing 900 m resolution South Florida and Florida Straits (SFFS) model within this model to focus resolution on the Florida Keys region. Each model is limited by the range of scales and the processes it represents, but the combination is sufficient to address the fate and impact of oil in both coastal and oceanic regions of the Gulf. Currently, these models are undergoing testing and represent the highest resolution information on the 3D oceanic state available for these regions.

In the majority of oil spill simulations, the oil trajectory and fate prediction model is typically only integrated using outputs from a single circulation model. In the present case, however, we are interested in modeling the effects of the discharge in the Northern Gulf of Mexico and its effects on environmentally sensitive regions far removed from the DWH site. This requires dynamically integrating the oil spill trajectory model with both the inner and outer hydrodynamic models. This makes our oil trajectory model more data intensive since it uses multiple time-varying large ocean model fields for a relatively short computation.

In essence our integrated application fits neatly within the MTC paradigm. The application involves many components, each composed of tightly coupled MPI tasks. The components themselves are loosely coupled applications, but have dependencies among them, and use data files residing on geographically distributed machines for inter-process communication.

In section 2 we describe the application and the various components. Following this, in section 3, we present the execution characteristics of the various components on our local machines, and discuss the application as an example of MTC. Section 4 presents some sample results from our simulations. We conclude by providing a summary and discussion in Section 5.

II. The Oil Spill Application and its Components

Our application combines data from high resolution circulation models with an oil fate and trajectory model to simulate the spreading, dispersal and degradation of oil within the water column. Our goal is to run a large number of oil fate prediction forecasts to obtain a statistical picture of the oil fate and its possible impact on environmentally sensitive Florida Keys area. Below we describe the individual components and their computational aspects.

A. The Ocean Circulation Model

We use the HYbrid Coordinate Ocean Model (HYCOM) for our application. HYCOM is a state-of-the-art ocean model used by the US Navy and NOAA for operational ocean forecasting. It belongs to the class of Ocean General Circulation Models (OGCM), and solves the hydrostatic Navier-Stokes equations (primitive equations) on a finite difference grid as applied to a thin layer of stratified ocean on a rotating Earth. HYCOM's distinguishing feature is a generalized vertical coordinate system that optimizes the distribution of vertical computational layers making it simultaneously suitable for application domains that include both the open ocean and shallow or unstratified regions [2], [3]. This is an important consideration in our choice of the ocean model since our domain of interest includes both open ocean and coastal regions. The model also includes sophisticated parameterizations for unresolved process and is equipped with a suite of advanced data assimilation schemes. Further, architectural details can be found in [2], and details on model performance and validation can be found in [4]. An exhaustive list of model capabilities and relevant technical details are also available in the HYCOM Users Manual (www.hycom.org) and the references therein. Our model for the Gulf of Mexico is embedded within a larger operational data-assimilative global HYCOM model that provides near real-time information on sea water velocity and water properties such as temperature, salinity and density at the boundaries of our domain. This information is available via OPeNDAP from a dedicated data server that hosts about 140 TB of Global Ocean HYCOM data.

1) Computational Aspects: HYCOM is written in a mix of Fortran-77 and Fortran-90, and is maintained as a single code base. It has a wide user base and is well tested on numerous architectures. Over the years it has undergone modifications to enhance performance. The model is parallelized for distributed memory machines using MPI, and for shared machines using OpenMP. It can also be used in an hybrid MPI+OpenMP fashion for clusters with multicore nodes. The distributed memory parallelization uses the Single Program Multiple Data (SPMD) paradigm, and is implemented by means of 2 dimensional domain decomposition that ensures good load balance. The parallelized code avoids global reductions in order to produce bit for bit reproducible results on any number of processing cores. The code exhibits good scalability upto several thousands of cores. A Global HYCOM model at 3-8 km resolution is operational at the Naval Research Lab and is routinely run on O(1000) cores. The current official release of HYCOM use Fortran binary files with zero padding to make the record size an integer multiple of the block size. We however use a modified version of HYCOM which uses NetCDF for IO.

