Accurate Weather Forecasting Through Locality Based Collaborative Computing

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

and the Norwegian Meteorological Institute
Forecasting Division for Northern Norway
Email: bard.fjukstad@uit.no

John Markus Bjørndalen
Department of Computer Science
Faculty of Science and Technology
University of Tromsø, Norway
Email: jmb@cs.uit.no

Otto Anshus
Department of Computer Science
Faculty of Science and Technology
University of Tromsø, Norway
Email: otto@cs.uit.no

***Abstract*—The Collaborative Symbiotic Weather Forecasting (CSWF) system lets a user compute a short time, high-resolution forecast for a small region around the user, in a few minutes, on-demand, on a PC. A collaborated forecast giving better uncertainty estimation is then created using forecasts from other users in the same general region. A collaborated forecast can be visualized on a range of devices and in a range of styles, typically as a composite of the individual forecasts. CSWF assumes locality between forecasts, regions, and PCs. Forecasts for a region are computed by and stored on PCs located within the region. To locate forecasts, CSWF simply scans specific ports on public IP addresses in the local area. Scanning is robust because it avoids maintaining state about others and fast because the number of computers is low and only a few forecasts are needed.**

Keywords: Weather Forecast, Distributed Computing, Collaboration, Peer to Peer

I. Introduction

Access to weather forecasts for practically any location on Earth is available free of charge over the Internet from meteorological services, like the Norwegian Meteorological Institute Yr.no [1], the European Centre for Medium-Range Weather Forecasts (ECMWF), and the US National Weather Service [2]. There are collaborations between the weather services in that lower resolution forecasts are used to compute forecasts of higher resolution. For instance, the Norwegian Meteorological Institute uses the forecasts from ECMWF to do higher resolution forecasts for Scandinavia. However, there are no collaborations with users, and between users of weather forecasts. The resolution and accuracy of weather forecasts can be increased if there is collaboration between national forecast services and users, and between users.

Three types of collaboration related to weather forecasting can be identified. The first is the collaboration between national weather services. The national service can produce forecasts that other services use as a starting point or as boundary values in their own production of numerical forecasts. The forecasts at national level are medium to long-term, large area, and medium to high-resolution forecasts. They take several hours to compute on supercomputers. The second type of collaboration is when users use forecasts from the national weather services to produce short-term, small area, and higher resolution forecasts. There is no feedback of these forecasts from the users to the weather services. These forecasts take minutes to compute on a single multi-core PC. The third is a symbiotic collaboration where users share on-demand their locally produced forecasts with each other.

In a complex terrain like the fjords and mountains of Norway, the topography have a significant impact on the weather on the very local scale. This represents a serious challenge for numerical models where the spatial resolution limits the ability to produce accurate forecasts. The weather services can increase the resolution to better reflect the topography. While this is gradually happening, it is still primarily done for regions of special interest, like airports. This is because the compute resources applied are not sufficient to do a timely delivery of the forecasts for very large regions, let alone the whole of Earth. Many commercial weather services typically have the same lack of resolution because they repack the forecasts from the national weather services. While they integrate this with other forms of weather related information few compute their own numerical atmospheric models, and if they do, this is often for specialized purposes like wind mill farms for premium customers, and not publicly available.

Local weather forecasts have seen some industry attention. IBM's Deep Thunder project [3] has developed a system delivering targeted high-resolution weather forecasts for smaller areas. The intended use is limited in that the forecasting typically is for a pre-determined fixed area and for a specific use like the 2014 World Cup in Rio.

In summary, the accuracy of an actual forecast is a function of the characteristics of the models used to compute it, how many models are computed, the resolution of the background data, the size of the forecasted area, how far into the future the forecast is for, and the time interval. The model area, resolution and forecast interval must be determined in light of the compute resources available, and the necessary hard wall clock deadlines before the numerical forecasts must be ready

COLLABORATECOM 2013, October 20-23, Austin, United States

DOI 10.4108/icst.collaboratecom.2013.254178

for use. The national services do forecasts for larger regions and at lower resolutions. Commercial services do specific high-resolution forecasts for paying customers.

Even though forecasts are made available whenever a user requires them, the organization around, as well as the approaches used to produce forecasts, lead to a situation where they are pre-computed, instead of being done on-demand. The weather services often strongly select the data they make available, aiming at the most typical usage.

This paper proposes a three-tier approach to producing forecasts. The first tier, **the global forecasts**, is the forecasts for large areas, low resolution, and long time periods produced by the national weather services. These are typically computed as parallel computations on a supercomputer or a compute cluster. The second tier, **the local forecasts**, is producing very high-resolution forecasts for small areas and for short time periods using the tier 1 forecasts as a starting point. The computation is typically parallell and done on multi- and many-core modern PCs. The PCs are located in private homes, and in private and public offices. The geographical locations of these computers are typically in the area for which they produce a forecast. The third tier, **the collaborative symbiotic forecasts**, is producing amalgamated forecasts based on the local forecasts. This is done on and by the same computers used to produce the local forecasts. The global and local forecasts are sufficient to produce very high-resolution forecasts that can be used as is. The symbiotic forecast is used to achieve better error estimations of the forecasted weather, using multiple forecasts computed using a slightly different center position.

