Para 2010 – State of the Art in Scientific and Parallel Computing – extended abstract no. 60
University of Iceland, Reykjavik, June 6–9 2010
http://vefir.hi.is/para10/extab/para10-paper-60.pdf

Interactive Weather Simulation and Visualization on a Display Wall with Many-Core Compute Nodes

Bård Fjukstad*, Tor-Magne Stien Hagen†, Daniel Stødle‡, Phuong Hoai Ha§, John Markus Bjørndalen¶, and Otto Anshus‖

Department of Computer Science, Faculty of Science and Technology, University of Tromsø, Norway

Abstract

The task of Weather forecasting requires visualization of large data volumes. Numerical Weather Prediction models (NWP) are essential tools, and are typically run in batch systems for predetermined areas where data is available only at the end of each complete run . The NWPs are often run for as large an area as the available computer resources allow, this limits the resolution and area covered.

We report on a system using a high-resolution, wall-sized tiled display where a user can select a region of interest, and have an NWP model run on-demand for the selected area. The NWP is only run for the requested areas, and may therefore also be run with high spatial resolution. Visualizing these large datasets using a display wall enables the user to view high level of detail for a larger area than regular displays.

To enable interactive use of NWP, we make extensive use of several many-core systems to drive the on-demand computational needs.

We identify and document the bottlenecks and computational challenges the combination of interactivity and traditional batch oriented computing creates. The main bottleneck is identified as the execution time of the NWP and preparing data for visualization.

Keywords Interactive numerical weather model, WRF, Visualization, Tiled display wall, Live Data sets, On-Demand computations

1 Introduction

Numerical Weather Prediction models are typically computed for a fixed static region at a fixed resolution. One very high-resolution turbulence forecasting system in daily operational use by the Norwegian Meteorological Institute [8] is the SIMRA system by Eidsvik and colleagues [4]. The number of locations with results from this model available to the weather forecaster is limited by the available computing resources. This limits the areas where the forecaster can assess the current level of risk of severe turbulence. Therefore, only places of interest with a previously known high level of risk have pre-computed models available. At any given day this may or may not be the actual trouble spots.

Results from NWPs are later interactively visualized with the pre-computed resolution on a typical PC display. For any given area and selected parameter it may not be possible to view all the details in addition to the whole area. One example may be viewing the detailed wind field in connection with strong precipitation cells. The use of very large high-resolution displays has previously been shown to be advantageous in such settings [5].

We report on a system and approach for a wall-sized high-resolution tiled display [10], where the user can select a region of interest by zooming in on that region and have NWP done on-demand for the selected area at the desired resolution. The result is visualized in high level of detail. The user can then pan the visualization in all directions. While panning, the on-demand many-core compute cluster will provide the updated weather forecasts.

The ability to select smaller regions of interest and have high-resolution, high quality model forecasts combined with a display wall supported by on-demand computing, enables an interactive experience for the user. This will alter and may also reduce the total computing resources needed within the organisation.

This work is based on an idealized use case:

1. The forecaster browses a coarse resolution model for possible trouble spots.

2. The forecaster zooms in to view details, which triggers a new NWP run.

3. The forecaster receives the result from a run of a high-resolution model for the specific area.

4. The forecaster pans the view to include nearby trouble spots, or zooms out and focuses on a new area.

Both the area of interest and wanted resolution is inferred from the zoom level used by the user.

*Email: baard.fjukstad@uit.no
†Email: tor-magne.stien.hagen@uit.no
‡Email: daniels@cs.uit.no
§Email: phuong.hoai.ha@uit.no
¶Email: john.markus.bjorndalen@uit.no
‖Email: otto.anshus@uit.no

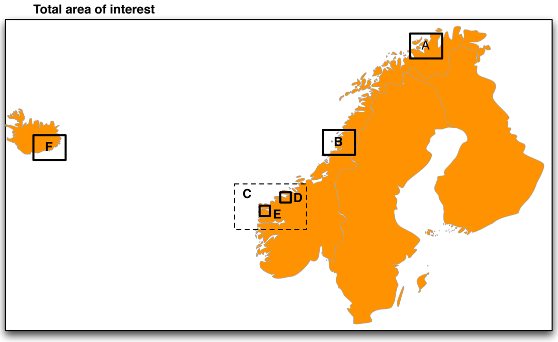

Figure 1: Cases A,B,D, E and F are the trouble spots in this situation. A background model is assumed to be available in the whole area of interest.