B. The Oil Spill Model

The oil spill model is a recently developed 3D transport code that predicts the dispersal of oil discharged in the marine environment. The model represents the discharged oil as particles and predicts the dispersal based on 3D currents and mixing parameters obtained form ocean circulation model. At each time step a specified number of particles are released at the source. Particles are released at a depth of 1200 m where the plume of mixed oil and gas is estimated to spilt into individual oil droplets and gas bubbles (pers Comm Pete Cayragher, BP geologist). Each particle represents a fraction of the mass of the oil released and the number of particles released is adjusted so that each particle represents about 1 kilogram of oil. The hydrocarbon fractions and the droplet size distribution at the source are two crucial parameters affecting the time varying footprint of oil in the water column. The model assumes that the oil is composed of three fractions, light, medium and heavy. The droplets (particle) sizes are assigned randomly between an assumed minimum and maximum value. In this study, as we are interested in studying the long-term budget of oil depending on its ability to stay at depth, we only modeled oil droplets with diameter ranging from 1 to 300 microns. This range allows studying particles that will remain at depth, as well as those that will raise rapidly to the surface. Particles larger than 300 microns are considered to reach the surface so quickly that their size does not influence their fate. The model also includes the effects of evaporation and degradation due to physical and biological processes. These effects are parameterized based on the specified half-life times. Only particles at the surface are subject to evaporative decay while particles at all depths are subject to degradation.

1) Computational Aspects: The oil model is written in a mix of Fortran-90/C++ and is also parallelized for distributed and shared memory architectures. The parallelization uses a task pool approach. The master process distributes the particles equally amongst the specified number of tasks at the beginning of the simulation. Additional tasks from the pool are assigned to slave processes as they each finish their initial allocations. Typically, each task has a copy of the flow and water properties fields from the circulation model. The time history of oil particles is evolved passively with no interaction between particles. This allows us to drastically reduce the communication requirements. However, when large physical domains or high resolutions are involved the flow and water property fields may not fit within the local memory so the flow field is also domain decomposed. This requires intercommunication whenever particles cross from one domain to another. In this case, an SPMD approach is used instead of the master slave approach and the particles are periodically redistributed to ensure effective load balance.

The model is designed to handle flow and water property fields from multiple circulation models simultaneously and choose the appropriate fields for updating the fate of the oil. This makes the model data intensive since it uses a large amount of data for relatively short computations. The model is OPeNDAP enabled and uses a Just-In-Time approach to access remote data from distributed sources.

C. Uncertainty Quantification

Simulations of the spreading and fate of the oil are dependent on numerous input parameters, such as discharge rate, droplet size distribution, oil properties most of which are not well known. By running a large number of simulations we can expect to obtain a statistical picture of the predicted fate and influence areas. Typically, such analyses use a Monte Carlo approach which is a form of complete enumeration of parameters. This usually requires a large number of model runs to adequately sample the parameter range and convergence is often slow. Much work on uncertainty quantification has focused on reducing the number of model runs and obtaining faster convergence. In recent years, a spectral formalism based on Wieners [5] concept of homogeneous chaos has been shown to be efficient in propagating and quantifying uncertainties [6], [7]. In contrast to Monte Carlo analysis, this method quantifies both the extent and dependence of uncertainties in the model solution to specific input uncertainties. Here we use this approach for uncertainty analysis. The method consists of representing the uncertain inputs as functions of random variables of a known or assumed probability density function. The model solution is then expanded as a polynomial series in terms of suitable orthogonal basis functions associated with the probability of the inputs. The model output is then available as a constant, plus a linear combination of the inputs followed by higher order non-linear terms involving products of the inputs. The coefficients of the expansion are then computed by an ensemble calculation at carefully chosen points.