The usage scenario is comprised of users around the world wanting to compute on-demand accurate and high-resolution forecasts for a small area. They do so on their own PCs. Firstly, background data is pulled in, or is already pre-fetched, from a national service. Secondly, by restricting the forecast in space and time, a high-resolution numerical forecast is produced in minutes using a professional model. To increase the accuracy of the forecast, forecasts from other persons in and for the area are located and pulled in.

This paper documents the developed architecture, design and implementation of a prototype third tier weather forecasting system, the Collaborative Symbiotic Weather Forecasting (CSWF) system. It has the following characteristics: (i) It produces forecasts that are more accurate and at a higher resolution than what is available from the national weather services. (ii) Forecasts are produced on-demand for a short time period and for any small area of the Earth. (iii) Forecasts take into account the topography. (iv) The user gets access to all data generated by the numerical model, avoiding the selection of data done by the weather services. (v) A future version of the system can utilise observations done by users.

The contributions of this paper is the presentation of a novel way of generating collaborative created, user computed, on-demand weather forecast with very high spatial resolution.

II. Collaborative Weather Forecasting

Modern weather forecasting is based on one of the most successful international collaborative efforts [4]. Using observations and forecast products exchanged with a global telecommunications networks, GTS[1], that predates the internet, national weather services have access to all the background information needed for both global, regional and local weather forecasting.

Dedicated supercomputer clusters are used for running large numerical forecast models, and a very large storage infrastructure is used for storing the forecasts and observational background data. The size of this infrastructure can be illustrated with the budgeted $ 23.7 million[2] 2013 update of infrastructure and computing facilities to the National Weather Services, NWS, following the Sandy hurricane.

It should be stressed that such very expensive systems are indeed needed for providing forecasts for large areas. These systems provide the necessary background meteorological data for the personalized collaboration system described in this paper.

National weather services typically uses a client-server model in the form of web-based systems for preparing and visualizing meteorological data, and for making available the raw datafiles. The user accesses the data through a regular web browser or use local apps, which typically download the data onto the device. One example of this model is the NOAA Operational Model Archive and Distribution System (NOMADS) [5].

III. Collaborative Symbiotic Weather Forecasting

The observation behind the Collaborative Symbiotic Weather Forecasting (CSWF) model is that national and commercial weather forecasting services do not have the resources to offer high-resolution forecasts for arbitrary parts of the Earth selected on-demand by public users. Therefore, pre-computed lower resolution forecasts are provided for large areas, and higher resolution forecasts are created for pre-selected areas, like airports, and for paying customers.

Previous work [6] have shown that users can do their own numerically computed forecasts using a widely used numerical atmospheric model, WRF [7], in a few minutes using their own commodity quad-core home PCs, if they are willing to limit the forecast to a small area, e.g. 40x40 km, and for a short time period, e.g. six hours.

Small region forecasts are embarrassingly distributed because each forecast is computed in isolation. In collaboration between geographically close neighbors, forecasts with overlapping areas can be exchanged and combined. Just having a few extra forecasts for an area will enable uncertainty estimation and thereby increase the value of the participants own forecasts.

[1]http://www.wmo.int/pages/prog/www/TEM/GTS/index_ en.html

[2]http://www.washingtonpost.com/blogs/capital-weather-gang/wp/2013/05/15/game-changing-improvements-in-the-works-for-u-s-weather-prediction/

If the same model, terrain and meteorological background data were used in the forecast, the differences in the forecasts are simply a result of the slightly different model terrain representation for the different numerical grids. If different models and meteorological backgrounds were used, the differences would also incorporate other sources of uncertainty in the forecasts, further increasing the value of this collaboration.

The method of locating other people to collaborate with is based on the assumption that people living in an area will compute the majority of forecasts for the area. The assumption is also that their forecasting PC's will be located in the area and that this is where we can find the forecasts we need to exchange. Different approaches can be used for location possible collaborative partners. (i) A third party approach can be used, where forecasts are stored on or CSWF systems are reported to a central server. (ii) A semi-permanent peer-to-peer and a distributed hash table approach where forecasts are stored in a known structure using a hash value as a key. (iii) Nodes can be found after searching in search engines. (iv) Receiving gossip from other nodes. (v) Simply probing the local network.

Our prototype uses the last approach. The geographical area of interested are known, so a few IP addresses for PCs in the area can be found and probed directly to see if they are running the CSWF system and if they have forecasts of interest. Not all computers in the area will be found, but finding a few can be enough because just a few extra forecasts are of great value to the combined forecast. The scanning for other CSWF systems can be overlapped with the local forecast computation, and will finish well before the local forecast.

