Environmental Modelling & Software 93 (2017) 418–435

Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

Integrating hydrological modelling, data assimilation and cloud computing for real-time management of water resources

Wolfgang Kurtz [a, b, *], Andrei Lapin [c], Oliver S. Schilling [d], Qi Tang [a, b], Eryk Schiller [e], Torsten Braun [e], Daniel Hunkeler [d], Harry Vereecken [a, b], Edward Sudicky [f, g], Peter Kropf [c], Harrie-Jan Hendricks Franssen [a, b], Philip Brunner [d]

[a] Institute of Bio- and Geosciences (IBG-3): Agrosphere, Forschungszentrum Jülich GmbH, Jülich, Germany
[b] Centre for High-Performance Scientific Computing in Terrestrial Systems (HPSC-TerrSys), Geoverbund ABC/J, Jülich, Germany
[c] Computer Science Department (IIUN), Université de Neuchâtel, Neuchâtel, Switzerland
[d] Centre for Hydrogeology and Geothermics (CHYN), Université de Neuchâtel, Neuchâtel, Switzerland
[e] Communication and Distributed Systems (CDS), University of Bern, Bern, Switzerland
[f] Department of Earth and Environmental Sciences, University of Waterloo, Waterloo, Ontario, Canada
[g] Aquanty, Inc., Waterloo, Ontario, Canada

ARTICLE INFO

Article history:
Received 20 November 2016
Accepted 11 March 2017
Available online 27 April 2017

Keywords:
Cloud computing
Integrated hydrological modelling
Data assimilation
Water resources management
HydroGeoSphere
Wireless sensor networks

ABSTRACT

Online data acquisition, data assimilation and integrated hydrological modelling have become more and more important in hydrological science. In this study, we explore cloud computing for integrating field data acquisition and stochastic, physically-based hydrological modelling in a data assimilation and optimisation framework as a service to water resources management. For this purpose, we developed an ensemble Kalman filter-based data assimilation system for the fully-coupled, physically-based hydrological model HydroGeoSphere, which is able to run in a cloud computing environment. A synthetic data assimilation experiment based on the widely used tilted V-catchment problem showed that the computational overhead for the application of the data assimilation platform in a cloud computing environment is minimal, which makes it well-suited for practical water management problems. Advantages of the cloud-based implementation comprise the independence from computational infrastructure and the straightforward integration of cloud-based observation databases with the modelling and data assimilation platform.

1. Introduction

Hydrological and hydrogeological systems are highly heterogeneous, and the temporal evolution of their spatially variable states is driven by dynamic forcing functions. Deterministic numerical models are an important tool for understanding and managing such systems. Such models can support the water management decision making process with predictions of the temporal evolution and the spatial distribution of target state variables. Groundwater management often relies on simulations with distributed, physically-based hydrological models, e.g., for well field operations adjacent to a river. The available numerical models have greatly improved in recent years. There are, for example, ongoing efforts towards a better description of the dynamic feedbacks between subsurface and surface water processes (Kollet and Maxwell, 2006; Brunner and Simmons, 2012). One of the advantages of such physically-based, fully-coupled (or integrated) surface-subsurface models is that the location of surface water features, such as the position of rivers, no longer needs to be predefined through boundary conditions. They are therefore very well-suited for simulating changing surface water conditions such as floods or droughts.

Deterministic models need to be calibrated based on existing observations. However, there is a growing awareness of the uncertainty related to such deterministic model predictions (Liu et al., 2012). The uncertainties stem from the limited knowledge about the spatial distribution and magnitude of important model parameters, such as hydraulic conductivity or porosity (Chen and Zhang, 2006; Hendricks Franssen and Kinzelbach, 2008). Also, the

* Corresponding author. Institute of Bio- and Geosciences (IBG-3): Agrosphere, Forschungszentrum Jülich GmbH, Jülich, Germany.
E-mail address: w.kurtz@fz-juelich.de (W. Kurtz).

http://dx.doi.org/10.1016/j.envsoft.2017.03.011
1364-8152/

high spatio-temporal variability of boundary conditions and input variables, such as precipitation, can highly affect the quality of model predictions. Moreover, the limited availability of spatial and temporal field data limits the reliability of the calibration process. Finally, the computational requirements of many numerical simulators, especially fully-coupled, physically-based models, exclude in most cases a solid uncertainty analysis. These uncertainties can be substantial and they often undermine the credibility of hydrological and hydrogeological models, especially once it comes to predicting highly dynamic systems.

However, a range of technological and mathematical advances allows overcoming some of the previous limitations. Above all, these advances are related to three key developments: data acquisition techniques, the increasing computational capacities of hydrological models, as well as the integration of measurement data in the modelling process.

Firstly, the acquisition of field data has been greatly facilitated. Traditionally, hydrological field measurements such as piezometer levels, precipitation, soil moisture, discharge or water quality indicators, were acquired either manually in the field at predefined measurement intervals or recorded with data loggers, which have to be read out on a weekly or monthly basis. Ongoing advances in sensor technology and telemetry make it now possible to obtain hydrological data shortly after their acquisition in the field, even for very remote field sites. Wireless sensor networks (WSNs) are increasingly applied in environmental studies, e.g., in the context of soil moisture monitoring (Robinson et al., 2008; Ritsema et al., 2009; Bogena et al., 2010) or surface water monitoring (Li et al., 2011), studies on wetland dynamics (Watras et al., 2014) or the acquisition of solute transport data for modelling purposes (Loden et al., 2009; Barnhart et al., 2010). Such WSNs consist of distributed sensors that transmit the measured data through wireless in-built radio modules to a set of router units that manage the communication within the network (Bogena et al., 2010). As a result, the measured data can be accessed by the user on a permanent storage device in near real-time. This can be of great advantage for water management purposes, especially when a system needs to be controlled and regulated continuously (such as pumps for river bank filtration), and the hydraulic forcings are highly transient (such as the water level in a river).

Secondly, the computational efficiency of hydrological models is continuously increasing. Recent advances in numerical mathematics lead to the development of more efficient solvers and preconditioning techniques (Herbst et al., 2008; Maxwell, 2013). Parallelisation of model codes (Ashby and Falgout, 1996; Vereecken et al., 1996; Jones and Woodward, 2001; Mills et al., 2007) makes it possible to solve hydrogeological problems with a high spatio-temporal resolution and on large scales. These advancements in computational efficiency also facilitated the usage of more sophisticated physical process descriptions in the modelling process. For example, state-of-the-art hydrological models now also provide a full 3D solution of the Richards equation, and a physically consistent coupling between surface and subsurface flow equations (Kollet and Maxwell, 2006; Brunner and Simmons, 2012).

Finally, the combination of sequential data assimilation techniques like the ensemble Kalman filter (EnKF) (Evensen, 1994; Burgers et al., 1998) with hydrogeological models now allows integrating real-time data into the modelling process. These methods can be used to effectively merge uncertain model predictions with uncertain observation data in a Bayesian sense. The uncertainty of model predictions is approximated through the forward simulation of an ensemble of model realisations, where each realisation can have a different combination of initial conditions, model forcings and model parameters. The uncertain model predictions are then sequentially updated with measurement data. In this updating step, the uncertainties in the model predictions and the uncertainties of the observations are optimally weighted and the model predictions are effectively adjusted towards the measured data. Besides the correction of state variables, it is also possible to use observation data to update model parameters jointly with the model states (Hendricks Franssen and Kinzelbach, 2008), which makes these methods very effective calibration tools. This methodology has already been applied to a variety of hydrogeological problems including assimilation of hydraulic head data (Chen and Zhang, 2006; Nowak, 2009), transport problems (Liu et al., 2008; Li et al., 2012), surface water-groundwater interactions (Kurtz et al., 2014; Rasmussen et al., 2015; Tang et al., 2015), assimilation of discharge data (Camporese et al., 2009), operational flood forecasting (Seo et al., 2009; Weerts et al., 2010) and integrated hydrological modelling (Shi et al., 2015; Rasmussen et al., 2015; Kurtz et al., 2016). An application in a hydrogeological setting was given by Hendricks Franssen et al. (2011) for groundwater management of the upper Limmat aquifer in Zurich (Switzerland). In this case, a groundwater model is run on a daily basis to support management decisions on groundwater abstraction, and the EnKF methodology is used to continuously correct the model predictions and model parameters with available hydraulic head data. These corrected model predictions are then used as input for the real-time optimisation of groundwater management activities (Bauser et al., 2010, 2012). In the particular case of the Limmat aquifer, the updated hydraulic head distribution from the groundwater model is used to optimally control the groundwater abstraction from a well field according to predefined management goals, which include the total abstraction rate and the maintenance of certain hydraulic conditions to prevent the leakage of contaminants to the well field from a close-by disposal site. Other applications of real-time optimisation of groundwater resources include the energy efficient operation of well fields (Hansen et al., 2012; Bauer-Gottwein et al., 2016) or the accounting for the thermal regime within an aquifer (Marti, 2014). In Schwanenberg et al. (2011), data assimilation methods are used in conjunction with optimal control algorithms for a large-scale river network.

Such methods, especially in combination with fully-coupled, physically-based hydrological models, are usually associated with a high computational burden due to the need to perform hundreds of model simulations in a Monte Carlo framework. This requires the availability of a dedicated computer infrastructure, which is not readily available for every end-user due to the high personal and financial effort for acquiring and maintaining such systems. This can, in part, be overcome by cloud-based services that provide computational resources on demand, which is seen as an upcoming solution for different environmental applications (Granell et al., 2016). Cloud computing has already been suggested as a future platform for hydrological modelling, model calibration and uncertainty analysis (Hunt et al., 2010; Bürger et al., 2012; Ercan et al., 2014; Zhang et al., 2016) and as a promising tool in the context of decision making in water management (Sun, 2013). Mure-Ravaud et al. (2016) recently also presented an example of a flood forecasting system which is hosted on a cloud server. Such cloud-based solutions are flexible with respect to the choice of the computing environment (operating system, CPU, main memory, etc.), and can thus host a variety of simulation platforms with different computational requirements. Furthermore, such services are paid according to the actually consumed computation time. Therefore, the costs to the end-user are effectively reduced by avoiding the financial overhead that is required for installing and maintaining an own in-house computer system.

This study presents a fully-operational architecture for a cloud-based stochastic real-time prediction and management system in the context of groundwater management. The proposed system

allows to perform data assimilation with the fully-coupled, physically-based hydrological model HydroGeoSphere in a cloud environment, and to use the improved model predictions for the optimisation of groundwater management. It also includes a link to the online acquisition of field data, which transfers the measurement data acquired in the field to a cloud-based storage system. To the best of our knowledge, this is the first time that such a comprehensive system for stochastic real-time water resources management is described in the context of cloud computing. A second main contribution is that the real-time simulation and management is done with a fully-coupled, physically-based hydrological model, which is challenging in the context of data assimilation due to the large amount of required computational resources.

In the next section, a detailed overview of the different components of this cloud-based real-time monitoring and modelling platform is provided. In section 3, we apply this platform to a synthetic groundwater management problem in order to demonstrate the efficiency and value of this approach and to investigate the performance of the cloud-based solution in comparison to a more conventional (cluster-based) execution environment. Ultimately, we draw conclusions and provide recommendations for using this platform in real-world applications.

2. Conceptual framework

The cloud-based real-time monitoring and simulation platform outlined in this paper offers two principal functionalities to water resources managers: (i) real-time access to field measurement data, and (ii) real-time stochastic simulations, which are continuously improved by assimilating the most recent field measurements, with the possibility for additional real-time control of water resource systems. The first functionality includes the full process chain from acquiring the measurement data with online sensors, collecting these data via wireless sensor networks and transferring the data to online storage that can easily be accessed by the end-user. This allows decision makers to monitor the measured key hydrological variables in real-time. The data stored in the online cloud storage also serve as a basis for adjusting hydrological model predictions with data assimilation techniques. The monitoring process chain used in our cloud-based real-time monitoring and simulation platform has been described in detail by Lapin et al. (2014) and is summarised in section 2.1. The second functionality is the stochastic real-time prediction of hydrological variables with a physically-based, distributed hydrological model that is corrected by the assimilation of measurement data from the sensor network. This module is the main focus of this paper and aims to support management decisions by providing a probabilistic nowcasting of the system states. This real-time, cloud-based simulation system consists of several components: (i) an interface for the user to orchestrate the cloud-based simulation service and to operate on the input/output, (ii) a cloud-based computational infrastructure that hosts and manages the execution of the simulation code, and (iii) a software that manages the stochastic forward simulations and performs the assimilation of measurement data. These system components are described in detail in sections 2.2, 2.3 and 2.4.

2.1. Data acquisition and management

WSNs and Wireless Mesh Networks (WMNs) can be used to acquire and collect hydrological sensor data, especially in remote locations. For example, in a typical alpine or pre-alpine watershed rainfall intensity can vary strongly between sub-catchments, and measuring these variations properly is crucial for a correct estimation of the water budget of the whole watershed. Accessing such sub-catchments, and even more so, installing and connecting all the sensors, however, can be a big challenge. Additionally, cellular network coverage is often not provided in such environments. WSNs have proven to be a viable solution in this context due to their portability and relatively low installation costs. A WSN represents a set of smart devices, called sensor nodes, which are equipped with environmental sensors and transmit data from the environment in which they are deployed (Kropf et al., 2014; Lapin et al., 2014). WSN systems work well on smaller scales of up to a few hundred meters. On larger distances, i.e., at the scale of a few kilometres, transporting information from the sensor networks in the field to a fixed Internet network requires an additional WMN infrastructure (Jamakovic et al., 2013). Fig. 1 illustrates a typical data-flow from a WSN/WMN to the end-user. Data from a WSN is delivered through an intermediate WMN to an Internet Gateway, connecting the WMN with the wired Internet. The data can then be stored in an online storage system and later accessed by an end-user or any kind of processing software over the Internet. Such a data acquisition and management system has been presented by Lapin et al. (2014) and provides the first cornerstone of our cloud-based real-time monitoring and modelling platform.