1) Computational Aspects: The computational cost of this method is determined by the number of model runs required to evaluate the coefficients of the polynomial representing the solution. In practice, the polynomial series representing the solution, is truncated to P terms. The value of P is determined by the number N of input random variables (uncertain parameters) and the highest degree, K, of polynomial used to represent the solution by the formula:

$$P = \frac{(N+K)!}{(N!K!)} - 1 \qquad (1)$$

Model runs at judiciously chosen values of input parameters then allow efficient evaluation of the polynomial coefficients. Obtaining model solutions realizations can be expensive but the simulations can be sped up significantly as they are independent.

III. Oil Fate Prediction as an MTC Application

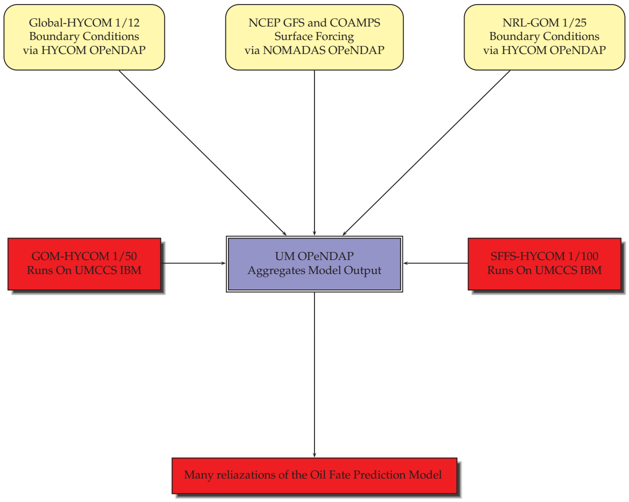

Fig. 1. Components of the Oil Spill Application. The core computational tasks in the application are shown in red. The data management is through the OPeNDAP server shown in blue. Multiple realizations of the oil model are run using circulation data from the SFFS-HYCOM and one of the Gulf of Mexico models.

We have two ocean models, one for the Gulf of Mexico (GOM-HYCOM), and a second one for South South Florida and Florida Straits (SFFS-HYCOM). In addition, we also use outputs from a lower resolution operational Gulf of Mexico forecasting system while we transition to the GOM-HYCOM. The data from these models are integrated with an oil model to accomplish our objective of predicting the fate of oil and quantifying the uncertainty in our predictions. Towards this end many loosely coupled tasks with data dependencies have to be executed regularly and the workflow between them managed. The overall application, the various components and data sources are illustrated in Figure 1. The three components used in our application are independent MPI based parallel programs that generally require substantial computational resources. However, they have differing computational demands in terms of CPU, memory, interconnect and data requirements. For example, the ocean circulation models are best performing on systems with low-latency and high bandwidth interconnects while the oil model is latency tolerant and can use a Gigabit Ethernet as the interconnect. This heterogeneity in the application components provides flexibility in adapting the application to different types of computational resources that are available. Our aim is to make optimal use of the available computational resources which in the present case is a 5040-core, Xeon/Opteron based IBM BladeCenter cluster with Gigabit Ethernet as interconnect, and a 576-core IBM p5-575 cluster with low latency, high bandwidth HPS interconnect. Below, we discuss the computational requirements for the individual components, data management and workflow aspects of the application.

A. Requirements and execution characteristics of the GOM-HYCOM Component

The GOM-HYCOM application is a moderate sized ocean circulation model. The model state vector is evolved on a 1073x769x26 grid with a time step of 2 minutes for the slow processes and a 10 second time step for the fast processes. In our application, we need to execute this model daily with forcing information from atmospheric and ocean forecasting systems. When run in real-time mode, each run will produce output for 10 days, from 4 days in the past to 5 days in the future. This configuration requires 28 GB of memory and 2 GB of data input for every model day. Further, it produces 4 GB of output for every model day. Efficient use of this model for daily ocean forecasting requires O(100) cores for about an hour daily. Due to the domain decomposition and numerous halo update operations the configuration requires a low latency and high bandwidth interconnect. For our application, we run this model on the IBM P575 cluster on 64 cores.