IV. Architecture

The Collaborative Symbiotic Weather Forecasting, CSWF architecture is illustrated in Fig. 1, and have two abstractions: the forecast presentation and the forecasts themselves. The forecast presentation represents how forecasts are presented and visualized. It will materialize as client applications on the users devices, typically smart phones, tablets, and laptops.

The forecast abstraction comprises of the local forecast for a given small area, the collaborators forecasts available for the same, or part of the same area, and the symbiotic amalgamated forecast. The forecast abstraction will materialize as a server on the user's PC.

The local forecast is done on a user's PC using background meteorological data from a national forecast service. Other collaborators PCs compute their forecasts. Each collaborator computes the symbiotic amalgamated forecast locally based on the local forecast and the others forecasts. The amalgamated forecasts would typically contain the mean, minimum, maximum and standard deviation values for any wanted parameter, computed used all available forecasts.

The security model in the CSWF system is very simple. If a user does not want anybody to monitor the forecast interests, the collaborative symbiotic system will not locate and fetch other collaborators forecasts at all. If forecasts have been manipulated, this is discovered by comparing with the local forecast. If they differ too much, the remote forecasts are discarded. If a system is flooded by requests from collaborators for forecasts, the collaboration system detects this, and ignores all incoming requests for a certain period. Both the computers and devices presenting and producing forecasts are under user control, and do not need other computers or external networks to produce and present a new local forecast. A byzantine forecast can easily be detected by the accumulated differences from the local forecast, and can be discarded based on some given statistical criteria for acceptance. This would typically be done after collecting all collaborators forecasts, for a more stable statistical criterium based rejection.

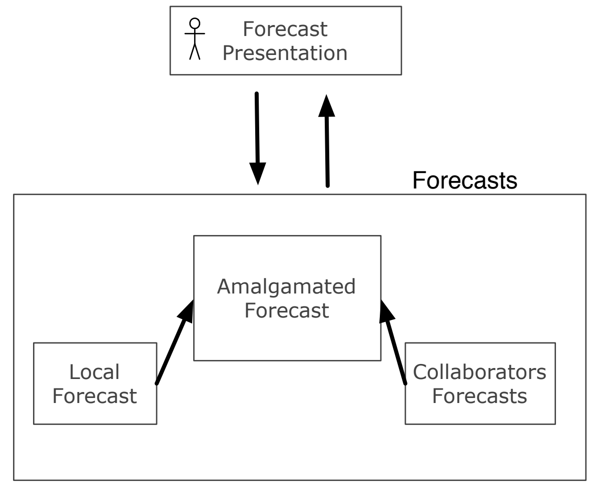

Fig. 1. The Collaborative Symbiotic Weather Forecasting (CSWF) architecture.

V. Design

The design illustrated in Fig. 2, is based on a client-server model. All communication between services and between services and applications use simple REST [8] style HTTP request. The presentation application on a client device request a local forecast and an amalgamated forecast from a forecast server, which maintains a list over existing forecasts. If it already has relevant forecasts ready to serve the request, it just returns to the presentation application an URL for each requested forecast. If it does not already know of relevant forecasts, the front-end uses two sub systems, the computational and the forecast amalgamate subsystem, to produce forecasts.

If an amalgamated forecast is not available, the amalgamated sub-system uses the collaborative sub-system to fetch remotely produced forecasts. When the computational sub-system, and the collaborative sub-system finishes, they tell the amalgamated sub-system this. The amalgamated sub-system then combines the local forecast with the remote forecasts into an amalgamated forecast, and stores it on the local file system.

To provide the local forecast system with background forecasts from national forecast services, the local forecast computation system automatically fetches meteorological data from a

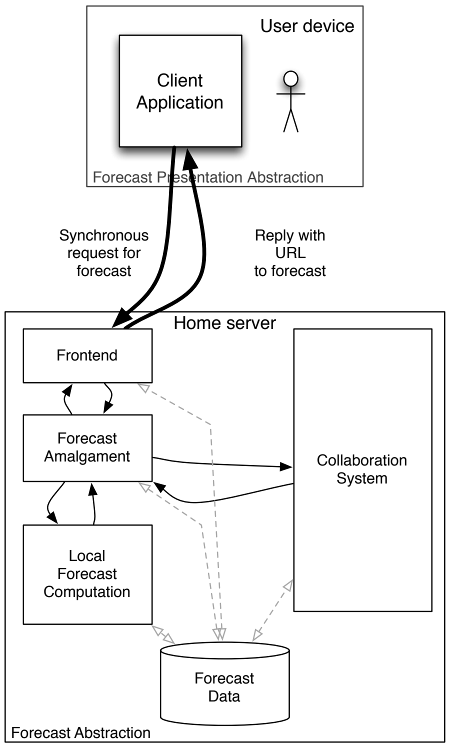

Fig. 2. The Collaborative Symbiotic Weather Forecast (CSWF) design.

meteorological service and stores them on the local file system. This is done regularly one or several times each day, depending on the availability of new background meteorological data. The Collaborative sub-system comprises of several sub-systems as illustrated in Fig. 3. The discovery system locates remote computers running the CSWF system, and maintains a short-lived list of discovered computers. Discovery is presently done by probing a specific port on computers with IP addresses assumed to be within the relevant geographical area.