2.2. HydroGeoSphere

HydroGeoSphere (HGS) is a state-of-the-art numerical code for the fully-integrated simulation of surface water, groundwater, and vegetation processes (Brunner and Simmons, 2012; Aquanty Inc. 2016). It is able to dynamically solve variably-saturated subsurface flow in three spatial dimensions and uses a fully-consistent coupling between surface and subsurface flow equations. It has been applied successfully for the simulation of highly complex hydrological systems, such as large-scale solute transport (Blessent et al., 2011), transport in complicated fractured systems (Blessent et al., 2014), systems with natural and anthropogenic structures (De Schepper et al., 2015), as well as interactions between surface water discharge, groundwater recharge and tree ring growth (Schilling et al., 2014). The prediction of surface water discharge, groundwater levels and fluxes demands that the parameters of HGS are calibrated, and that an uncertainty analysis is carried out subsequently. This in turn requires running many instances of HGS in a Monte Carlo framework, which makes HGS perfectly suited for parallelisation and implementation in a cloud-based infrastructure.

2.3. Data assimilation with HydroGeoSphere

Data assimilation within the cloud-based modelling system is realised via the ensemble Kalman filter (Evensen, 1994; Burgers et al., 1998). As already mentioned in the introduction, this method uses a Monte Carlo approach to approximate the uncertainty of the model predictions. This is achieved by forward simulations of an ensemble of model realisations, which can differ with respect to the initial conditions, forcing terms and model parameters. The model state vector $\mathbf{x}$ for each model realisation i at time step t (where observations are available) is derived by forward propagation of the dynamical model M using as input the state vector from the previous time step $(t-1)$ and parameters $\mathbf{p}$ and forcings $\mathbf{q}$. Parameters and forcings are different for each model realisation:

$$\mathbf{x}_i^t = M\left(\mathbf{x}_i^{t-1}, \mathbf{p}_i, \mathbf{q}_i\right) \tag{1}$$

The dynamical model M is the fully-coupled, physically-based hydrological modelling software HGS in our case. The prognostic variables of HGS can be hydraulic head, temperature and river discharge, amongst others. In this study, only hydraulic heads are

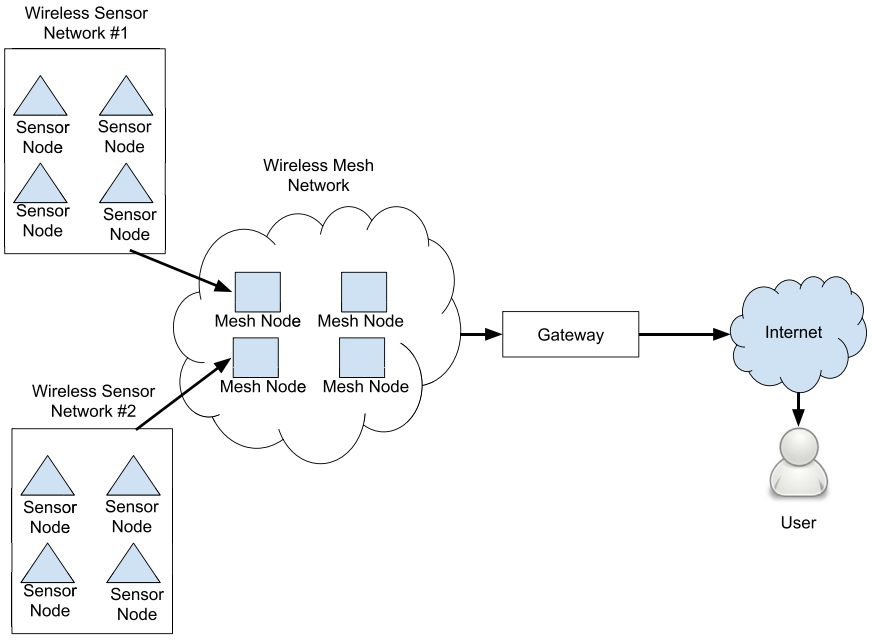

Fig. 1. Typical data-flow from a Wireless Sensor Network.

simulated and therefore the model state vector $\mathbf{x}$ only consists of this prognostic variable. The measurement equation, which links measured and modelled states, is given by:

$$\mathbf{y}_i^t = \mathbf{H}\mathbf{x}_i^t \tag{2}$$

where $\mathbf{y}$ is the vector containing the measurements and the matrix $\mathbf{H}$ extracts or interpolates the state vector onto the observation locations. In order to account for measurement uncertainty, the actual observations $\mathbf{y}^0$ for time step t are perturbed with values drawn from a normal distribution N with a mean of zero and a standard deviation corresponding to the estimated measurement error ε:

$$\mathbf{y}_i^0 = \mathbf{y}^0 + N(0, \varepsilon) \tag{3}$$

The updated state vector $\mathbf{x}^+$ for each realisation i is then given by:

$$\mathbf{x}_i^+ = \mathbf{x}_i^t + \mathbf{K}\left(\mathbf{y}_i^0 - \mathbf{y}_i^t\right) \tag{4}$$

where the Kalman gain matrix $\mathbf{K}$ is given by:

$$\mathbf{K} = \mathbf{C}_{xy}\left(\mathbf{H}\mathbf{C}_{xy} + \mathbf{R}\right)^{-1} \tag{5}$$

where $\mathbf{C}_{xy}$ is the covariance matrix between the model states $\mathbf{x}$ and the predicted observations $\mathbf{y}$ and $\mathbf{R}$ is the covariance matrix of the measurement errors.

If the measurement errors can be considered independent the matrix $\mathbf{R}$ consists of diagonal elements representing the individual measurement errors of the observations. As can be seen from Eq. (5), the Kalman gain matrix $\mathbf{K}$ weights the uncertainties in the model predictions (represented by $\mathbf{C}_{xy}$) with the measurement errors (represented by $\mathbf{R}$). These weights are used in the updating equation (Eq. (4)) to correct the model prediction $\mathbf{x}^t$ with the perturbed measurements $\mathbf{y}^0$. This state vector can contain several model variables that are then updated with the measurements $\mathbf{y}^0$. This might also include model parameters like hydraulic conductivity or porosity. For the examples presented in this study, hydraulic head is the only model variable that is used to form $\mathbf{x}^t$.

Data assimilation was implemented in combination with the HGS model in a C program called EnKF-HGS, which manages the forward propagation of the ensemble of HGS model realisations (Eq. (1)) and performs the update of the simulated state (or state-parameter) vector with the measurements (Eqs. (2)–(5)). As the forward propagation of a large ensemble of a highly sophisticated hydrological model like HGS is very CPU intensive, the program is parallelised with respect to the ensemble forward propagation and the filtering step in order to speed up calculations. The parallelisation is done by the distribution of different realisations among available CPUs so that each CPU handles the forward propagation and updating steps of a specific subset of the data assimilation problem. The interfacing between EnKF and HGS is done via the input and output files of HGS. The input of HGS usually consists of a text file that contains all the information to perform the simulation which includes, for example, the definition of the computational grid, the assignment of boundary conditions, solver settings and the definition of initial model states and model parameters. Simulations with HGS are then usually performed by first running a pre-processor called 'GROK', which translates the settings specified in the input text file into binary input files which are the actual input used for the HGS simulator. Certain parts of the text input file for the pre-processor GROK (e.g., transient boundary conditions, definition of parameters, etc.) can be sourced out into separate text input files. This feature is used for defining stochastic transient boundary conditions and stochastic model parameters for HGS in combination with EnKF-HGS. The main program EnKF-HGS writes these specific input files for boundary conditions and parameters separately for each model realisation and each time step. This requires a proper definition of the respective boundary conditions and parameter settings in the text input files for the pre-processor GROK, i.e., all the input information that should be treated as stochastic needs to be sourced out in separate input files. The exchange of prognostic variables (i.e., hydraulic head) between HGS and EnKF-HGS is managed via binary input/output files that are

used by HGS. These binary files are written by EnKF-HGS as an input for HGS for the next model integration. After the HGS-simulation has finished, the predicted state variables are extracted by EnKF-HGS from the binary output files of HGS. This information is later used in the ensemble of state vectors of the data assimilation algorithm. The complete program scheme of EnKF-HGS is summarised in Fig. 2. Note that the ensemble forward propagation as well as the filtering step are calculated in parallel, meaning that each available processor handles a subset of the ensemble members. The program was first built and tested on a standard Linux cluster and then slightly modified to be able to also run in a cloud environment, which is described in the following section.

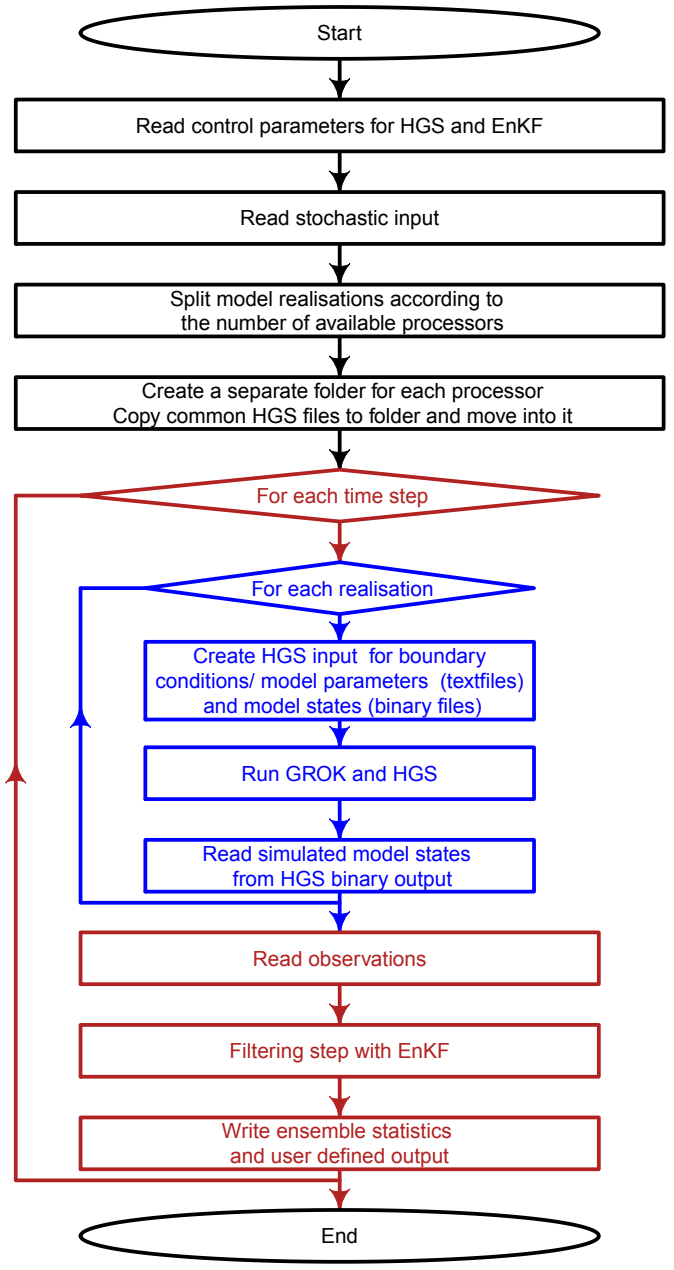

Fig. 2. Schematic program execution of EnKF-HGS. Black colours mark the initialisation phase of the program, blue colours indicate the ensemble calculations and red colours represent the model forward propagation and filtering step. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

2.4. Cloud computing with CLAUDE

Cloud computing (Varia, 2008) is a still evolving paradigm, in which various computing resources are delivered to an end-user as services (e.g., Platform as a Service (PaaS), Infrastructure as a Service (IaaS), Software as a Service (SaaS), or XaaS (Everything as a Service)). Virtualisation is a fundamental element of cloud computing. In computing, the term *virtualisation* refers to provisioning of a virtual resource (e.g., a computer network resource, a storage device, a server) that behaves identically to its physical counterpart. Virtualisation allows for physical resource sharing. Virtual resources access the infrastructure through a specialised middleware (i.e., a hypervisor) to improve the infrastructure flexibility and elasticity. While end-users experience cloud computing as an easy-access and easy-to-use technology, cloud computing is a complex and powerful technology with a multitude of important characteristics (Mell and Grance, 2011):

- Rapid elasticity and scalability in near real-time via dynamic resource provisioning upon a self-service basis, which allows to adapt system resources to momentary load conditions on-the-fly.
- Measured resource consumption: Cloud computing resource usage can be transparently measured, controlled, and reported by both the provider and the consumer.
- On-demand self-service: The consumer can purchase various services such as applications, networking, or computing using little interaction with the cloud provider.
- Broadband access to the cloud infrastructure is provided by the Internet making use of standard network infrastructures and protocols.

Fig. 3 presents a simplified scheme of a) traditional computation and b) typical cloud infrastructures. Traditional computation infrastructure is usually provided by the users themselves, which often results in one of two extremes: (i) Overprovisioning, i.e., a waste of money on the maintenance of non-utilised computational resources, or (ii) Underprovisioning, i.e., limited computing performance due to missing storage or processing infrastructure. Contrarily, cloud users do not need to care about the infrastructure capacity, as they only temporarily purchase computing resources from a cloud provider. The cloud provider is therefore responsible for the maintenance of resources dimensioned for the needs of all the cloud-users.

Cloud computing offers SaaS as a major service delivery model. SaaS may provide applications that use a Web browser as their user interface, or predefined Application Program Interfaces (APIs). In the following, we describe our cloud-based simulation service, which we hereinafter refer to as CLAUDE (see also Lapin et al., 2015), that implements a typical provisioning model for applications delivered as SaaS. CLAUDE provides an easily migratable SaaS supporting a wide range of cloud platforms (e.g., Amazon EC2 (Amazon Web Services, Inc. 2016), OpenNebula (Open Nebula Project, 2016), OpenStack (Open Stack Foundation, 2016)), while at the same time allowing users to benefit from the features of cloud environments for ordinary, non-interactive, computation-intensive applications commonly used in the high performance computing domain.