B. Requirements and execution characteristics of the SFFS-HYCOM Component

The SFFS-HYCOM application is a very-high resolution, limited area ocean circulation model. The model state vector is evolved on a 437x361x26 grid which is not big by current standards. However, due to the high resolution a very short time step of 20 seconds is required which makes the overall computation expensive. Further, due to the small size of the domain the optimal number of processors to decompose the domain is relatively small. In our application, we execute this model daily with forcing information from atmospheric and ocean forecasting systems. Each run produces output for 10 days, from 4 days in the past to 5 days in the future. This configuration requires 4 GB of memory and 0.5 GB of data input for every model day. Further, it produces 0.8 GB of output for every model day. Efficient use of this model for real-time ocean forecasting requires a few tens of cores. We normally run this model on single Linux nodes with 8 or more cores.

C. Requirements and execution characteristics of the Oil Fate Model

When run in a real-time mode, each run of the oil model tracks approximately 25K particles. These particles are passively evolved with no intercommunication requirements. Therefore, this model is deployed on the Linux Cluster with GigE interconnect and scales well (Figure3). Further, in real time model the flow and water property fields needed for trajectory and fate prediction is on the order of a few tens of GB and can be replicated to all nodes. In this case the daily computations are quite fast and can be accomplished in less than an hour with a few cores by using a pool of tasks approach. However, when used in the hindcast mode the model is substantially costly since on the order of 10 million particles are released. In addition, the model is also integrated for longer duration which requires the corresponding flow and water property information for that time period. This dataset can be several hundred gigabytes and may it may not be possible to load more than a few days of data into memory. The model then becomes IO bound since it has to load two set of state vectors from the ocean models frequently.

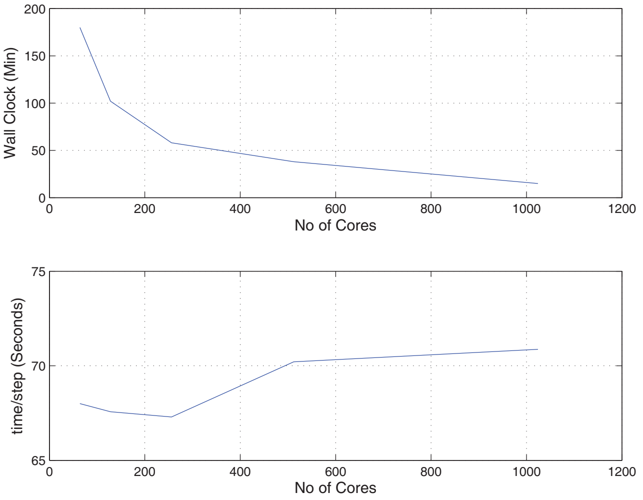

Fig. 2. Performance of the Oil Model on Linux Cluster with 2.5 GHZ Xeons and Gigabit Ethernet Interconnect. The bottom panel illustrates the system performance under varying loads

D. Data Flow and Management

We use the Data Access Protocol (DAP) extensively for moving data between applications. All the component tasks in our application are DAP enabled by either linking to OPeNDAP libraries or by using DAP enabled NetCDF libraries. Further, a custom interface to the OPeNDAP server allows us to regrid and reduce complexities involved in the use of multiple-models. All data is stored on an Opteron based Linux machine with 16 cores, 32 GB of memory and 20 TB RAID 5 disk which runs a THREDDS/OPeNDAP server. We chose OPeNDAP based data management because it provides easy means to aggregate individual files and adequately describe the datasets with meta-data thus simplifying the problem of handling several hundred individual files. The performance implications of using a single OPeNDAP server for reads in our application was considered in some detail. While the OPeNDAP server might not be able to handle hundreds of requests, the performance is quite stable for up to 20 concurrent requests. For example, a single downloading request of 10 GB of data takes about 10 minutes. This rate is stable up to 20 concurrent requests after which the performance starts to fall. Therefore we use a single OPeNDAP request to pre-stage the data instead of allowing each task to directly access the OPeNDAP server. The data are typically downloaded to a Global File System from where it can be moved to local disks if necessary. The output data are then pushed to this Linux machine hosting the THREDDS/OPeNDAP for the next set of runs. Because of the sequential nature of the execution and the slight staggering of the batch runs, the number of concurrent data transfer attempts is kept to a effective level. There are other alternatives such as pulling the data which might be appropriate for other scenarios.