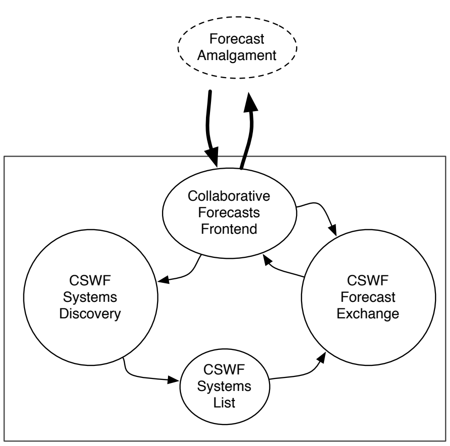

Fig. 3. The Collaborative sub-system design.

The exchange system fetches remote forecasts from computers in the list of newly discovered CSWF computers, and makes local forecasts available to them. This is done peer-to-peer, where each collaboration system communicates directly with each of the other collaboration systems having relevant forecasts. Whether a CSWF system actually downloads forecasts from recent contacted systems are dependent on the preferences of each individual CSWF system. The discovery system makes no assumptions on the availability of previous discovered systems. When a remote CSWF system is discovered it will list all current forecasts that is made available and the local collaboration system can decide on which forecasts to retrieve based on their geographical area and time.

VI. CSWF Implementation

Applications for the forecast presentation abstraction have been implemented for a range of platforms and operating systems. This include applications for iPhones, iPads, mobile device web browsers, web browsers on laptops or stationary devices and a simple data conversion programs to create NETCDF CF[3] compliant data usable in the DIANA [9] application.

The DIANA application is used for visualizing forecasts on laptop and desktop computers and on a display wall. For this DIANA is executed on a PC displaying into a very large virtual VNC [10] frame buffer of 22Mpixels. The frame buffer is used by 28 viewers on 28 PCs driving 28 projectors on the Tromsø Display Wall [10]. In this application, we can combine the local, collaborated forecast with globally available forecasts with lower resolution.

The various services in the CSWF system are a set of multi-threaded processes communicating using HTTP REST [8]. Each sub-system comprises of one or several processes. All server processes are compiled for running on either Linux or OS X. The 3.4 version of the WRF atmospheric model is compiled for running on a Linux computer.

Forecast presentation applications can request forecasts for any small geographical area for a 6-hour period. The prototype is limited to an area including Scandinavia because of limitations in disk space for the background topographical and other data. Global coverage of these static background data sets is freely available, and takes around 10 GB of storage at the current spatial resolution.

A. Firewall and NAT

Accessing a CSWF system located at home from a roaming mobile device is complicated because home networks frequently are behind firewalls, and use network address translation, NAT. The CSWF system could have been accessed from devices on external networks using NAT traversal techniques [11]. In the prototype, no techniques for NAT traversal are used. A CSWF system is expected to be made accessible by correct setup of firewalls, possibly using Simple Network Management Protocol (SNMP) and Universal Plug & Play (UPnP).

[3] http://cf-pcmdi.llnl.gov/

VII. USER APPLICATIONS

Several prototype forecast presentation applications for Linux, OS X and iOS have been created. Some of these are described briefly in the following sections. Each user will be able to use the applications on various platforms to specify and adjust the visualization of the meteorological parameters for various specific purposes. To explore the effects of supporting different demands locally, several different visualization options have been created

A. 2D visualization

To visualize forecasts in 2D and on typical platforms, a browser on the forecast presentation device is used. The browser runs a small Javascript visualization script that uses the Google Maps API. The script pulls in image tiles from CSWF and renders them on the presentation device.

B. 3D visualization and Augmented Reality

3D graphics has seen little use in operational weather forecasting where the focus is often on detailed values at specific locations. 2D graphics is simpler to work with in these situations. This is reflected in the number of 2D products available from the National Weather Service, NWS[4] versus the relatively smaller number of 3D products.

An application for a tablet that shows the current view of the back facing camera overlaid with meteorological information have also been created. Using data from the GPS, the compass and the accelerometers on the device, the location, which way the camera is facing and the tilt of the device is known.

The user can request a forecast centered around his GPS coordinates, and then explore the weather forecast by pointing the tablet's camera into the surrounding landscape to study on the tablets display the weather forecast superimposed with the camera image. To view data from another location, the user must physically move around. A screen shot of the application is shown in Fig. 4. This application uses a device with a screen size usable for detailed visualization. The device has communication capabilities sufficient to receive the data in KML format, and has the processing power to do the visualization on the device.