A basic CLAUDE deployment requires a physical or virtual machine for the installation of the CLAUDE core, access to a cloud data storage that supports the Amazon S3 protocol (Amazon Web Services, Inc. 2016) for input/output data storage, and access to a cloud resource pool through an Amazon EC2 endpoint. CLAUDE automatically distributes work among available resources, and allows an end-user to balance between minimising the execution time by

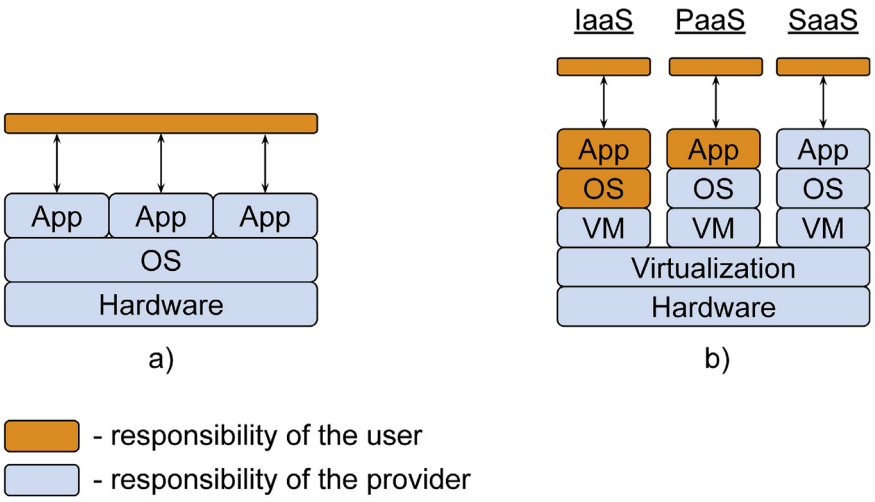

Fig. 3. (a) Traditional computation infrastructure layers including hardware, operating system (OS) and applications (App); (b) Typical cloud infrastructure layers including virtualisation with virtual machines (VM) for three different usage types: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS).

exploiting parallel execution (e.g., for Monte Carlo applications) and the final costs by minimising the amount of consumed resources. CLAUDE allows including data from external data sources (e.g., online databases, WSNs). Fig. 4 depicts the integration of the EnKF-HGS real-time modelling using CLAUDE in a cloud environment with a sensor-based live data-flow. The user interacts with the system through a web interface and controls the simulation process (e.g., initialisation, monitoring), the input/output, and the amount of consumed cloud resources. CLAUDE, in turn, orchestrates the running applications on the available computing resources.

Accessing and launching EnKF-HGS through CLAUDE has been specifically designed in a very user-friendly way, so that no prior knowledge of running applications on a cloud is required. Furthermore, CLAUDE can easily be generalised for other non-interactive user applications. Once the CLAUDE core is installed and configured, one needs to prepare an Operating System (OS) image for the Virtual Machine (VM) with the required software installed (in our case EnKF-HGS and HGS) and uploading the OS image to the cloud repository. The image is used to create virtual machines, which allow executing the pre-installed software. The uploading process may slightly differ depending on the cloud

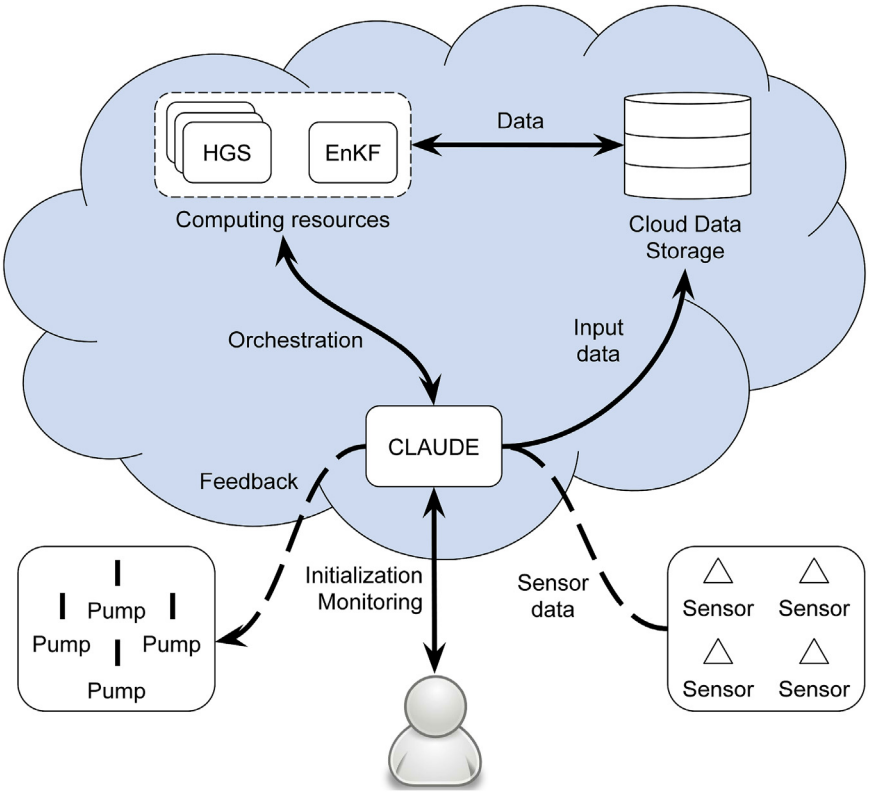

Fig. 4. CLAUDE-based setup of the EnKF-HGS modelling system.

platform, but the most popular providers (e.g., Amazon EC2, OpenNebula, OpenStack) have step-by-step tutorials explaining the whole process. Additionally, the user is expected to provide the CLAUDE core with a sample Python script that defines execution parameters and input/output locations. After completing all these steps, the system is prepared for execution. As an advantage, CLAUDE provides a set of communication drivers, which implement communication operations in the cloud. This minimises the necessary changes of the original application source codes. The drivers interact with the CLAUDE core through a standard messaging mechanism provided by the widely used message broker RabbitMQ (Present Pivotal Software, Inc. 2007). CLAUDE therefore allows to easily port an existing application, which was not designed for a cloud environment, to a cloud environment.

In order to port EnKF-HGS to a cloud environment, the CLAUDE driver was integrated in the program source code, and all the direct execution calls of HGS were replaced with their CLAUDE-based counterparts. This "cloudification" procedure changes the MPI-based execution stack of EnKF-HGS. The cloudified EnKF-HGS separates the main EnKF-HGS simulation loop from the HGS instances, which run on remote computing resources. For this purpose, the CLAUDE driver transmits the relevant input for each HGS model realisation to the cloud data storage, and requests the CLAUDE core to execute corresponding simulations on available cloud computation resources. When all the simulations are completed, the CLAUDE driver retrieves the relevant output from the cloud data storage. The data is then returned to EnKF-HGS, which continues the data assimilation loop. This service oriented approach allows dynamically adjusting the amount of computing resources to momentary workload conditions. Moreover, one can benefit from an external scheduling component for better resource utilisation. Additionally, CLAUDE allows minimising the network Input/Output (I/O) operation using data locality.

3. Validation example

For the demonstration of the cloud-based real-time modelling system, the well-known tilted V-catchment problem was selected, because it is often used as a benchmark for integrated hydrological models (e.g., Panday and Huyakorn, 2004; Kollet and Maxwell, 2006) and has also been applied in the context of data assimilation with integrated hydrological models before (e.g., Camporese et al., 2009; Bailey and Baù, 2012). The internal flow dynamics of the tilted V-catchment model have furthermore been systematically analysed in original and modified configurations (Gaukroger and Werner, 2011). The original model consists of a V-shaped valley, which is inclined in order to form a river at the bottom of the valley. The model is forced with one or more precipitation events followed by a recession phase, resulting in distinct discharge peaks at the catchment outlet. The temporal evolution of these peaks also depends on the assigned surface flow properties and the exchange between the surface water and the groundwater domain. Similarly to Gaukroger and Werner (2011), the original V-catchment model was modified to better fit the purpose of this study. First of all, the tilted V-catchment model was extended to allow for more substantial surface water-groundwater interactions: Instead of using the original setup that consisted of a 3 m deep soil-layer to represent the subsurface, a permeable aquifer of 33 m vertical extent was used, allowing significantly more groundwater dynamics. Secondly, the model was enhanced with eight groundwater wells that simulate typical alluvial groundwater extraction processes for drinking water purposes in the vicinity of the river. With these modifications to the original model, the modified tilted V-catchment model allows the simulation of substantial surface water-groundwater interactions in a typical alluvial setting with realistic drinking water management. For the synthetic data assimilation experiment, a reference HGS simulation with the modified tilted V-catchment model provides observation data (i.e., hydraulic heads) that are subsequently used to correct stochastic model predictions with the EnKF. In addition to the data assimilation aspect, the synthetic experiment used in this study also includes a groundwater management component, where the corrected stochastic model predictions are used to control the groundwater withdrawal of the pumps. This setup represents a simplified groundwater management problem at the catchment scale, where ensemble-based real-time simulations are corrected with observed groundwater levels at regular time intervals and are subsequently used to assist decision makers in the management of groundwater resources (i.e., a synthetic representation of a real world system such as presented by Hendricks Franssen et al. (2011) and Bauser et al. (2010, 2012)). The access to measurement data from the field is treated in a simplified manner in this synthetic experiment by assuming that the observation data are already provided to the cloud infrastructure by the sensor network. For real-world cases, the access to data from the EnKF-HGS modelling and data assimilation platform would involve the data gathering system described in Jamakovic et al. (2013) and Lapin et al. (2014), followed by some steps to treat potentially missing values and to assure the quality of each measurement. Such additional pre-processing of the data is usually quite case-specific and is therefore strongly simplified for this synthetic example.

3.1. Hydrological model and data assimilation setup

The modified tilted V-catchment that is used for the data assimilation experiments has a spatial extent of 1620 × 1000 × 33 m and is discretised into 81 × 50 × 11 model nodes. The horizontal discretisation is 20 m in x- and y-direction. The 10 vertical model layers span a total of 33 m and have a variable thickness with increasing resolution towards the top of the model domain (10 m, 10 m, 6 m, 3 m, 2 m, 1.10 m, 0.50 m, 0.25 m, 0.10 m, 0.05 m from bottom to top). The tilted V-catchment topography is formed by assigning a slope of ± 0.05 m/m in x-direction for the western and eastern half of the model. The entire model domain is additionally inclined in y-direction with a slope of −0.02 m/m. The resulting topography for the uppermost model layer is shown in Fig. 5.

A constant head boundary condition of 23.2 m is assigned to the upstream model face (y = 0 m), which provides a temporally constant base flow to the river. A critical depth boundary condition is assigned to the surface water domain at the downstream model face (y = 1000 m). The lateral model boundaries (x = 0 m and 1620 m) are impermeable. Four wells are placed on either side of the river in the downstream part of the model domain, which operate at a constant withdrawal rate of 0.45 m^3/s per well (well locations indicated in Fig. 5). The hydraulic parameters for surface and subsurface flow are spatially constant and are summarised in Table 1. The total simulation time for the model forward runs is 72 h, which is discretised in 144 time steps of 1800 s. The reference run is forced with transient time series for rainfall and evapotranspiration (see Fig. 6). The rainfall time series includes three rainfall events, each with a duration of 6 h and a precipitation rate of 2.6 × 10^{-6} m/s. In between the precipitation events, evapotranspiration follows a diurnal cycle with an amplitude of 2× 10^{-7} m/s and a duration of 12 h. Fig. 6 shows the temporal sequence of rain and recession phases.

The observation data (i.e., hydraulic heads) are taken from the reference simulation. In total, 20 observation points are used. The observations are made on the bottom layer of the model (see Fig. 5), and are assimilated every time step (i.e. every 1800 s) with the EnKF. A measurement error of 0.01 m is assumed for these

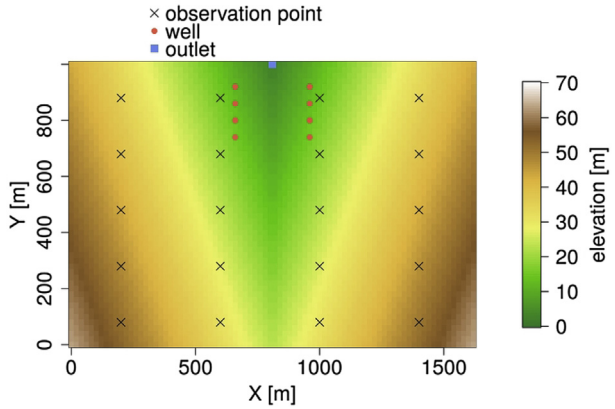

Fig. 5. Topography, pumping wells and observation network for the modified tilted V-catchment model.

Table 1
Model parameters for the modified tilted V-catchment problem.

Variable	Value	Unit
Saturated hydraulic conductivity	2.9×10^{-3}	m/s
Porosity	0.43	$m^3\ m^{-3}$
van Genuchten α	3.48	m^{-1}
van Genuchten n	1.75	–
Residual saturation	0.05	$m^3\ m^{-3}$
Manning's coefficient	0.15	$s/m^{1/3}$
Coupling length	1×10^{-7}	m
Rill storage height	0.1	m

observation data. The ensemble of uncertain model realisations (100 ensemble members) is generated by perturbing the rainfall and evapotranspiration rates with multiplicative noise drawn from a log-normal distribution with a standard deviation of 0.3 (see Fig. 6). Additionally, a constant bias of 10% is added to the precipitation rates. Precipitation input for hydrological models is typically derived from radar or rain gauge data, or a combination of both. The precipitation rates derived from these measurement techniques are usually associated with a considerable uncertainty range (Liu et al., 2012). The uncertainty in the precipitation input and the additional bias used in this setup intends to resemble the uncertainty of these precipitation measurements. Fig. 6 shows the assigned uncertainties of precipitation and evapotranspiration in relation to the reference case. Uncertainties in precipitation are quite large and cover the reference precipitation.

In addition to the assimilation of hydraulic head observations, a simple groundwater management scheme is used in the simulations. This illustrates that the cloud-based EnKF-HGS modelling and data assimilation platform can also be used for real-time adaptation of water resources management, for example to avoid negative impacts of excessive pumping. The example in this paper targets to maintain a minimum 'ecological' river flow by adjusting the well pumping rates based on the simulated ensemble river discharge. The management rule is as follows: If the 30%-percentile of the simulated discharge distribution of the ensemble falls below a threshold value of 1.6 m^3/s, the pumping rates are reduced by 5% in the following time step.