E. Workflow

The major tasks in this MTC application include executing numerous instances of the two ocean circulation models and the oil model. Apart from these tasks, a second set of tasks are needed to handle the data flow and processing operations necessary for each of the major tasks. We make extensive use of scripting to manage the necessary data and the workflow. The tasks comprising the application are executed sequentially using multiple scripts, but are managed by a master script. All our scripts are currently written in Perl. The master script is designed to loop through the components to launch one or more instances of the components. At present we have only a single instance of the ocean models but multiple (36) instances of the oil model. The workflow control loop of the master script is illustrated in Figure 3. When run in the near real-time mode, first the input data required for circulation models is acquired from remote OPeNDAP servers just-in-time by the script. This take anywhere between 15 to 30 minutes depending on the responsiveness of the HYCOM data server. The downloads are typically problem free although some times remote data download fails because the hycom data server has not been updated for the day or if there is a system related problem. These problems occur infrequently but can persist for a few days. If the remote data download fails the model runs proceed using climatological data as backup. Next the script launches the jobs on specified clusters. The outputs are pushed to a filesystem visible on all the clusters via OPeNDAP. The script then updates the local OPeNDAP server which hosts the circulation model data for input to the oil model. Finally the script launches the oil model on the Linux cluster. The outputs from the oil model are again hosted on the local OPeNDAP server for further analysis. We are fine tuning our workflow currently, and anticipate that the entire process will be automated end-to-end.

There are no particular problems when scheduling the application for hindcast runs where timeliness is not paramount. However the situation is different for near-real time runs. Since the resources are shared, job execution is not guaranteed when the machine load is high. Further, in our case, we have multiple models that need to be executed within time constraints and resource constraints. The oil models can be executed in any of the available clusters while the circulation models require machines with high performance interconnects. We experienced an average delay time of about 4 hrs for the large ocean model job. This is the first task in the sequential set and a delay here is problematic as the useful time-window for forecasts might have passed. Therefore we made use of advanced reservations which allowed us to run on 64 cores daily without delays. A more sophisticated approach might be to use intelligent job migration facilities to migrate or suspend low priority jobs for meeting the time constraints for near real-time runs.

The oil-model tasks are essentially independent and can be launched as many single jobs on the Linux cluster. However the queuing and dispatching overheads and the prioritization of large parallel jobs penalize such single job submissions and results in delays. We have essentially packaged such single into a large MPI job to make best use of the resources. This works well in our case as the jobs are homogeneous and can be easily structured to achieve good load balance.