VIII. RESULTS

A. The Symbiotic Effect on Forecasts

Symbiotic forecast are created by aggregating forecasts from collaborators using the local forecast grid as a basis. By having many possible forecast values for each parameter for each grid point, statistical products can be created. Typically a mean value and a standard deviation is useful.

The individual forecasts from all collaborators can also be used in visualizations together with the local forecast. One example is shown in Fig. 5 where the center location for each forecast is varied slightly, but the background meteorological data used is the same. The figure shows that large differences can be observed in the forecasts for specific locations. In this

[4]http://www.nws.noaa.gov

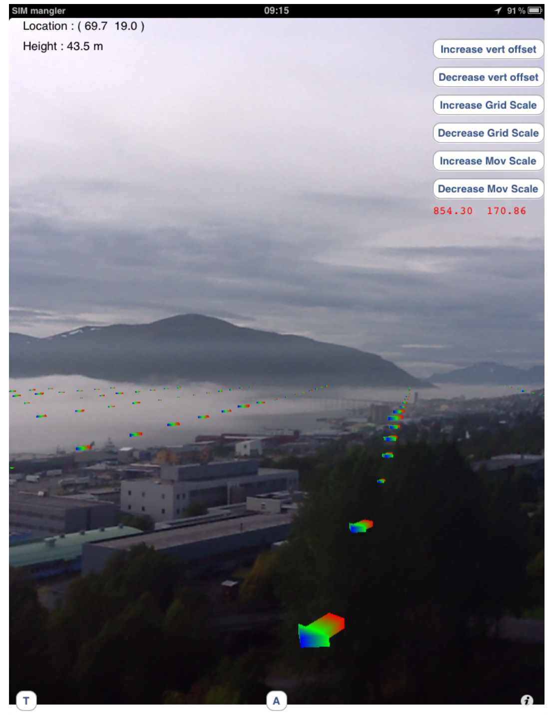

Fig. 4. Screenshot of iPad2 with camera and an example of overlaid meteorological information

case, the area studied is around a planned site for a windmill farm. The forecasts indicate both the local uncertainties in the forecast for a specific point and the variations expected over the whole area. The variations are here illustrated by displaying several individual forecasts from the local forecasts and from collaborative collected forecasts. This variation could also have been illustrated using a map of standard deviations computed from the aggregated local and collaborative collected forecasts.

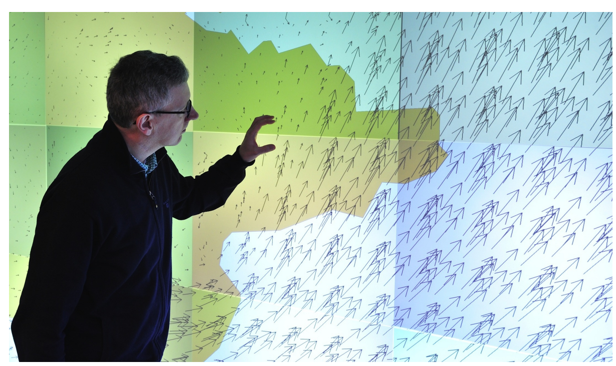

Fig. 5. Studying several forecasts collected from collaborating nodes on a display wall (Colored background is land areas, white background is the sea.)

B. Performance Measurements

Some of the system's performance characteristics have been documented through a set of experiments.

The Forecast Presentation applications are lightweight enough to run interactively on a smartphone or tablet with no noticeable lag. The bandwidth requirements are also modest: only 1MB is transferred to the mobile device to visualize the wind arrows shown in Fig. 4

The CSWF computational sub-system is configured to consume all CPU resources of the host computer when computing new forecasts using the WRF atmospheric model. On the desktop computer with four hyper-threaded cores, a forecast is computed in 2 to 7 minutes depending on the resolution of the forecast. The memory footprint is just 1GB and the CPU load is 100% .

C. Detecting Remote Forecasts

One experiment to measure how fast we could detect remote CSWF computers were conducted. All computers had good Internet connections with bandwidths better than 10 Mb/s. Some of the computers were at the university of Tromsø and were visible from the Internet. Other computers were behind firewalls with NAT operated by either the University of Tromsø or a commercial ISP used by many households in Tromsø.

The nmap[5] application was used to measure the time it took to scan for any computers in an IP address range with 256 possible computers (nnn.nnn.234.00/24) in two different settings. The results are shown in Table I. This example shows little difference in scanning for host from within the same network and therefore the same IP address range, or from a computer on another network. Notable is how fast it is to scan for a single open port, regardless of the scanning computers location.