3.2. Simulation results

Three scenarios were considered with the setup described above, with each scenario including an open-loop run (runs without feedback, i.e., no assimilation of hydraulic head data) and an assimilation run: (I) No well management for the reference and the ensemble simulations. This scenario is intended to first verify the effectiveness of the data assimilation within this setup. (II) Well management for the ensemble simulations but not for the reference. This scenario is intended to test the stochastic well management in conjunction with data assimilation with a large bias between observations and model predictions. (III) Well management for ensemble and the reference simulations. In this case, the measurement data from the reference run are also influenced by the well management which would be typically the case in real-world settings.

Fig. 7 shows the temporal evolution of simulated discharge for the open-loop simulation and the assimilation experiment without well management (scenario I). The simulated discharge for the open-loop simulation shows a positive bias (compared to the reference), which is related to the assigned bias of 10% in the precipitation input data. The spread in the simulated discharge is more pronounced during the rainfall events. During the recession phase, however, the positive bias in the precipitation input data leads to a

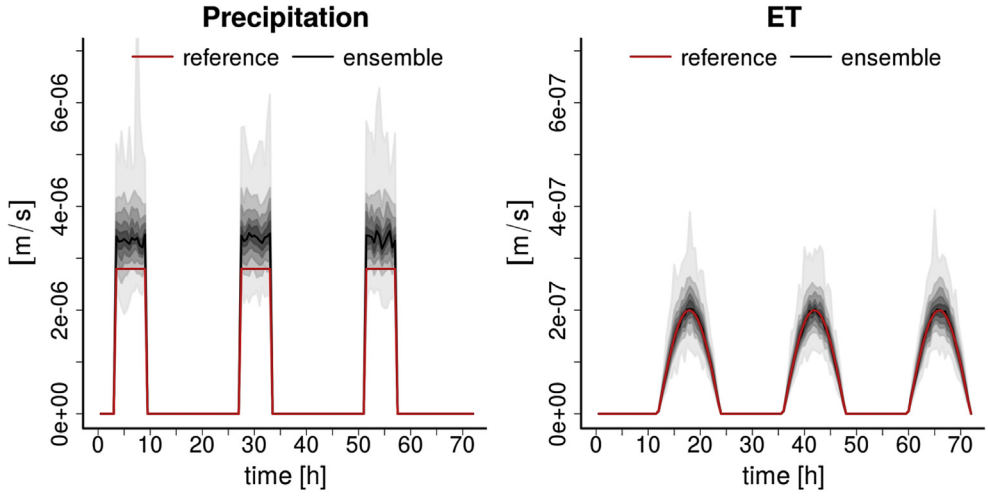

Fig. 6. Precipitation (left) and evapotranspiration (right) rates for the reference run (in red) and the ensemble simulations (in grey). The different grey areas represent 10%-percentiles of the ensemble distribution and black lines represent the ensemble median value. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

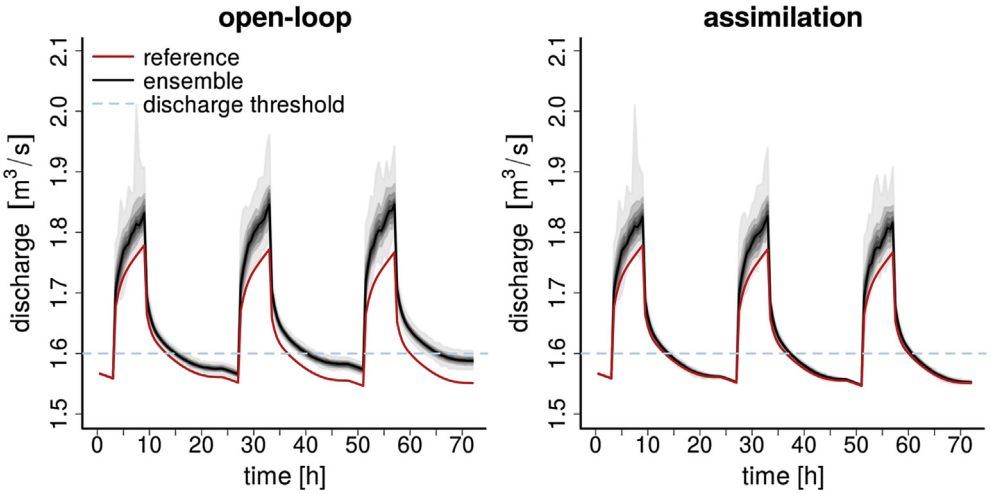

Fig. 7. Simulated discharge for the open-loop simulations (left) and the assimilation experiment (right) of scenario I. The red lines show the discharge for the reference run and the dashed blue lines mark the discharge threshold for the well management (which, however, was not enforced in this scenario). The different grey areas represent 10%-percentiles of the ensemble distribution, and the black lines represent the ensemble median value. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

mismatch that is increasing with time: in the open-loop simulation the overall groundwater withdrawal through the wells is slightly lower than the precipitation input, compared to the reference simulation. The assimilation of hydraulic head values from the reference run leads to a correction of these discharge values towards the reference discharge, particularly during the recession phases but also during the rainfall events, and especially in the later phase of the simulation period. The correction of simulated groundwater levels towards the observations leads to a correction of the mass balance within the aquifer, which indirectly improves the integrated discharge signal and effectively adjusts it towards the true observations.

The improvement of model predictions was also quantitatively assessed by calculating the Mean Absolute Error (MAE) for hydraulic heads and discharge:

$$\mathrm{MAE}(x) = \frac{1}{N_n\, N_t} \sum_{i}^{N_n} \sum_{j}^{N_t} \left| \bar{x}_{ij} - x_{ij}^{\mathrm{ref}} \right| \tag{6}$$

where N_n is the number of model nodes (only used in case of hydraulic heads), N_t is the number of model time steps, $\bar{x}_{ij}$ is the simulated ensemble mean of hydraulic head or discharge and x_{ij}^{ref} is the corresponding reference value. For scenario I (no well management), the MAE of hydraulic heads improved from 0.48 to 0.21 m with the assimilation of hydraulic head values. The MAE of discharge reduced from 0.033 to 0.015 m^3/s. Fig. 8 shows the spatial distribution of MAE values for hydraulic heads (averaged over the simulation time) for the open-loop simulation and the assimilation run. It can be seen that the assimilation of hydraulic head data in this setup has a significant positive effect on the predicted hydraulic heads, which leads to a significant reduction of errors in hydraulic head and discharge of about 50% compared to the open loop simulations.

Results for the discharge behaviour of scenario II are shown in Fig. 9. In both the open-loop and the data assimilation cases, the reduction of pumping rates on the basis of the management criterion effectively keeps the simulated discharge around the predefined discharge threshold value of 1.6 m^3/s. In both cases, slight oscillations in discharge occur, which are related to a switching off of the management criterion during the recession phases. When the discharge threshold value of 1.6 m^3/s is reached, the pumping rate of the wells is reduced until the management criterion is satisfied again, which keeps the discharge of the following time steps above the threshold. The lag time between two consecutive activations of the pump management depends on the response time of the groundwater flow field to the pumping in the wells, which is relatively short in this particular case. At certain times, slightly lower discharge values can be observed in the assimilation experiment during the recession phases, compared to the open loop simulation. This is related to the fact that the correction of simulated groundwater levels in the assimilation run leads to a decrease of the simulated discharge towards the values of the reference run (see Fig. 7), which also affects the scheduling of the well management. Overall, the management module provides an effective way to keep the discharge around the predefined level (i.e., minimum ecological flow) even if there is a quite large discrepancy between the ensemble simulations and the actual measurements. The MAE values for hydraulic heads and discharge for scenario II are higher than for scenario I due to the management in the ensemble simulations, which is not present in the reference case. MAE of hydraulic head decreases from 0.48 to 0.28 m and MAE of discharge decreases from 0.05 m^3/s to 0.04 m^3/s in the assimilation run of scenario II.

For scenario III, both the ensemble simulations as well as the reference case include well management. Now, the measurements from the reference case better resemble the aquifer response to the management in the ensemble simulations as compared to scenario II (where measurements are not influenced by the management). However, due to the uncertain and biased precipitation input in the ensemble, there is a discrepancy between the measurements (from the reference run) and the ensemble model predictions, which nonetheless needs to be corrected by EnKF. In Fig. 10, one can see that the open-loop simulations and the reference show a very similar behaviour during the recession phase where management takes place. For the open-loop simulations the predicted discharge is slightly higher and the amount of correction with the management module is lower than in the reference case, which is due to the positive bias in precipitation. When measurement data are assimilated (right hand side of Fig. 10), the ensemble discharge prediction in the recession phase is almost identical to the reference, which shows that the precipitation uncertainty can effectively be corrected for by data assimilation in combination with the

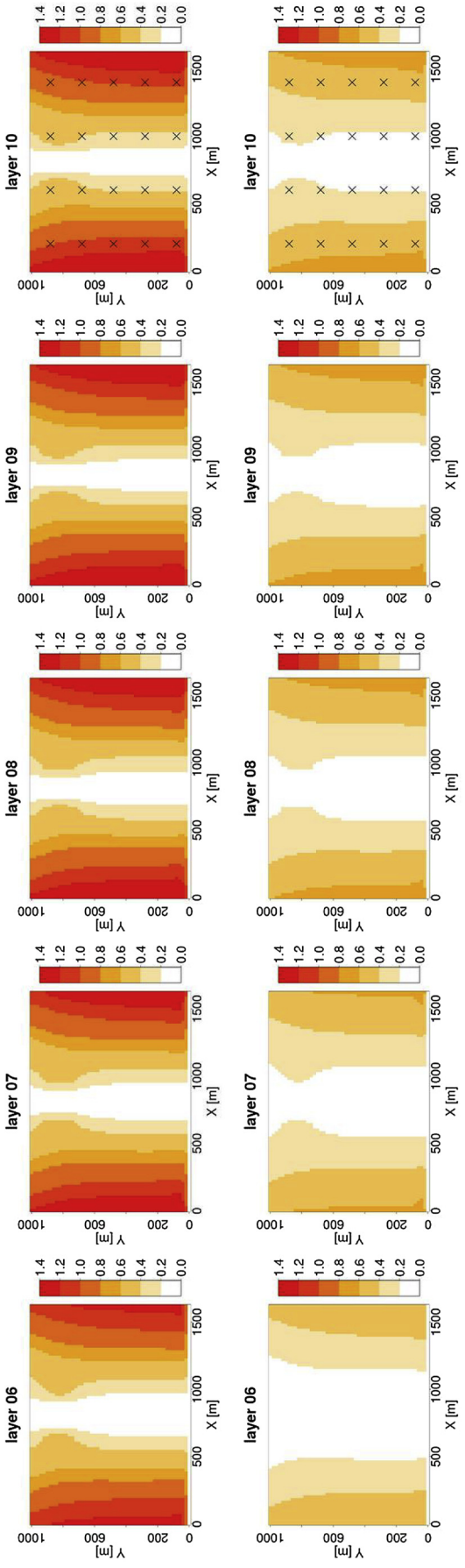

Fig. 8. Mean absolute error of hydraulic heads for the five uppermost model layers for the open-loop simulation (upper row) and the assimilation run (lower row) of scenario I.

management module. For scenario III, the MAE values are reduced to a similar magnitude as in scenario I, with a reduction from 0.48 to 0.21 m for hydraulic heads and 0.024 to 0.015 m^3/s for discharge.

3.3. Performance analysis of the EnKF-HGS modelling and data assimilation platform in a cloud environment

The EnKF-HGS simulator uses a Monte Carlo approach with an iterative workflow. In every iteration, two phases can be identified: (i) simulation of an ensemble (i.e., a set) of model realisations (referred to as the forward simulation phase), and (ii) an update, or feedback, phase for state vectors (and optionally parameters) (referred to as the filtering phase). One forward simulation phase computes all individual model realisations of the ensemble. Computing a single model realisation requires executing two applications, i.e., GROK and HGS. The run-time of each HGS realisation strongly depends on input parameters and model complexity, but normally, HGS is a comparably long-running and compute-intensive process due to a high non-linearity of the simulated hydrological processes. Moreover, due to a large number of parallel model realisations in the forward simulation phase, the main demand for computing power in the EnKF-HGS modelling platform comes from the execution of HGS. In the cloudified version of the EnKF-HGS modelling platform, the forward simulation phase is therefore distributed over multiple cloud computing resources. The filtering phase remains centralised. The focus of this analysis therefore is on the performance of the forward simulation phase (section 3.3.2) by measuring the execution time of each implementation. However, execution results of the filtering phase are also presented and discussed (section 3.3.3). The execution times of two implementations of the EnKF-HGS platform are directly compared in sections 3.3.4 and 3.3.5. In particular, CPU utilisation, scheduling, and data locality aspects are discussed.

3.3.1. Setup of performance comparison

As discussed in section 2.4, the original EnKF-HGS was designed for execution in a traditional cluster environment (i.e., MPI-based), which normally provides: (i) a low latency broadband network (e.g., InfiniBand, Myrinet 10G), (ii) access to a Network File System (e.g., NFS), and (iii) homogeneity of computing resources. None of these properties are guaranteed in a typical cloud environment. This can result in a dramatic performance loss of EnKF-HGS. In order to properly compare the performance of the MPI-based and the cloudified EnKF-HGS, both versions were executed on a local computing infrastructure, rather than the cloudified version being executed on a different infrastructure. Two distinct execution approaches can be identified: Approach 1: an execution of the original MPI-based EnKF-HGS in a virtual cluster environment (i.e., a single VM resource pool), and Approach 2: execution of the cloud-based EnKF-HGS with the CLAUDE driver integrated in a cloud environment. Each approach uses a different network file storage. In Approach 1, a VM-based deployment of GlusterFS (Red Hat, Inc. 2016) was used. The storage nodes were organised in a distributed volume with no data replication employed to maximise storage performance. On the computing nodes, the FUSE-based Gluster Native Client (Red Hat, Inc. 2016) was installed, enabling highly concurrent access to the file system. In Approach 2, a VM-based deployment of Riak CS (Basho Technologies, Inc. 2016) was used.