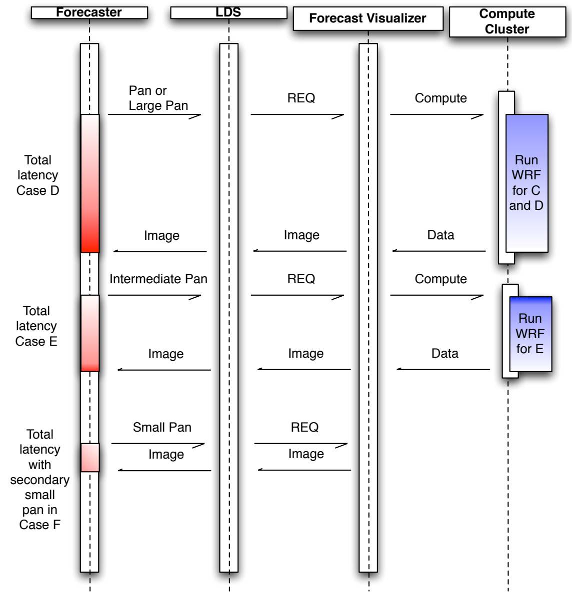

Figure 2: Three different cases are shown. Case D with no intermediate level available. Case D, with the C area available, and Case F where the model is first run on a slightly larger area than requested so that minor Pans does not trigger a full generation of a new area.

2 WRF

We have chosen to use the WRF NWP model [1] for a complete solution that takes into account the time evolution of all relevant meteorological factors. WRF is at present a very popular research model for high-resolution weather forecasting systems. WRF is available in numerous settings and is extensively used in many meteorological research and operational centers [2].

A simpler downscaling of the wind field for each timestep, like the SIMRA system would reduce the work load, but not provide the forecaster with all the information needed.

To simplify the prototype, we limited the model resolution to a fixed set of discrete resolutions corresponding to a fixed number of zoom levels in the visualization. This is also practical given the available background topographical and environmental data.

For our study we used a fixed set of environmental and large scale model forecasts. We also chose not to run an independent start analysis. We still incur most of the workload that an operational system would require.

Figure 1 shows a possible scenario with several trouble spots. Areas A, B and F are large enough for the resolution difference to be small enough between the background model and the wanted areas, for running WRF directly. Areas D and E require an intermediate step, area C, to be computed. When requesting an intermediate area for D, a choice of area C is made so that panning only creates the need to compute for the new smaller case E, given the previous computed area C. Around F we have indicated the actual area used, so that small pans are immediately available. The effect of these scenarios on the perceived latency for the user is shown in Figure 2.

3 Experimental Platform

3.1 The Display Wall

The Display Wall [11] consists of 28 projectors arranged to give a 7168x3072 pixel total display that is perceived by the user as one single coherent display. Each projector is controlled by one computer that is part of the display wall cluster. Zoom and pan is implemented using the touch free interface created by Stødle and colleagues [10].

3.2 Wall Scope

Our visualization and computation platform is the WallScope [9] system. WallScope implements the architecture of Live Dataset (LDS). Visualization clients run on each computer in the Display Wall cluster. Each client requests data from the LDS which initiates local or remote computations to satisfy the request. LDS may also return a cached copy if the computation has been performed earlier.

The LDS architecture is part of the architecture shown in Figure 3. For interactive visualization of weather forecasts, the WallScope system is extended by adding an on-demand simulation and visualization backend using WRF. The architecture separates visualization from data management, and also data management from the data producer. This makes our system independent of the actual display used by the forecaster, and the system can also be used from a regular workstation.