TABLE I
SCAN TIMES USING NMAP WITH 256 POSSIBLE HOSTS

Setting	Seconds	Hosts located
From same network, all ports	146	6
From other network, all ports	107	23
From same network, single port	3.1	8
From other network, single port	3.7	23

D. Wind Forecast Validation

The general performance of the WRF model used in the prototype has been documented many times; two examples are given in [12], [13], and has also been validated for wind forecasting [14]. As an illustration of the quality, a short time series of wind speed forecasts and observations for the local Airport in Tromsø is given in Fig. 6.

Only in a few of the shown days were the observed windspeed significantly outside the range of the symbiotic amalgamated forecasts. The mean value of the symbiotic amalgamated forecast resulting from the collaboration with other systems were a better forecast than the single local forecast on a majority of days. Both forecast show significant skill in forecasting windspeed.

[5]http://nmap.org/

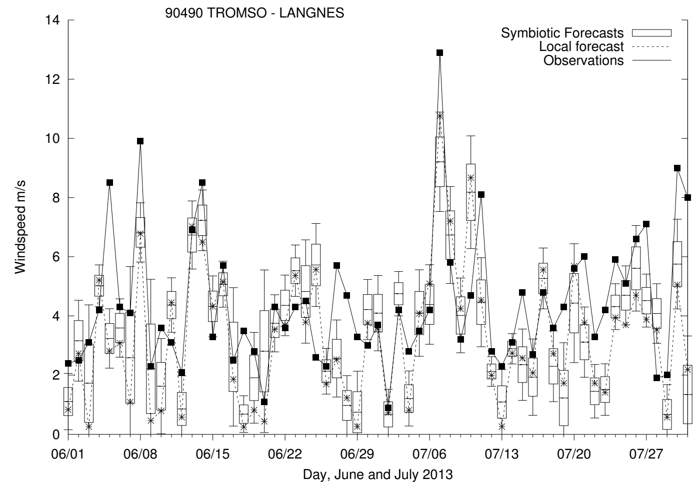

Fig. 6. Timeserie of forecasted and observed wind speed. Solid dots and lines shows observations. Starts and dotted lines shows a single forecast. Bars with whiskers shows distribution of the 28 symbiotic forecasts with a mean value indicated.

IX. RELATED WORK

The PrPI systems [15] is a decentralized infrastructure for social networking. PrPI, short for private-public, uses a Personal Cloud Butler service to support sharing with fine-grained access control. The Butler can be run on a home server or use a paid service. It uses OpenID[6] for decentralized management where the users can use their established personas for accessing data. The focus of the PrPI system is sharing of data, not on computations or processing. The shared data can be distributed on remote data systems and may be encrypted. Data in PrPI is identified using Resource Description Framework, RDF [16]. The semantic index is schema-less. Accessing data in PrPI uses Universal Resource Identifiers, URI, pointing to Data Stewards, which in turn periodically sends heartbeats to the Butler with updated information on both data content. The Data steward can therefore map a virtual PrPI URI to a physical URL starting with http://. The CSWF system assumes a much shorter lifetime of the shared data. The forecasts will become stale after a few hours. The CSWF system can afford the short time it takes to scan for neighbor CSWF systems since the scanning is overlapped with the computation of the new local forecast.

NASAs Climate in a Box initiative [17] also built a system for running large models on Desktop supercomputers. The focus has been to create a better system for opening the model development process to a community using very high end desktop systems like the Cray CX1 [18]. The system uses many of the tools for job control and administration used on supercomputer clusters. The Cray CX1 can, with these tools, be used as a small 8 to 16-node cluster. This represents therefore the type of system we envision will be commonplace in a few years. In the CSWF system the focused is on using a

[6]http://openid.net/

single model and on practical weather forecasting, not climate modeling. The model is optimized for the physical hardware it is run on, and is be able to deliver forecasts on demand.

One example of using crowd-sourced sensing [19] has illustrated the potential of using social networks for both reporting and collecting sensor network data and personal observations. These systems would fit nicely into a system where local observations are assimilated into the starting state of the atmosphere before computing a new forecast. The reported systems do not do local computations on the collected values or produce forecasts that are further shared.

X. Discussion

Independently computed weather forecasts for the same area can be combined using a collaboration system that exchanges numerical weather forecasts between users within overlapping geographical areas. The combined forecasts add value to each users own locally computed forecast by improving the uncertainty estimations. This can be visualized either by a simple combination of several forecasts in the same map, or calculating standard deviation of some parameter and visualizing this together with the local forecast.

A network of CSWF systems is not a typical Crowd Sourcing [20] system. Perhaps CSWF is an example of accidental crowd sourcing where it just so happens that forecasts for a given area is done by a crowd. CSWF can also be compared to a flash crowd in that events like bad weather or emergencies create a large interest for forecasts in or close to an area resulting in a crowd of locally computed forecasts.

In the CSWF system, we use custom applications for visualizing the weather forecasts. They execute on the devices available to the user. These devices range from mobile devices to large display walls comprised of many computers. The range of device capabilities make dedicated applications better suited to utilize the characteristics of the devices.