The modified tilted V-catchment model served as a test case. In the experiment, the model performs 144 iterations with an ensemble of 100 model realisations. Since all realisations are independent HGS instances, a set of 4 experiments with varying numbers of CPU cores was conducted: (a) 10, (b) 20, (c) 50, (d) 100. In each experiment, one single CPU core was assigned per HGS instance. In the original MPI-based implementation of EnKF-HGS,

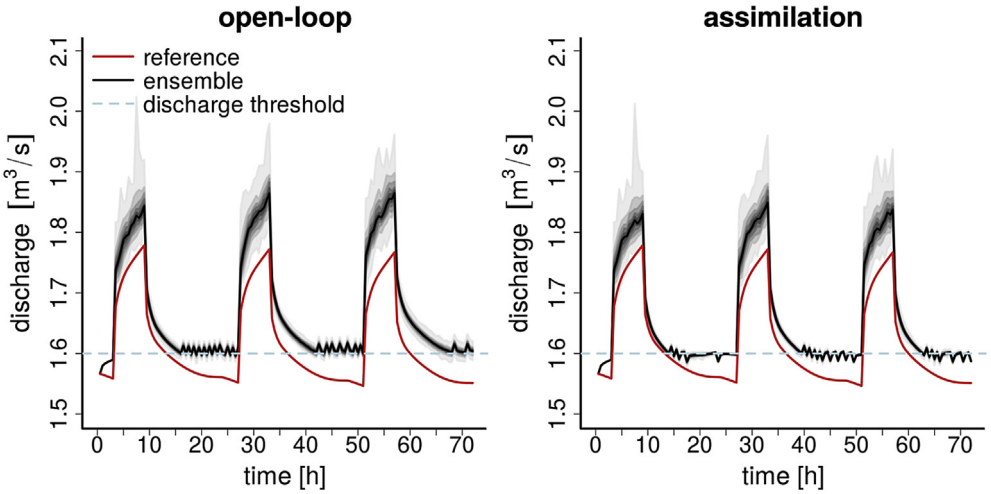

Fig. 9. Simulated discharge for open-loop simulations (left) and assimilation experiment for scenario II. The red lines show the discharge for the reference run and the dashed blue lines mark the discharge threshold for well management. The different grey areas represent 10%-percentiles of the ensemble distribution, and the black lines represent the ensemble median value. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

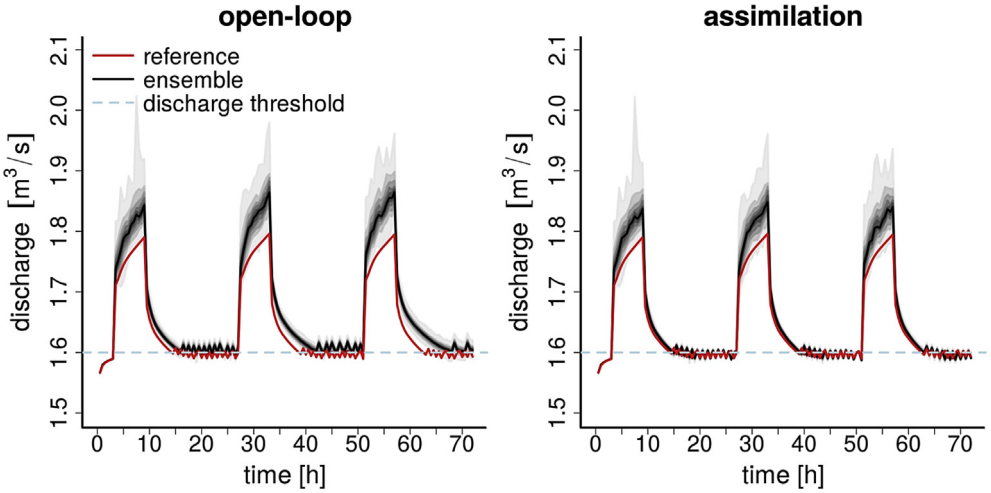

Fig. 10. Simulated discharge for open-loop simulations (left) and assimilation for scenario III. The red lines show the discharge for the reference run and the dashed blue lines mark the discharge threshold for well management. The different grey areas represent 10%-percentiles of the ensemble distribution, and the black lines represent the ensemble median value. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

this resulted in 10, 5, 2, and 1 HGS executions per CPU core, for (a), (b), (c), and (d) respectively. For the cloudified version of EnKF-HGS, CLAUDE automatically distributed the execution of the HGS instances over available computing resources for every iteration, using the First-In-First-Out (FIFO) algorithm, which maximised resource utilisation. Table 2 summarises the characteristics of selected VM instances for both execution approaches. All VMs were connected via a1 GB/s network.

3.3.2. *Performance analysis of the forward simulation phase*

In this first experiment, only the duration of the forward simulation phase is considered. Fig. 11 illustrates the computing time ratio between Approach 1 and 2 against the iteration step. In Approach 1, this time is defined as the elapsed time between the beginning of the forward simulation phase and the end of the longest running MPI process. In Approach 2, it is the time interval between the beginning of the forward simulation phase and the end of the last finishing HGS instance. In Approach 2, the overhead of input/output transmissions and cloud scheduling is therefore also accounted for. The execution time of the forward simulation phase for the two approaches is comparable. A small difference of 4–15% in favour of the original MPI-based approach can be observed in experiments (c) and (d).

3.3.3. *Performance analysis of the filtering phase*

In contrast to the forward simulation phase, the filtering phase was not distributed, as it requires significantly less computing power and thus much shorter execution times. The EnKF-HGS modelling platform performs filtering on initially allocated system resources. In Approach 1, it includes the entire resource pool ranging from 10 to 100 CPU cores. In Approach 2, only 1 CPU core is initially provided to the filtering step (see Table 2). The relative execution time ratios of the filtering phase of Approaches 1 and 2 are shown in Fig. 12. Surprisingly, in (d), the filtering phase running on 100 CPU cores is 20% slower than on a single CPU. This difference in performance is a result of the network connection of a virtual cluster in the cloud environment. A regular physical cluster guarantees high throughput and low latency. On a cloud, physical

Table 2
VM instance selection for the execution of the original MPI-based implementation and the cloudified version of the EnKF-HGS modelling platform. As the memory requirement per HGS instance for the test model is less than 500 MB, the available memory per computing VM is sufficient.

VM role	Approach 1: MPI-based version		Approach 2: Cloud-based version		
	EnKF-HGS platform execution	storage	HGS execution	EnKF-HGS filtering execution	storage
Number of VMs	2/3/7/13	3	2/3/7/13	1	1
CPU cores per VM	8	4	8	1	4
Memory (GB)	7.4	3.5	7.4	2.5	3.5

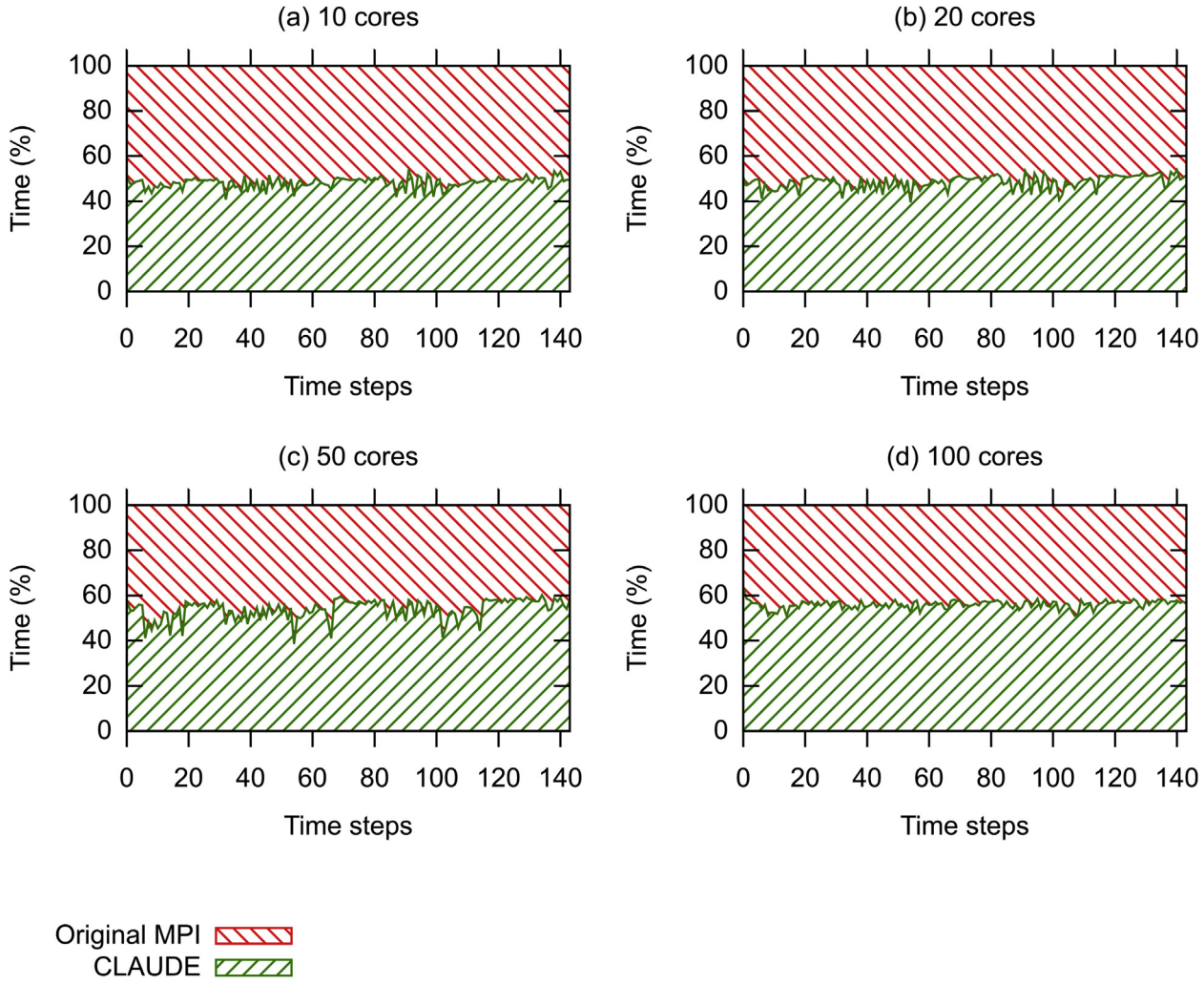

Fig. 11. Relative computation time of the forward simulation phase for the two different execution approaches.

machines can be geographically distributed, and therefore might be connected with links of lower capacity and higher latency. In our experiments, cloud workers were connected with a regular 1 GbE-T Ethernet connection, which is considered slow from the point of view of a regular cluster environment. As a result, the data transmission time between multiple MPI processes surpassed the benefit of parallel computation. The regular, MPI-based EnKF-HGS setup is therefore not suited for cloud computations with heterogeneous or low capacity resources.

3.3.4. Performance analysis of the original MPI-based implementation

In order to measure the impact of the concurrent access to a network file storage on the execution time of the HGS simulator, it needs to be assured that each HGS instance performs the same amount of I/O operations. We therefore use a similar technique for assessing I/O saturation as Caino-Lores et al. (2016), running a similar I/O stress experiment in a cloud setup, which is described in Table 2. In our case, we use the tilted V-catchment problem as the input hydrogeological model. To this end, a modification of EnKF-HGS to compute the same model N times in a row instead of simulating N different realisations was made. The modified EnKF-HGS modelling platform was executed using a range of 1–64 available CPU cores (see Fig. 13) to measure the execution time of a HGS instance in every step against the iteration count. The HGS performance plots resemble the results of Caino-Lores et al. (2016). They are, however, skewed towards longer relative execution times for a growing number of concurrent HGS instances. As shown in

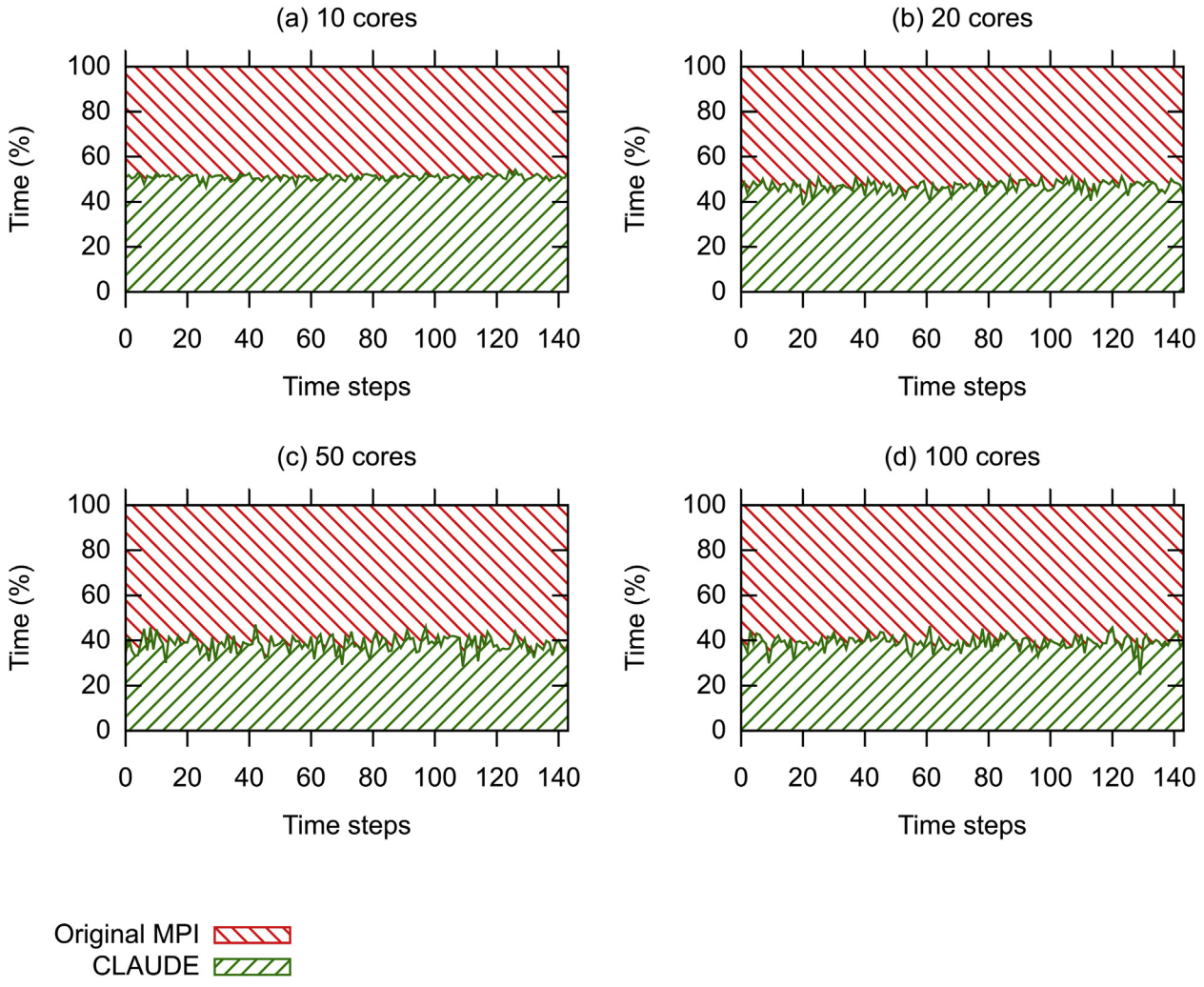

Fig. 12. Relative computation time of the filtering phase for two execution approaches.