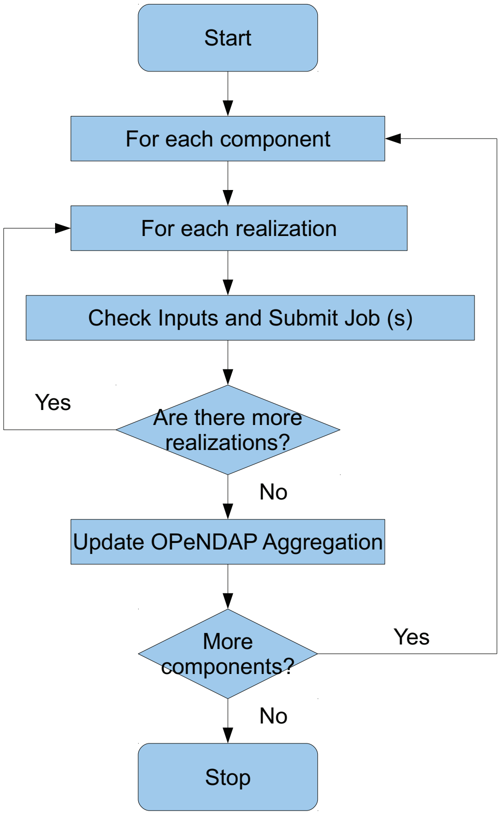
Start
For each component
For each realization
Check Inputs and Submit Job (s)
Yes
Are there more realizations?
No
Update OPeNDAP Aggregation
More components?
Yes
No
Stop

Fig. 3. The application workflow implemented in the master script. The script executes the components sequentially, checking for appropriate inputs, submitting jobs, and updating the OPeNDAP server before proceeding to the next component.

TABLE I
CHARACTERISTICS OF THE COMPONENTS AND THE MTC APPLICATION WHEN RUN IN NEAR-REAL-TIME

Component	No of Tasks	Avg. Time and Data I/O per Run
GOM-HYCOM	64	200 (min) 20 / 40 (GB)
SFFS-HYCOM	8	300 (min) 5 / 8 (GB)
Oil-Model	12000	30 (min) 50 / 14 (GB)
Entire Application	12100	600 (min) 75 / 60 (GB)

The time to execute the various components and the MTC application along with the breakdown of the data transferred between the steps in the pipeline is listed in Table 1.

IV. Sample Results for the MTC Application

We deployed the integrated application described above to estimate the fate of the discharged oil and its possible impact on the Florida Keys region. Three hourly data on the ocean state from an operational GOM forecasting system and the

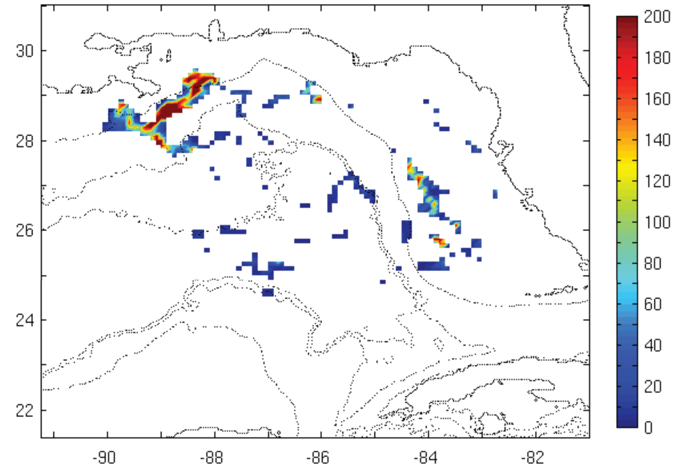

Fig. 4. Surface slick simulated by the oil model for June 11. The simulated slick compares well with satellite based composites for June 11 (not shown)

SFFS-HYCOM models for the April-20th to July 15th timeframe were hosted and regularly updated on an OPeNDAP server. Before each run of the oil model, the necessary input data were downloaded Just-In-Time from the OPeNDAP server by oil model. In total, 36 runs of the oil model, each on 1024 cores were used to quantify the uncertainty in the results due to uncertainties in the inputs. The application used 25K cpu hours and 1.2 TB of data. Sample results of the analysis are illustrated in Figure 4 and 5. The model simulation of the surface slick (Figure 4) is in good agreement with satellite composites for that time period.

Further, the results also suggest that about 80 % of the oil has been removed from the water column by evaporation and degradation(Figure 5). The uncertainty analysis suggests a 20 % uncertainty in this estimate. A more complete analysis of the results is currently underway. Additional simulations including the effects of winds, tides and including other physical processes such as emulsification and sedimentation are planned or underway.