To reduce the compute time for a small area forecast we need faster processor cores. The number of grid-points used in the high-resolution forecasting model needs to be adjusted to balance the tradeoff between the desired resolution, the available computing resources and how long the user is willing to wait. Increasing the resolution of the model increases the running time of the simulation since we have to increase the number of time steps in the simulation.

Table II shows how the number of time steps needed for a 6-hour forecast, increases with the resolution. Four km is the same resolution as the available background data from the Norwegian Meteorological Institute; 1 km is what will be available in a few years and 100 m is what we can envision that a user needs at some point in the near future.

Each time step must be completed on all nodes before the next can be started. This limits how much the model can be parallelized and therefore how quickly we can run the model.

The area covered by the numerical atmospheric model must be such that the meteorological issues can be resolved with the higher resolution. In our experiments, we have observed the meteorological effect of the border values (the outer edges of the high-resolution grid) extend as far as 3-4 grid points into our grid. The experiments were done using a grid with a size of either 39 x 41 or 43 x 43, for various spatial resolutions.

TABLE II
A simple rule of thumb comparison between model resolution, length of time step and number of time steps in a 6 hour forecast

Resolution	Time step in seconds	# time steps
4 km	24	900
1 km	6	3 600
100 m	0.12	180 000

The forecasts have been both expected and appears credible to an experienced weather forecaster and the very steep and complex topography in our area do introduce known steering of wind, and these are at least in some regard reproduced by the model.

The results in Fig. 6 illustrate the effect of collaboration. The mean value of the symbiotic amalgamated forecast were often better that the single local forecast, and on many days the range of the amalgamated forecast included the observed value. This is also an expected result that is similar to the effect seen from EPS forecast validations [21].

Scanning for geographically nearby CSWF systems for collaboration is robust because it is simple and avoids maintaining state about others. It is also fast because the number of computers nearby is low and because just a few extra forecasts are needed for added value to the local forecast.

It will be interesting to see as our research progresses if we are right in assuming that forecasts for an area are sufficiently correlated to where the PC's producing them are physically located to be used as the basis for a locality based approach for finding forecasts for the area.

The characteristics of small area weather forecasting is that it can be computed on a typical PC in a few minutes, and that it is perfectly usable and at a professional level. Adding other forecasts from the same area will improve the forecast, but the local forecast does not dependent upon them and is still usable. The collaborative part is voluntary. These characteristics help in making the system simple and robust against external threats. If problems are encountered, the collaborative system interacting with the environment will simply switch off and isolate the local system. Even when this is happening, the local system will continue to be usable and produce perfectly useful forecasts.

XI. Future work

The prototype of the CSWF systems uses a very simple protocol for establishing collaboration between peers. Further research should include using XMPP [22] for standardizing the communication. The prototype CSWF system also resembles most a serverless XMPP setting and ideas from Klauck et. at [23], [24] both on localized P2P collaboration. Other aspects for further studies are using cloud infrastructure that could replace and enhance the information discovery and distribution of the system.

XII. Conclusion

National Weather Services primarily offer pre-computed forecasts a few times a day for a fixed large region and at a resolution limited by available compute resources and the time available until the forecasts must be ready for use. Therefore, on-demand forecasting for any small region, for a short time span, and at a resolution, reflecting the effect of complex terrain is presently not offered. However, using the large region low-resolution forecasts as input data, this can be done on-demand on a modern PC in a few minutes even when using a professional forecasting model and implementation. Interestingly, combining several forecasts from overlapping but slightly different regions results in a collaborated forecast reflecting the actual weather better than any of the component forecasts do. To do a combined forecast, collaboration between the forecasting systems residing on each users PC is needed to locate relevant forecasts and to exchange them. The CSWF system does a trade-off between scalability and simplicity. The number of forecasts needed to do a useful collaborated forecast is low. It is also assumed that there is a locality between a forecasted region and the PC computing and storing the forecast. Based on this, the CSWF system uses a simple scanning approach to locate PCs in nearby networks, and inquire them about forecasts. While this will not scale to interactively fast finding PCs and forecasts in and for large regions, it will in typically a few seconds find enough forecasts to do better than the stand-alone forecast can. However, we have not documented that the assumed locality is real. We expect this to be the case because so few forecasts are needed, and because local weather typically will primarily interest local users.

Acknowledgment

This work has been supported by the Norwegian Research Council, projects No. 159936/V30, SHARE - A Distributed Shared Virtual Desktop for Simple, Scalable and Robust Resource Sharing across Computers, Storage and Display Devices, and No. 155550/420 - Display Wall with Compute Cluster.