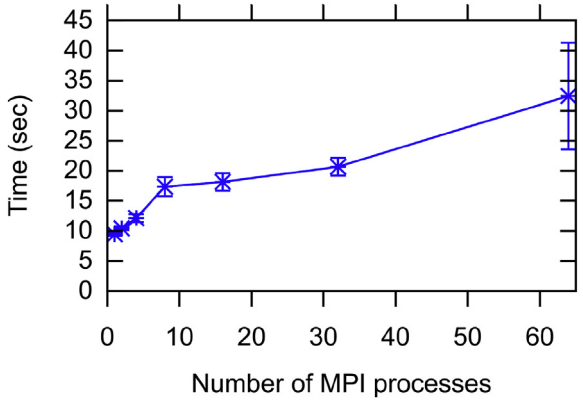

Fig. 13. Impact of the concurrent access to a network file storage on the execution time of the HGS simulator.

Fig. 13, the concurrent access to network file storage significantly affects the performance due to increased I/O latency. This is a result of the saturation of the network link. In this experiment, however, the concurrent I/O operations were artificially equalised. This resulted in a large performance drop as all individual simulations last for the same amount of time and thus highly stress the storage by performing I/O access at the same time. In a realistic ensemble simulation, however, the influence of concurrent execution is less likely to result in such a large number of overlapping I/O operations of parallel HGS instances on input and output.

The second important aspect of the MPI implementation is a statically assigned number of model realisations per MPI process. This is a typical design choice of an MPI-based application due to a lack of load balancing, e.g., through a built-in job-stealing mechanism. In an ensemble with individual simulations of heterogeneous duration, EnKF-HGS might result in relatively poor performance, because some MPI processes finish all assigned simulations sooner than others, and EnKF-HGS needs to wait for all individual realisations to finish before the filtering step can be performed. Therefore, the CPU idle time during the simulation was measured. Fig. 14 shows the CPU idle time per iteration for the four experiments (a), (b), (c), and (d) described above. The majority of the realisations, or ensemble members, have a comparable simulation time, which results in a relatively short CPU idle time. However, in some cases, especially when a single MPI process executes multiple assigned simulations, the CPU idle time grows up to 150 s, which corresponds to 34% of the overall iteration time. For more complex scenarios than the modified tilted V-catchment, with longer simulations runs, the overall execution time of EnKF-HGS could

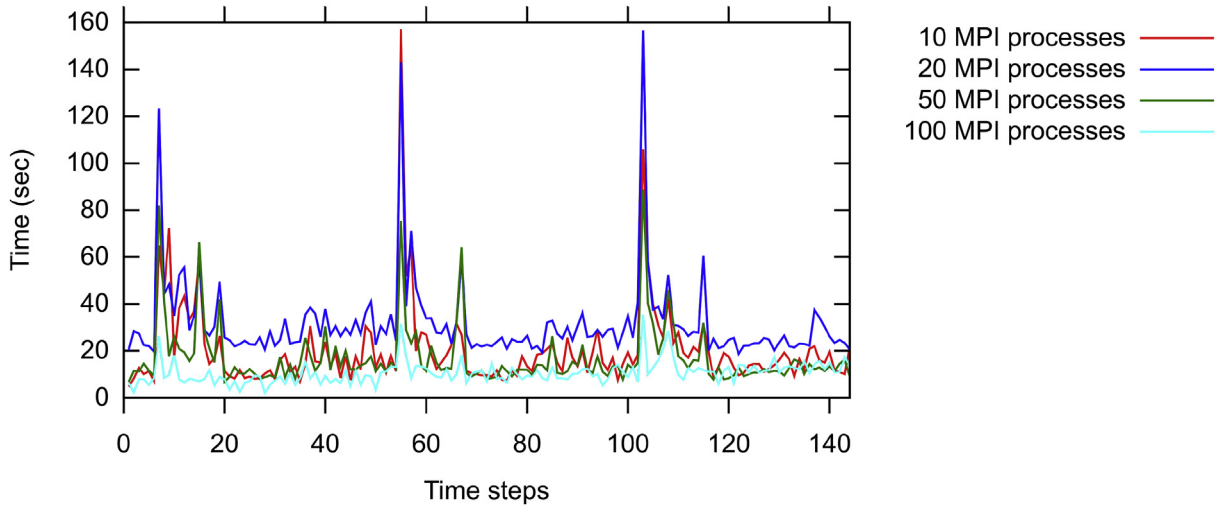

Fig. 14. CPU idle time duration.

increase substantially due to the possibly long wait for all individual realisations to finish.

3.3.5. Performance analysis of the cloudified version

The forward simulation phase in the cloudified EnKF-HGS accommodates three individual, time-consuming processes: (i) input/output data transmission, (ii) scheduling, and (iii) the simulation phase. In the initial four experiments, the duration of every phase was measured. Moreover, we separated the communication overhead (i.e., worker overhead) of the HGS computing resources with the CLAUDE core from the actual simulation time. Fig. 15 shows the relative duration of all these processes. The time spikes in the simulation phase correspond to long-running realisations. Fig. 16 shows the absolute overhead. The I/O data transmission time is almost constant for the full ensemble execution, as well as, for the duration of each individual realisation. The total overhead slightly decreases with the growing number of the parallel instances of HGS, but is constant for the duration of one single realisation. The scheduling time grows with the amount of available resources, however, is constant for the duration of each realisation. In practice, this allows parallel execution of complex hydrogeological models with long-running individual realisations in the cloud, while still having comparable constant overhead. While CLAUDE is still in a proof-of-concept phase, it already indicates the benefits of the service-oriented execution approach with external scheduling and enhanced data locality techniques. By using an I/O-optimised implementation and a high-performance scheduler the total overhead could be significantly reduced, as at the moment it corresponds to 40%–80% of the total overhead.

4. Discussion

The data assimilation and modelling platform described in this paper provides a fully-operational architecture for a real-time water management system using the integrated, physically-based hydrological model HydroGeoSphere in a cloud computing environment. This system is able to provide stochastic predictions of key hydrological variables, which can be continuously improved by assimilating real-time measurement data acquired through wireless sensor networks. The stochastic predictions can then be used to optimise water resources management in a subsequent step. As individual components of such a real-time modelling system like online data acquisition, data assimilation, or real-time optimisation have already been described in literature (e.g. Bogena et al., 2010; Hendricks Franssen et al., 2011; Bauser et al., 2012), this paper is intended to show the integration of these different methodological advances into an overarching water management framework. An important aspect of this overarching framework is the deployment of such a system in a cloud computing environment. Given the fact that stochastic simulations with integrated hydrological models are usually very CPU-intensive, practitioners in water management may often shy away from the investments in computational infrastructure that is necessary for performing stochastic predictions with such kinds of models. However, integrated hydrological models are of pivotal importance for many water management problems, especially for systems where surface water-groundwater interactions take place. In such systems, integrated models have a clear advantage over traditional groundwater models due to their physically more consistent process description of water exchange and surface water routing, which enhances, e.g., the prediction of discharge. In addition, stochastic model predictions in conjunction with data assimilation techniques allow to assess the uncertainty of model predictions and to incorporate field measurements sequentially into the modelling process, which is particularly useful in the context of flood forecasting and contaminant transport.

Cloud technology has the advantage that the computation time for such kind of simulations can be requested on an on-demand basis, meaning that there is no financial and personal overhead for acquiring and maintaining the required computational infrastructure. In this work, the initially cluster-based EnKF-HGS data assimilation system for HydroGeoSphere was translated to a cloud infrastructure with only minimal changes in the original model code and a linkage to pre-existing management tools for the cloud-based computation. These implementation changes were relatively straightforward as EnKF is a Monte-Carlo-based data assimilation approach. In the Monte Carlo approach an ensemble of slightly differing model realisations is integrated forward in time to approximate the uncertainties and mutual correlations within the model. This ensemble of simulations can easily be distributed among different CPUs or nodes. A cloud infrastructure, therefore, provides an ideal environment for such tasks. These findings are also promising for the usage of other, already existing, simulation and data assimilation platforms, since we have been able to show that only a minimal effort is required to port such systems for the use in a cloud infrastructure. Our comparison of the performance of the EnKF-HGS data assimilation platform in a cloud-based and cluster-based computation environment showed that the cloud implementation produced an affordable overhead with respect to

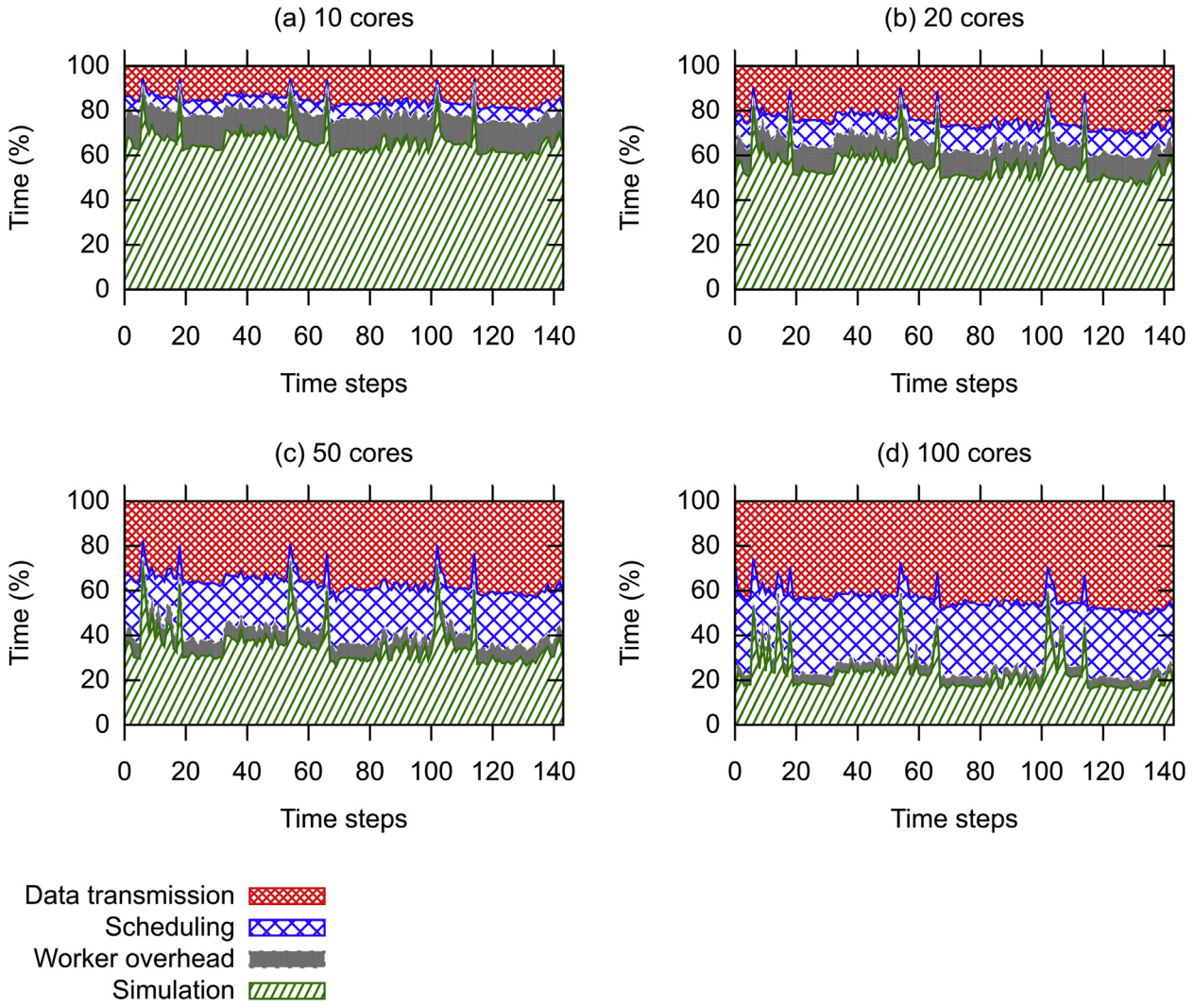

Fig. 15. Processes involved in the forward simulation phase for the cloudified modification of the simulator.

data transfer and execution time. Considering the cloud execution, we created the cloud-oriented service CLAUDE that allows distributing an application over available cloud resources. As a result, we obtained a functional, cloud-based version of EnKF-HGS with an execution performance that is comparable to a cluster-like environment, and a predictable service-related overhead. The usage of CLAUDE gives us the possibility to rely on external services for automatic job distribution, efficient job scheduling and optimal utilisation of computational resources, which in turn facilitates the performance of the application and eases the application development process. Additionally, CLAUDE offers a well-integrated data storage service, which lowers the demands for the target infrastructure. The comparison between the cloud and the cluster infrastructure was made with a relatively simple model setup with considerably smaller problem size and execution time than real-world applications. On the one hand, a larger number of model grid cells, as encountered in many practical applications, would increase the size of the data packages that need to be transferred through the network in the cloud-based execution. This would certainly increase the overhead related to data transfer as compared to a cluster-based solution. On the other hand, however, the simulation times for more complex real-world model setups would be considerably higher than the model that was used here in the comparison study. As a consequence, the relative overhead for data exchange in the filtering part of the cloud execution for more complex models should still be in a comparable range to what has been found in this study.

