ARTICLE IN PRESS

Future Generation Computer Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Holistic approach to management of IT infrastructure for environmental monitoring and decision support systems with urgent computing capabilities

Bartosz Balis *, Robert Brzoza-Woch, Marian Bubak, Marek Kasztelnik, Bartosz Kwolek, Piotr Nawrocki, Piotr Nowakowski, Tomasz Szydlo, Krzysztof Zielinski

AGH University of Science and Technology, Department of Computer Science, Krakow, Poland

HIGHLIGHTS

- IT infrastructures for environmental monitoring systems are investigated.
- Holistic management of their configuration to maintain QoS is proposed.
- The approach optimizes the system as a whole rather than isolated subsystems.
- Optimization goals are different in urgent and normal modes of operation.
- The approach is validated with a levee monitoring use case.

ARTICLE INFO

Article history:
Received 11 April 2016
Received in revised form
12 August 2016
Accepted 13 August 2016
Available online xxxx

Keywords:
IT infrastructure management
Urgent computing
Environmental computing
Smart levee monitoring
Internet of Things
Cloud computing

ABSTRACT

Modern environmental monitoring and decision support systems are based on complex IT infrastructures comprising multiple hardware and software subsystems that need to provide a variety of Quality of Service (QoS) guarantees required for *urgent computing* services, essential in emergency situations. Such IT infrastructures need to be managed in order to maintain the quality of service, which – especially when operating in the urgent mode – involves optimization of multiple, often conflicting, objectives and making trade-offs between them. Existing approaches do not solve this issue optimally because they focus on delivering quality of service within individual subsystems in isolation. We propose a holistic approach to system management which takes into account knowledge about the system as a whole—in particular the interplay of conflicting objectives and configuration options across all subsystems. We argue that such an approach produces a better configuration of the involved subsystems, improving the resolution of trade-offs between cost, energy and performance objectives, leading to their better overall fulfillment in comparison with the non-holistic approach in which individual subsystems are managed in isolation. We validate our approach using a prototype implementation of the holistic optimization algorithm—the Holistic Computing Controller, and applying it to a smart levee monitoring and flood decision support system.

1. Introduction

Environmental monitoring and decision support systems increasingly rely on modern IT technologies, notably the so-called Internet of Things (IoT) integrated with cloud computing [1]. Together, these technologies enable real-time monitoring of natural phenomena, provide advance warning of approaching disasters and help mitigate their impact through results of data analyses and resource-intensive simulations. However, these results must be delivered in a timely fashion and therefore the IT infrastructure must provide *urgent computing* (UC) [2] capabilities in order to support applications subject to soft deadlines. On the other hand, infrastructure limitations such as the use of resource-constrained devices (e.g. environmental sensors) must be taken into account. Such requirements imply that the infrastructure needs to be managed in order to adjust its configuration to changing conditions and maintain the required quality of service.

* Corresponding author.
E-mail address: balis@agh.edu.pl (B. Balis).

http://dx.doi.org/10.1016/j.future.2016.08.007
0167-739X/

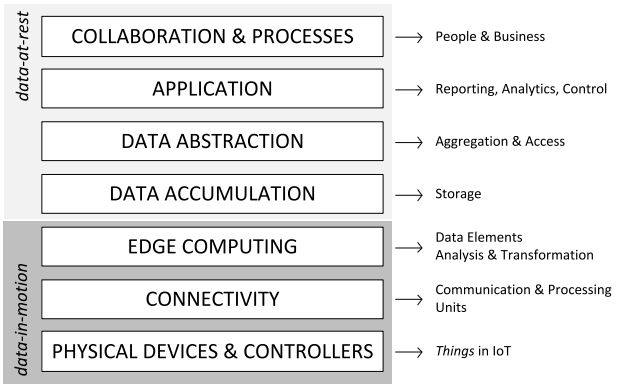

Fig. 1. Internet of things reference layered architecture [3].

The IoT system architecture [3] defines seven layers dealing with data acquisition, transmission, processing and applications (Fig. 1). However, only the four topmost layers have been addressed in existing UC systems. Such systems focus on quality of service through on-demand resource allocation using e-Infrastructures [4], high-throughput data stream processing [5], or provisioning of storage resources for data-intensive urgent applications [6]. There is a notable lack of a holistic approach addressing the management of the entire IoT-Cloud stack, including its three bottom layers, i.e. physical devices (sensors), connectivity and edge computing. Implementing such an approach poses a challenge for two reasons. First, environmental monitoring systems may operate in several modes – typically a 'normal' and an 'urgent' mode – characterized by different resource and QoS requirements. Second, the QoS requirements of different IT subsystems conflict with one another and almost all of them are in contradiction with the need to maintain energy efficiency and reduce operating costs. Consequently, the holistic approach requires knowledge about the system as a whole and a mechanism to calculate and orchestrate execution policies for individual subsystems in order to manage the operation of the entire system and resolve trade-offs between conflicting objectives, preferring some of them at the expense of others depending on the current mode of operation.