References

[1] met.no, 11 2012. [Online]. Available: http://yr.no

[2] NOAA, 11 2012. [Online]. Available: http://ruc.noaa.gov/hrrr/

[3] IBM, 08 2011. [Online]. Available: http://www.ibm.com/ibm100/us/en/icons/deepthunder/

[4] ITU. (2005) WMO Case Study. [Online]. Available: http://www.itu.int/ITU-D/ict_ stories/themes/case_ studies/wmo.html

[5] NOAA. NOAA National Operational Model Archive & Distribution System - NOMADS Home Page. [Online]. Available: http://nomads.ncdc.noaa.gov

[6] B. Fjukstad, J. M. Bjørndalen, and O. Anshus, "Embarrassingly Distributed Computing for Symbiotic Weather Forecasts," in *Proceedings of the International Conference on Computational Science, ICCS*, Jun. 2013, pp. 1217–1225.

[7] J. G. Michalakes, M. McAtee, and J. Wegiel, "Software Infrastructure for the Weater Research and Forecast Model," *Presentations UGC 2002*, pp. 1–13, 2002.

[8] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, University of California, Irvine, 2000.

[9] E. Martinsen, A. Foss, L. Bergholt, A. Christoffersen, H. Korsmo, and J. Schulze, "Diana: a public domain application for weather analysis, diagnostics and products." Sep. 2005, http://diana.met.no/ref/0279_Martinsen.pdf. [Online]. Available: http://diana.met.no/ref/0279_Martinsen.pdf

[10] D. Stødle, J. M. Bjørndalen, and O. J. Anshus, "De-Centralizing the VNC Model for Improved Performance on Wall-Sized, High-Resolution Tiled Displays," *NIK*, 2007. [Online]. Available: http://folk.uio.no/nik/2007/05-Stodle.pdf

[11] A. Müller, N. Evans, C. Grothoff, and S. Kamkar, "Autonomous NAT Traversal," in *10th IEEE International Conference on Peer-to-Peer Computing (IEEE P2P 2010)*. IEEE, 2010.

[12] K. M. Hines and D. H. Bromwich, "Development and Testing of Polar Weather Research and Forecasting (WRF) Model. Part I: Greenland Ice Sheet Meteorology*," *Monthly Weather Review*, vol. 136, no. 6, pp. 1971–1989, Jun. 2008.

[13] J. J. Ruiz, C. Saulo, and J. Nogués-Paegle, "WRF Model Sensitivity to Choice of Parameterization over South America: Validation against Surface Variables," *Monthly Weather Review*, vol. 138, no. 8, pp. 3342–3355, Aug. 2010.

[14] D. Carvalho, A. Rocha, M. G. mez Gesteira, and C. Santos, "A sensitivity study of the WRF model in wind simulation for an area of high wind energy," *Environmental Modelling & Software*, vol. 33, no. 0, pp. 23–34, 2012.

[15] S.-W. Seong, J. Seo, M. Nasielski, D. Sengupta, S. Hangal, S. K. Teh, R. Chu, B. Dodson, and M. S. Lam, "Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services Social Networks and Beyond - MCS '10," in *the 1st ACM Workshop*. New York, New York, USA: ACM Press, 2010, pp. 1–8. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1810931.1810939

[16] W3C. RDF - Semantic Web Standards. [Online]. Available: http://www.w3.org/RDF/

[17] G. S. Wojcik, M. S. Seablom, T. J. Lee, G. R. McConaughy, R. Syed, A. Oloso, E. M. Kemp, J. Greenseid, and R. Smith, "Nasa's climate in a box: Desktop supercomputing for open scientific model development," *AGU Fall Meeting Abstracts*, pp. B1082+, Dec. 2009.

[18] CRAY, "CRAY CX1." [Online]. Available: http://www.cray.com/Assets/PDF/products/cx1/CX1_ brochure.pdf

[19] M. Demirbas, M. A. Bayir, C. G. Akcora, Y. S. Yilmaz, and H. Ferhatosmanoglu, "Crowd-sourced sensing and collaboration using twitter," *World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a*, pp. 1–9, 2010.

[20] J. Howe. (2006, Jun.) Wired 14.06: The Rise of Crowdsourcing. [Online]. Available: http://www.wired.com/wired/archive/14.06/crowds_ pr.html

[21] C. Ziehmann, "Comparison of a single-model EPS with a multi-model ensemble consisting of a few operational models," *Tellus A*, vol. 52, no. 3, pp. 280–299, May 2000.

[22] O. Ozturk, "Introduction to XMPP protocol and developing online collaboration applications using open source software and libraries," *Collaborative Technologies and Systems (CTS), 2010 International Symposium on*, pp. 21–25, 2010.

[23] R. Klauck, J. Gaebler, M. Kirsche, and S. Schoepke, "Mobile XMPP and cloud service collaboration: An alliance for flexible disaster management," in *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2011 7th International Conference on*, 2011, pp. 201–210.

[24] R. Klauck and M. Kirsche, "Chatty things - Making the Internet of Things readily usable for the masses with XMPP," in *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference on*, 2012, pp. 60–69.