The application of the proposed cloud-based real-time modelling platform to real-world applications, of course, raises some additional issues. One of them is the question of data security/data ownership in cloud-based services. Field measurements as well as simulation results may have to be kept confidential by water managers. This means that the data policy of the cloud provider needs to be thoroughly checked for ownership rights before a data acquisition or simulation system is deployed in such environments. In addition, water managers need to be sure that they have a continuous access to their data and modelling systems, because they usually need these services on an operational basis. If the connection to these services fails, for example, related to a problem with the cloud provider, this could result in considerable problems for water managers, for example, when such a system is used for short-term predictions like flood forecasting. Therefore, when such a system is deployed in a cloud-computing infrastructure, a reliable access and functioning of the services needs to be guaranteed. This could be achieved, for example, by providing a backup management system on a second cloud. However, as cloud computing is a continuously growing business, and more and more IT companies move towards cloud-based infrastructures, water managers have

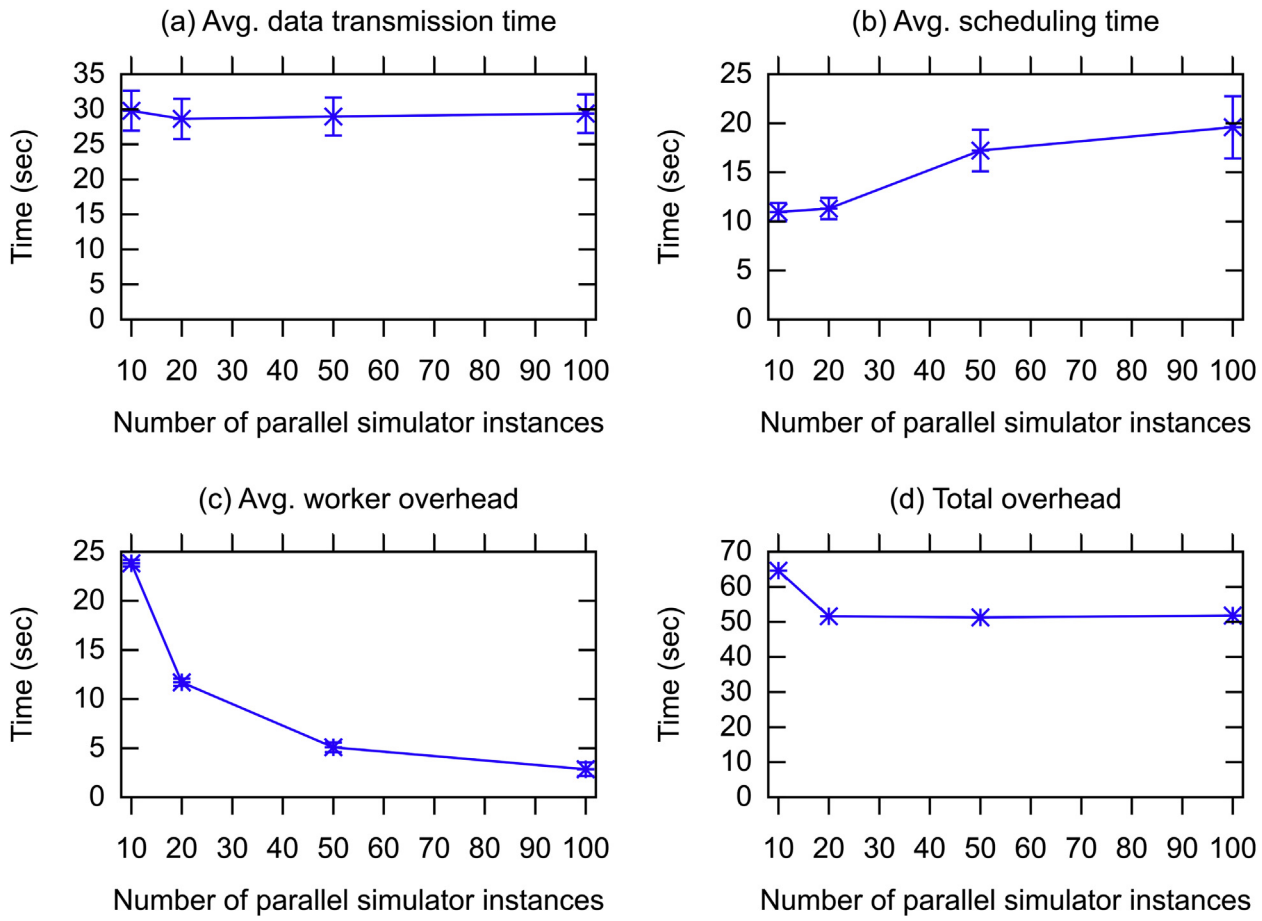

Fig. 16. Absolute time of the overhead processes.

some flexibility with respect to potential cloud providers. Hence, when a cloud-based modelling and monitoring platform needs to be deployed in practice, practitioners can choose between a variety of different cloud service providers in order to meet their specific requirements for data security and operational stability. Another issue in real-world applications is the connection of the modelling system to the online data storage of any kind of wireless sensor networks. In our test case it was assumed that the measurement data are readily available in the cloud infrastructure. However, in real-world applications it is also necessary to account for sensor failure, non-physical values of the measurements, disconnection from the measurement data base and the quality of the measurement data. This would require an additional pre-processing step that includes an exception handling for such events, which needs to be adjusted to the actually used data type and sensor network. The cloud-based modelling and data assimilation system proposed in this paper is, in general, specifically designed for the assimilation of hydraulic heads with the integrated, physically-based hydrological model HydroGeoSphere. Most parts of this system are not generic, meaning that the assimilation of another data type (e.g., concentration data) would require additional programming effort in the data assimilation module. Additionally, if another hydrological model with a different model input structure should be used in conjunction with the cloud-based data assimilation, the interface between this model and the data assimilation system would need to be re-programmed and the CLAUDE driver would have to be adjusted to the new model input structure. Our experience from the setting up of the data assimilation system is that the coupling of a new model with the data assimilation software constitutes a large part of the work. Moving the execution from a cluster-based environment to the cloud computation was, on the other hand, relatively straight-forward given the fact that the CLAUDE system took care of the management and scheduling of the ensemble simulations.

5. Conclusions

In this paper we established a cloud-based real-time modelling and data assimilation platform for the integrated, physically-based hydrological model HydroGeoSphere. As a first step, a data assimilation system was created that is able to perform stochastic predictions with HydroGeoSphere, taking into account uncertainties in model parameters as well as boundary and initial conditions. Additionally, the system allows to assimilate hydraulic head data sequentially with the ensemble Kalman filter. Updated model predictions may then be subsequently used to perform real-time management of water resources. In a second step, the originally cluster-based data assimilation and modelling platform was adapted to a cloud computing environment. The necessary changes in the code structure were minor and concerned the forward propagation of the different model realisations: in the originally cluster-based implementation, the ensemble integration was achieved by spawning a certain number of MPI processes on a computing cluster whereas in the cloud-based implementation the calculation of different model realisations was achieved by distributing the workload among different nodes in the cloud environment with CLAUDE. This workload distribution involved the sending of necessary input data to the nodes, the forward

simulation on the worker nodes and the subsequent retrieval of simulation results from these nodes. A performance comparison of the cluster-based and cloud-based implementation with a relatively simple tilted V-catchment model showed that the overhead in the cloud-based implementation is within an affordable range, showing that such a system is also well-suited for real-world applications.

Generally, we expect that stochastic real-time simulations using integrated, physically-based hydrological models are going to become more common in practical water resources management applications. The reason is that integrated hydrological models provide major advantages over traditional groundwater models with respect to process description and the coupling between groundwater, surface water, and land surface processes, which are, for example, useful for flood and drought predictions and water balance calculations. Furthermore, there is a growing awareness that uncertainties in model predictions need to be taken into account for successful water management. Along with that, data assimilation offers the opportunity to correct such uncertain model predictions with measurement data, which are increasingly collected by online sensors. Data assimilation can be used in conjunction with real-time optimisation methods to improve water resources management. The deployment of such a hydrological data assimilation system in a cloud environment, as shown in this paper, offers several advantages for the application in practical water management problems: (i) Cloud computing resources can be requested on an on-demand basis, which reduces the financial overhead for acquiring and maintaining the computation infrastructure that is required for the computationally demanding ensemble predictions with integrated hydrological models. (ii) Moving to cloud-based computations is relatively straightforward for ensemble-based methods like the ensemble Kalman filter, which only require minimal code changes in the forward propagation of the model realisations. (iii) The computational overhead for cloud-based implementations of ensemble predictions is in an affordable range, as shown for the tilted V-catchment problem in this paper. (iv) Cloud infrastructure is able to provide a seamless integration of observation data acquired in the field through wireless sensor networks and stochastic real-time simulations with integrated hydrological models for the purpose of data assimilation. Additionally, assessment and control of such systems can be achieved through customised web services, which ease the usability for practitioners. Such systems can be used to tackle a wide range of hydrological problems, like the adaptive operation of well fields, flood forecasting, monitoring and management of contaminated sites or optimal irrigation scheduling.

Code availability

The EnKF-HGS data assimilation software and the CLAUDE driver are available on request from the authors (email to h.hendricks-franssen@fz-juelich.de or peter.kropf@unine.ch).

Acknowledgements

W. Kurtz gratefully acknowledges the funding provided by the German Research Foundation (DFG) for project SFB-TR32 “Patterns in soil-vegetation-atmosphere systems: monitoring, modelling and data assimilation”. The University of Bern and University of Neuchâtel acknowledge the “AAA/SWITCH – e-Infrastructure for e-Science” programme supporting the development of the acquisition infrastructure via the “Authentication, Authorization, Accounting, and Auditing in Wireless Mesh Networks (A4-Mesh)” and “Easily Deployable A4 Wireless Mesh Networks (eA4-Mesh)” projects. Moreover, the University of Bern and University of Neuchâtel want to express their greatest gratitude to the “Swiss Academic Compute Cloud” supporting the development of the cloud computing infrastructure for hydrogeological use-cases.

References

Aquanty Inc, 2016. HydroGeoSphere. A Three-dimensional Numerical Model Describing Fully Integrated Subsurface and Surface Flow and Solute Transport. Waterloo, ON, Canada.

Ashby, S.F., Falgout, R.D., 1996. A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations. Nucl. Sci. Eng. 124, 145–159.

Bailey, R.T., Baù, D., 2012. Estimating geostatistical parameters and spatially-variable hydraulic conductivity within a catchment system using an ensemble smoother. Hydrology Earth Syst. Sci. 16, 287–304. http://dx.doi.org/10.5194/hess-16-287-2012.

Barnhart, K., Urteaga, I., Han, Q., Jayasumana, A., Illangasekare, T., 2010. On integrating groundwater transport models with wireless sensor networks. Ground Water 48, 771–780. http://dx.doi.org/10.1111/j.1745-6584.2010.00684.x.

Bauer-Gottwein, P., Schneider, R., Davidsen, C., 2016. Optimizing wellfield operation in a variable power price regime. Groundwater 54, 92–103. http://dx.doi.org/10.1111/gwat.12341.

Bauser, G., Franssen, H.-J.H., Kaiser, H.-P., Kuhlmann, U., Stauffer, F., Kinzelbach, W., 2010. Real-time management of an urban groundwater well field threatened by pollution. Environmental Sci. & Technol. 44, 6802–6807. http://dx.doi.org/10.1021/es100648j.

Bauser, G., Hendricks Franssen, H.-J., Stauffer, F., Kaiser, H.-P., Kuhlmann, U., Kinzelbach, W., 2012. A comparison study of two different control criteria for the real-time management of urban groundwater works. J. Environmental Manag. 105, 21–29. http://dx.doi.org/10.1016/j.jenvman.2011.12.024.

Blessent, D., Therrien, R., Gable, C.W., 2011. Large-scale numerical simulation of groundwater flow and solute transport in discretely-fractured crystalline bedrock. Adv. Water Resour. 34, 1539–1552. http://dx.doi.org/10.1016/j.advwatres.2011.09.008.

Blessent, D., Jørgensen, P.R., Therrien, R., 2014. Comparing discrete fracture and continuum models to predict contaminant transport in fractured porous media. Ground Water 52, 84–95. http://dx.doi.org/10.1111/gwat.12032.

Bogena, H.R., Herbst, M., Huisman, J.A., Rosenbaum, U., Weuthen, A., Vereecken, H., 2010. Potential of wireless sensor networks for measuring soil water content variability. Vadose Zone J. 9, 1002–1013. http://dx.doi.org/10.2136/vzj2009.0173.

Brunner, P., Simmons, C.T., 2012. HydroGeoSphere: a fully integrated, physically based hydrological model. Ground Water 50, 170–176. http://dx.doi.org/10.1111/j.1745-6584.2011.00882.x.

Bürger, C.M., Kollet, S., Schumacher, J., Bösel, D., 2012. Introduction of a web service for cloud computing with the integrated hydrologic simulation platform ParFlow. Comput. & Geosciences 48, 334–336. http://dx.doi.org/10.1016/j.cageo.2012.01.007.

Burgers, G., Jan van Leeuwen, P., Evensen, G., 1998. Analysis scheme in the ensemble kalman filter. Mon. Weather Rev. 126, 1719–1724. http://dx.doi.org/10.1175/1520-0493(1998)126<1719:ASITEK>2.0.CO;2.

Caino-Lores, S., Lapin, A., Kropf, P., Carretero, J., 2016. Lessons learned from applying big data paradigms to a large scale scientific workflow. In: Proceedings of the 11th Workshop on Workflows in Support of Large-scale Science, WORKS '16. ACM. http://dx.doi.org/10.1145/1235.

Camporese, M., Paniconi, C., Putti, M., Salandin, P., 2009. Ensemble Kalman filter data assimilation for a process-based catchment scale model of surface and subsurface flow. Water Resour. Res. 45, W10 421. http://dx.doi.org/10.1029/2008WR007031.