3.3 Compute clusters

Two clusters were used in this project. One is a local 32 node 3.2 GHz Pentium 4 cluster, the other is the 704 node 1408 cpu 5632 core Stallo [3] high performance cluster. For most of our testing our local cluster was used, but for simulating the availability of a modern efficient non-local cluster, a few experiments were conducted using the Stallo cluster. These experiments show the effect of running the

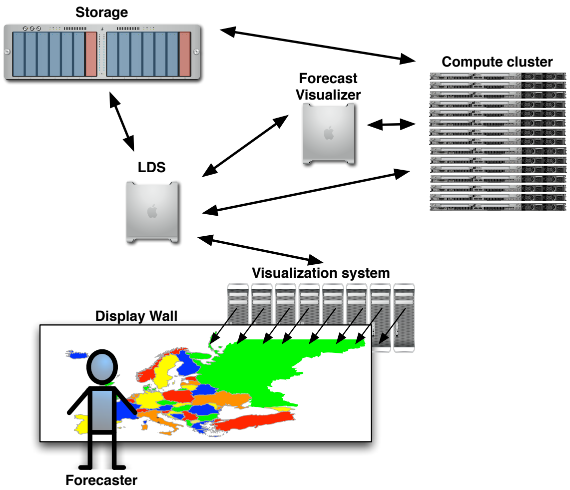

Figure 3: Architecture with the probable data paths indicated

data producing services on a multi node, multi-core platform. The WRF model is expected to scale well and performe well on this platform [6]. The Stallo cluster uses a standard batch job queueing system and are therefore not applicable to interactive use. An express queue with strong limitations on the number of cores available for each job was used for a near real-time interactive use.

One of our areas of continued investigation is the division of labour between the various participating elements in our architecture.

3.4 Network

The experiments were run on the Display Wall lab using gigabit ethernet for communication internally and with the compute clusters.

4 WeatherWall

Our contribution is an interactive system for visualizing state of the art meteorological numerical models at a very high-resolution, for a user-selected areas of interest, the WeatherWall.

Our idea is that the user does not know a priori where high-resolution forecasts would be most useful, and that the user based on available coarser models can select both the area and wanted resolution.

Our architecture is based on WallScope where a visualization client runs on each display cluster node and requests data from the live data set (LDS). LDS sends a processing message to one or more compute resources which generates the data for the visualization system. This allows load balancing and makes it possible to add compute resources as needed.

In our design we have tried to overlap computation with visualization to achieve low latency. This is possible if the visualization processes can access results from the NWP as it is running, and requires the visualization and computation to run on separate systems.

Our system for running WRF in an interactive way is implemented using a small front end to be used by the Live Data set, written in Python. WRFs use of simple text files to control each run makes this a good choice. This also makes it possible to scale the use of the compute cluster with the requested area and resolution.

WeatherWall is a platform for further experimenting with various ways to divide the total work load and also to investigate the many bottlenecks such complex combined systems present.

WeatherWall is also a system that generates and visualizes datasets on demand, as opposed to existing batch oriented systems where datasets are created prior to use and prior to visualizing.

5 Experiments

In order to establish a limit on how long a forecaster would be willing to wait for results we have conducted a small survey of the operational forecasters at the Norwegian Meteorological institute in Tromsø. A total of 14 out of 18 possible responded to the questionnaire.

Time	Count
5-15 sec	2
15-45 sec	0
45 sec - 1 min	3
1-2 min	2
3 - 5 min	3
5 - 10 min	2
more than 10 min	2

Table 1: How long a forecaster is willing to wait for high-resolution forecasts

Table 1 shows that almost 60% of the forecasters are willing to wait more than one minute for the display to update, if this implies higher resolution forecasts.

We are in the process of measuring the user perceived end-to-end latency when zooming and panning the display, and thereby triggering new model runs. Our initial trials show total latencies of the order of several minutes. This is larger than what practical operational use would require, but further optimizations may reduce this significantly.

One observation is that since the display wall and the chosen zoom factor determines the requested area and resolution of the model, the workload on the computational components is relatively constant.

The perceived latency after a pan or zoom is different depending on the available setup. Figure 2 illustrates this. If no high-resolution result is available at the requested zoom level, this will start a full run of the WRF model that may require several steps with increasing resolution before the requested resolution is computed. This is illustrated in the top part of Figure 2 where first area C has to be computed, and then area D.