V. Conclusion and Future Work

The paper described some of the challenges and opportunities to develop an integrated oil fate prediction application as a response to the DWH accident. We discussed the many independent tasks that comprise the application, the workflow and data management strategy and showed that the application fits the definition of Many Task Computing. We also described the execution characteristics and performance results obtained using the locally available computational resources. There are several sophisticated scientific workflow engines such as Kepler, Taverna, Swift and others that enable a user to put together a complex application based on a diverse set of components. However, as with any piece of software there is a learning curve associated with using it. Considering the time-critical nature and quasi-operational nature of study and the availability of substantial local resources, we utilized a simple script based solution to the workflow.

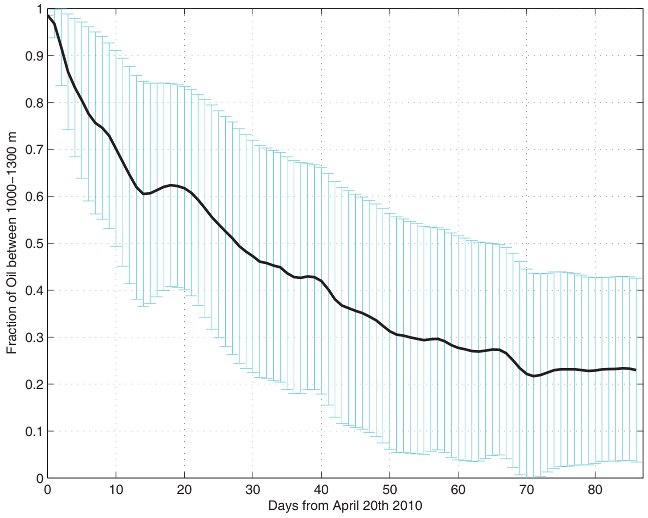

Fig. 5. The subsurface fraction of oil between 1000-1400 m as a function of time. The error bars are based on input uncertainties on the oil droplet size and the oil composition

We are currently extending our hindcast runs to better quantify uncertainty by including more uncertain parameters. Increasing the number of uncertain parameters increases the number of required tasks exponentially. For example, increasing the number of uncertain parameters in the oil model from 2 to 5 increases the number of runs from 36 to at least 256. This will increase the number of tasks from 40000 to 260000. Such a large number of tasks will require resources such as the Teragrid and a more robust scientific workflow solution might be appropriate. We are exploring ways to tune our workflow and anticipate migrating to Teragrid shortly.

Acknowledgment

The research reported here is supported by a NSF-RAPID grant (NSF OCE-1048697) and ONR Award (N00014-101-0498).

References

[1] I. Raicu and I. Foster and Y. Zhao, *Many Task computing for Grids and Supercomputers*, Proceedings of IEEE Workshop on Many Task Computing on Grids and Supercomputers(MTAGS08)”

[2] R. Bleck *An oceanic general circulation model framed in hybrid isopycnal-cartesian coordinates*, Ocean Modeling, **4** (2002), 55-88.

[3] N. G. Winther and G. Evensen *A hybrid coordinate ocean model for shelf sea simulation*, Ocean Modeling, **13** (2006), 221-237.

[4] E. P. Chassignet and L. T. Smith and G. R. Halliwell and R. Bleck, *North Atlantic simulations with the hybrid coordinate ocean model(HYCOM): Impact of vertical coordinate choice, reference pressure and thermobaricity.*, Journal of Physical Oceanography, **33** (2003), 2504-2252.

[5] N. Wiener, *The homogeneous chaos*, American Journal of Mathematics **60** (1938), 897-936.

[6] R. G. Ghanem and P. D. Spanos, *Stochastic Finite Elements: A Spectral Approach*, Springer-Verlag, New York, (1991).

[7] O. P. Le Maître and O. M. Knio, *Spectral Methods for Uncertainty Quantification*, Springer-Verlag, (2010).