Chen, Y., Zhang, D., 2006. Data assimilation for transient flow in geologic formations via ensemble Kalman filter. Adv. Water Resour. 29, 1107–1122. http://dx.doi.org/10.1016/j.advwatres.2005.09.007.

De Schepper, G., Therrien, R., Refsgaard, J.C., Hansen, A.L., 2015. Simulating coupled surface and subsurface water flow in a tile-drained agricultural catchment. J. Hydrology 521, 374–388. http://dx.doi.org/10.1016/j.jhydrol.2014.12.035.

Ercan, M.B., Goodall, J.L., Castronova, A.M., Humphrey, M., Beekwilder, N., 2014. Calibration of SWAT models using the cloud. Environmental Model. & Softw. 62, 188–196. http://dx.doi.org/10.1016/j.envsoft.2014.09.002 wOS:000346751800016.

Evensen, G., 1994. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. J. Geophys. Res. Oceans 99, 10 143–10 162. http://dx.doi.org/10.1029/94JC00572.

Gaukroger, A.M., Werner, A.D., 2011. On the Panday and Huyakorn surface-subsurface hydrology test case: analysis of internal flow dynamics. Hydrol. Process. 25, 2085–2093. http://dx.doi.org/10.1002/hyp.7959.

Granell, C., Havlik, D., Schade, S., Sabeur, Z., Delaney, C., Pielorz, J., Uslaender, T., Mazzetti, P., Schleidt, K., Kobernus, M., Havlik, F., Bodsberg, N.R., Berre, A., Mon, J.L., 2016. Future Internet technologies for environmental applications. Environmental Model. & Softw. 78, 1–15. http://dx.doi.org/10.1016/j.envsoft.2015.12.015 wOS:000371377800001.

Hansen, A.K., Madsen, H., Bauer-Gottwein, P., Falk, A.K.V., Rosbjerg, D., 2012. Multi-objective optimization of the management of a waterworks using an integrated well field model. Hydrology Res. 43, 430–444. http://dx.doi.org/10.2166/

nh.2012.142.
Hendricks Franssen, H.J., Kinzelbach, W., 2008. Real-time groundwater flow modeling with the Ensemble Kalman Filter: joint estimation of states and parameters and the filter inbreeding problem. Water Resour. Res. 44, W09 408. http://dx.doi.org/10.1029/2007WR006505.
Hendricks Franssen, H.J., Kaiser, H.P., Kuhlmann, U., Bauser, G., Stauffer, F., Müller, R., Kinzelbach, W., 2011. Operational real-time modeling with ensemble Kalman filter of variably saturated subsurface flow including stream-aquifer interaction and parameter updating. Water Resour. Res. 47, W02 532. http://dx.doi.org/10.1029/2010WR009480.
Herbst, M., Gottschalk, S., Reiel, M., Hardelauf, H., Kasteel, R., Javaux, M., Vanderborght, J., Vereecken, H., 2008. On preconditioning for a parallel solution of the Richards equation. Comput. & Geosciences 34, 1958–1963. http://dx.doi.org/10.1016/j.cageo.2008.02.020.
Hunt, R.J., Luchette, J., Schreuder, W.A., Rumbaugh, J.O., Doherty, J., Tonkin, M.J., Rumbaugh, D.B., 2010. Using a cloud to replenish parched groundwater modeling efforts. Ground Water 48, 360–365. http://dx.doi.org/10.1111/j.1745-6584.2010.00699.x.
Jamakovic, A., Dimitrova, D.C., Anwander, M., Macicas, T., Braun, T., Schwanbeck, J., Staub, T., Nyffenegger, B., 2013. Real-world Energy Measurements of a Wireless Mesh Network. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 218–232. http://dx.doi.org/10.1007/978-3-642-40517-418. http://dx.doi.org/10.1007/978-3-642-40517-4_18.
Jones, J.E., Woodward, C.S., 2001. Newton-Krylov-multigrid solvers for large-scale, highly heterogeneous, variably saturated flow problems. Adv. Water Resour. 24, 763–774. http://dx.doi.org/10.1016/S0309-1708(00)00075-0.
Kollet, S.J., Maxwell, R.M., 2006. Integrated surface-groundwater flow modeling: a free-surface overland flow boundary condition in a parallel groundwater flow model. Adv. Water Resour. 29, 945–958. http://dx.doi.org/10.1016/j.advwatres.2005.08.006.
Kropf, P., Schiller, E., Brunner, P., Schilling, O., Hunkeler, D., Lapin, A., 2014. Wireless Mesh networks and cloud computing for real time environmental simulations. In: Recent Advances in Information and Communication Technology. Proceedings of the 10th International Conference on Computing and Information Technology (IC2IT2014), Advances in Intelligent Systems and Computing. Springer. http://dx.doi.org/10.1007/978-3-319-06538-01.
Kurtz, W., Hendricks Franssen, H.-J., Kaiser, H.-P., Vereecken, H., 2014. Joint assimilation of piezometric heads and groundwater temperatures for improved modeling of river-aquifer interactions. Water Resour. Res. 50, 1665–1688. http://dx.doi.org/10.1002/2013WR014823.
Kurtz, W., He, G., Kollet, S.J., Maxwell, R.M., Vereecken, H., Hendricks Franssen, H.-J., 2016. TerrSysMP-PDAF (version 1.0): a modular high-performance data assimilation framework for an integrated land surface-subsurface model. Geosci. Model Dev. 9, 1341–1360. http://dx.doi.org/10.5194/gmd-9-1341-2016.
Lapin, A., Schiller, E., Kropf, P., Schilling, O., Brunner, P., Kapic, A. J., Braun, T., and Maffioletti, S.: Real-Time Environmental Monitoring for Cloud-Based Hydrogeological Modeling with HydroGeoSphere, in: 2014 IEEE Intl Conf on High Performance Computing and Communications, 2014 IEEE 6th Intl Symp on Cyberspace Safety and Security, 2014 IEEE 11th Intl Conf on Embedded Software and Syst (HPCC,CSS,ICESS), pp. 959–965, doi:10.1109/HPCC.2014.154, 2014.
Lapin, A., Schiller, E., Kropf, P., 2015. CLAUDE: cloud-computing for non-interactive long-running computationally intensive scientific applications. In: Unger, H., Halang, W.A. (Eds.), Autonomous Systems 2015: Proceedings of the 8th GI Conference. VDI Verlag, pp. 221–232.
Li, L., Zhou, H., Gómez-Hernández, J.J., Hendricks Franssen, H.-J., 2012. jointly mapping hydraulic conductivity and porosity by assimilating concentration data via ensemble Kalman filter. J. Hydrology 428¢429, 152–169. http://dx.doi.org/10.1016/j.jhydrol.2012.01.037.
Li, X., Cheng, X., Gong, P., Yan, K., 2011. Design and implementation of a wireless sensor network-based remote water-level monitoring system. Sensors 11, 1706–1720. http://dx.doi.org/10.3390/s110201706.
Liu, G., Chen, Y., Zhang, D., 2008. Investigation of flow and transport processes at the MADE site using ensemble Kalman filter. Adv. Water Resour. 31, 975–986. http://dx.doi.org/10.1016/j.advwatres.2008.03.006.
Liu, Y., Weerts, A.H., Clark, M., Hendricks Franssen, H.-J., Kumar, S., Moradkhani, H., Seo, D.-J., Schwanenberg, D., Smith, P., van Dijk, A.I.J.M., van Velzen, N., He, M., Lee, H., Noh, S.J., Rakovec, O., Restrepo, P., 2012. Advancing data assimilation in operational hydrologic forecasting: progresses, challenges, and emerging opportunities. Hydrol. Earth Syst. Sci. 16, 3863–3887. http://dx.doi.org/10.5194/hess-16-3863-2012.
Loden, P., Han, Q., Porta, L., Illangasekare, T., Jayasumana, A.P., 2009. A wireless sensor system for validation of real-time automatic calibration of groundwater transport models. J. Syst. Softw. 82, 1859–1868. http://dx.doi.org/10.1016/j.jss.2009.05.049.
Marti, B.S., 2014. Real-time Management and Control of Groundwater Flow Field and Quality. Ph.D. thesis. ETH-Zürich.
Maxwell, R.M., 2013. A terrain-following grid transform and preconditioner for parallel, large-scale, integrated hydrologic modeling. Adv. Water Resour. 53, 109–117. http://dx.doi.org/10.1016/j.advwatres.2012.10.001.
Mell, P., Grance, T., 2011. The NIST Definition of Cloud Computing. Tech. Rep. NIST Special Publication 800-145. National Institute of Standards and Technology. http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf (last access: 15th October 2016).
Mills, R.T., Lu, C., Lichtner, P.C., Hammond, G.E., 2007. Simulating subsurface flow and transport on ultrascale computers using PFLOTRAN. J. Phys. Conf. Ser. 78, 012 051. http://dx.doi.org/10.1088/1742-6596/78/1/012051.
Mure-Ravaud, M., Binet, G., Bracq, M., Perarnaud, J.J., Fradin, A., Litrico, X., 2016. A web based tool for operational real-time flood forecasting using data assimilation to update hydraulic states. Environmental Model. & Softw. 84, 35–49. http://dx.doi.org/10.1016/j.envsoft.2016.06.002 wOS: 000385595200003.
Nowak, W., 2009. Best unbiased ensemble linearization and the quasi-linear Kalman ensemble generator. Water Resour. Res. 45, W04 431. http://dx.doi.org/10.1029/2008WR007328.
Panday, S., Huyakorn, P.S., 2004. A fully coupled physically-based spatially-distributed model for evaluating surface/subsurface flow. Adv. Water Resour. 27, 361–382. http://dx.doi.org/10.1016/j.advwatres.2004.02.016.
Rasmussen, J., Madsen, H., Jensen, K.H., Refsgaard, J.C., 2015. Data assimilation in integrated hydrological modeling using ensemble Kalman filtering: evaluating the effect of ensemble size and localization on filter performance. Hydrol. Earth Syst. Sci. 19, 2999–3013. http://dx.doi.org/10.5194/hess-19-2999-2015.
Ritsema, C.J., Kuipers, H., Kleiboer, L., van den Elsen, E., Oostindie, K., Wesseling, J.G., Wolthuis, J.-W., Havinga, P., 2009. A new wireless underground network system for continuous monitoring of soil water contents. Water Resour. Res. 45, W00D36. http://dx.doi.org/10.1029/2008WR007071.
Robinson, D.A., Campbell, C.S., Hopmans, J.W., Hornbuckle, B.K., Jones, S.B., Knight, R., Ogden, F., Selker, J., Wendroth, O., 2008. Soil moisture measurement for ecological and hydrological watershed-scale Observatories: a review. Vadose Zone J. 7, 358. http://dx.doi.org/10.2136/vzj2007.0143.
Schilling, O.S., Doherty, J., Kinzelbach, W., Wang, H., Yang, P.N., Brunner, P., 2014. Using tree ring data as a proxy for transpiration to reduce predictive uncertainty of a model simulating groundwater¢ surface water¢vegetation interactions. J. Hydrology 519 (Part B), 2258–2271. http://dx.doi.org/10.1016/j.jhydrol.2014.08.063.
Schwanenberg, D., Breukelen, A. v., and Hummel, S.: Data assimilation for supporting optimum control in large-scale river networks, in: 2011 IEEE International Conference on Networking, Sensing andControl (ICNSC), pp. 98–103, doi: 10.1109/ICNSC.2011.5874881, 2011.
Seo, D.-J., Cajina, L., Corby, R., Howieson, T., 2009. Automatic state updating for operational streamflow forecasting via variational data assimilation. J. Hydrology 367, 255–275. http://dx.doi.org/10.1016/j.jhydrol.2009.01.019.
Shi, Y., Davis, K.J., Zhang, F., Duffy, C.J., Yu, X., 2015. Parameter estimation of a physically-based land surface hydrologic model using an ensemble Kalman filter: a multivariate real-data experiment. Adv. Water Resour. 83, 421–427. http://dx.doi.org/10.1016/j.advwatres.2015.06.009.
Sun, A., 2013. Enabling collaborative decision-making in watershed management using cloud-computing services. Environmental Model. & Softw. 41, 93–97. http://dx.doi.org/10.1016/j.envsoft.2012.11.008 wOS:000315974500009.
Tang, Q., Kurtz, W., Brunner, P., Vereecken, H., Hendricks Franssen, H.J., 2015. Characterisation of river'aquifer exchange fluxes: the role of spatial patterns of riverbed hydraulic conductivities. J. Hydrology 531 (Part 1), 111–123. http://dx.doi.org/10.1016/j.jhydrol.2015.08.019.
Varia, J., 2008. Cloud Architectures-White Paper. https://aws.amazon.com/about-aws/whats-new/2008/07/16/cloud-architectures-white-paper/ (last access: 15th October 2016).
Vereecken, H., Neuendorf, O., Lindenmayr, G., Basermann, A., 1996. A Schwarz domain decomposition method for solution of transient unsaturated water flow on parallel computers. Ecol. Model. 93, 275–289. http://dx.doi.org/10.1016/0304-3800(95)00224-3.
Watras, C.J., Morrow, M., Morrison, K., Scannell, S., Yaziciaglu, S., Read, J.S., Hu, Y.-H., Hanson, P.C., Kratz, T., 2014. Evaluation of wireless sensor networks (WSNs) for remote wetland monitoring: design and initial results. Environmental Monit. Assess. 186, 919–934. http://dx.doi.org/10.1007/s10661-013-3424-8.
Weerts, A.H., El Serafy, G.Y., Hummel, S., Dhondia, J., Gerritsen, H., 2010. Application of generic data assimilation tools (DATools) for flood forecasting purposes. Comput. & Geosciences 36, 453–463. http://dx.doi.org/10.1016/j.cageo.2009.07.009.
Zhang, D., Chen, X., Yao, H., James, A., 2016. Moving SWAT model calibration and uncertainty analysis to an enterprise Hadoop-based cloud. Environmental Model. & Softw. 84, 140–148. http://dx.doi.org/10.1016/j.envsoft.2016.06.024 wOS:000385595200011.