If the next requested area falls inside the already computed area of C, the request can be satisfied with running the model only for the new area E.

If the request is only for a small pan within the areas already computed, the request will be satisfied from the visualization nodes, as shown in the lowest part of Figure 2.

Task	**Time**
Running pre-processing on cluster front-end	18 sec
Running the WRF model	56 sec
Transfering result file to visualization host	0.4 sec
Retrieving one parameter for visualization	3 sec

Table 2: Average run-times Case E using the Stallo cluster using 8 cores on 1 node. Models domain is 39 x 41, 28 vertical levels, 6 hour forecast with 30 sec time steps.

Our preliminary results in table 2 indicate that the largest bottleneck is the execution of the WRF forecast model. When the forecast model is completed the bottleneck is the generation of visualization data from the computed model.

6 Discussion

Our preliminary results are at most valid for our limited use case. Our system is not an efficient system for delivering high-resolution numerical forecasts each day or at a specific schedule. For such use the traditional batch oriented systems would be better.

The WeatherWall system does not yet provide a true interactive system because of latency times for the user that are longer than operational use will tolerate. When the model data is available, our system has the ability to display high-resolution visualizations for large areas. This enables possible new insight into relevant meteorological problems.

7 Conclusions

We are in the process of implementing a very early prototype of an interactive numerical weather model system. New numerical weather prediction models are relatively easy to set up with a large range in resolutions, limited mostly by available environmental data, and available computing resources. At this time we believe that interactive running of NWPs is coming closer to be a practical solution for operational weather forecasting, and that our system show one possible solution.

8 Future work

Using GPUs in WRF will improve the runtime significantly [7]. Utilizing GPUs may also improve the visualization performance. Both are areas we intend to investigate further.

References

[1] The WRF model. `http://www.wrf-model.org`.

[2] WRF Real Time Forecasting. `http://wrf-model.org/plots/wrfrealtime.php`.

[3] The Stallo cluster, 2010. `http://docs.notur.no/uit/stallo_documentation/user_guide/key-numbers-about-stallo`.

[4] K. Eidsvik, A. Holstad, I. Lie, and T. Utnes. A prediction system for local wind variations in mountainous terrain. *Boundary-Layer Meteorology*, 112(3):557–586, 09 2004.

[5] B. Fjukstad, O. Anshus, and J. Bjørndalen. High resolution numerical models on a Display Wall. In *The 7th Annual Meeting of the European Meteorological Society (EMS) and the 8th European Conference on Applications of Meteorology*, 2007.

[6] J. Michalakes, 2003. `http://www.cisl.ucar.edu/dir/CAS2K3/CAS2K3%20Presentations/Thu/michalakes.ppt`.

[7] J. Michalakes and M. Vachharajani. Gpu acceleration of numerical weather prediction. *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*, pages 1 –7, april 2008.

[8] K. Midtbø, J. B. Bremnes, M. Homleid, and V. Ødegaard. Verification of wind forecasts for the airports hammerfest, honningsvåg, sandnessjøen, ørsta, værnes, sandane and narvik, 2008. `http://met.no/Forskning/Publikasjoner/metno_report/2008/filestore/report02_20081.pdf`.

[9] T. Stien Hagen, D. Stødle, and O. J. Anshus. On-demand high-performace visualization of spatial data on high-resolution tiled display walls, In press.

[10] D. Stødle, O. Troyanskaya, K. Li, and O. J. Anshus. Tech-note: Device-Free Interaction Spaces. In *3DUI '09: Proceedings of the IEEE Symposium on 3D User Interfaces*, pages 39–42, Mar. 2009.

[11] G. Wallace, O. J. Anshus, P. Bi, H. Chen, Y. Chen, D. Clark, P. Cook, A. Finkelstein, T. Funkhouser, A. Gupta, M. Hibbs, K. Li, Z. Liu, R. Samanta, R. Sukthankar, and O. Troyanskaya. Tools and applications for large-scale display walls. *IEEE Comput. Graph. Appl.*, 25(4):24–33, 2005.