Existing approaches do not solve this problem optimally because they focus on delivering quality of service within individual subsystems in isolation [4–6], where each subsystem acts separately in accordance with its own internal policy and QoS requirements defined for that subsystem alone. We propose a holistic approach to system management that (i) addresses all layers of the IoT-Cloud system stack; (ii) optimizes and adapts the configuration of the system as a whole rather than its individual subsystems in isolation. We introduce a new component complementing the IoT stack, called the *Holistic Computing Controller* which performs adaptation of the system configuration in two steps: (1) calculation of Pareto-optimal configurations for the entire IT infrastructure based on cost-of-operation and quality-of-service (QoS) requirements of individual IT subsystems; (2) resolution of trade-offs between conflicting optimization objectives in order to select the single best configuration to be deployed in the system.

This proposed holistic approach is validated in the context of the ISMOP system for smart levee monitoring and flood decision support. The ISMOP project[1] operates a research site featuring an experimental smart levee (Fig. 2), in order to conduct controlled flooding experiments, and support comprehensive research on smart levees, including the design of wireless sensors for levee monitoring, development of efficient data acquisition and transmission tools [7], modeling of levee behavior [8], and development of a data management and processing system leveraging cloud infrastructures [9].

The obtained results confirm that the holistic approach leads to improved configuration settings and, consequently, better fulfillment of the system's cost and QoS requirements than would have otherwise been possible had the configuration of all subsystems been managed in isolation.

The paper is organized as follows. Section 2 presents related work. Section 3 explains the holistic approach to system management. Section 4 outlines the architecture and quality-related objectives of a smart levee monitoring and flood decision support system, while Section 5 presents its practical implementation leveraging the holistic approach—the ISMOP system. Section 6 describes a case study and discusses the results of experiments. Section 7 concludes the paper.

2. Related work

Monitoring and decision support systems dealing with natural disasters typically require urgent computing services in order to ensure sufficient supply of computer resources and the required quality of service during a crisis event. Urgent computing systems are designed for applications characterized by the presence of a firm deadline, unpredictability of the urgent event's occurrence, and the possibility to mitigate the event's impact through resource-intensive computations [2]. Illustrative examples of such applications include severe weather forecasting workflows [10], simulation applications [11], storm surge modeling applications [12], wildfire forecasting workflows [13], etc. The common property of such systems is not their resource-critical nature, but rather the time-critical nature of computations (or the need to obtain results within a specified time frame), such as in simulation-based urgent clinical decision-making [14].

When an emergency unfolds, decision support systems require intensive environmental measurements typically provided by sensor networks (wireless sensors in high-tech solutions). Such systems follow the IoT system architectural paradigm [3], dealing with a whole range of processes, from low-level data acquisition by physical devices up to high-level presentation and collaboration features.

Two general operational stages can be distinguished: data provisioning and data processing. The former reflects layers 1–3 of the reference model ('data in motion' issues), while the latter deals with layers 4–7 ('data at rest' issues). Each of these stages is composed of a sequence of detailed tasks, such as data acquisition, preparation for transport (i.e. compression, encryption, aggregation), transfer, buffering, etc. All component tasks should be tuned in terms of QoS requirements. It should be noted that existing approaches focus on optimizing the performance of selected subsystems rather than the system as a whole, e.g. resource allocation [4], data stream processing [5], or provisioning of storage resources [6]. In particular, the correlation with data delivery has not been fully explored in processing subsystems.

In the context of data provisioning, the standards of IoT device management developed by the Open Geospatial Consortium (OGC) has begun to play an important role. This consortium is involved in the development and implementation of open standards to enable the processing and provisioning of various data and services. These standards define, among others, universal interfaces for IoT sensors that allow for effective collaboration between devices from different companies, which can significantly reduce the cost of system implementation and operation. One of the standards developed by OGC is Sensor Web Enablement (SWE) [15], which allows discovery and access to a variety of sensors, transducers, and data repositories through the Internet. This standard provides,

[1] http://www.ismop.edu.pl.

Please cite this article in press as: B. Balis, et al., Holistic approach to management of IT infrastructure for environmental monitoring and decision support systems with urgent computing capabilities, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.08.007

Fig. 2. ISMOP experimental levee (Oct 2015).

inter alia, open interfaces for web applications that use sensors and the possibility of location sensors using geospatial standards. The OGC standards are still under development, but will evidently play an important role in ensuring effective management of IoT devices.

Complex solutions for job prioritization and pre-emption have been developed for HPC or Grid computing infrastructure, e.g. the mechanism of right-of-way tokens in SPRUCE [16] or modification of scheduling engines in order to ensure the requisite turnaround times for urgent jobs [17]. A different approach to resource management, based on on-demand provisioning, renders clouds particularly useful in urgent computing. In the Common Information Space (CIS) [18] of the UrbanFlood early warning system [19] cloud services were used for on-demand deployment and autoscaling of warning system instances in emergency situations [20]. Based on the CLAVIRE platform [21], a model-based approach to management of heterogeneous computing resources for urgent computing scenarios was proposed [22]. Nevertheless, the cited works focus on optimizing the data processing stage (layers 4–7 of IoT model) without correlating it with data delivery issues (layers 1–3).

Urgent and massive data acquisition and transfers are usually taken for granted. In particular, most solutions take advantage of various internal storage solutions, for which the persistence of data resources and effective delivery channels are presumed to be guaranteed and reliable (e.g. Resource/Resource Access Layer in [12], Data Cloud in [10]). The use of proper network protocols or distributed system techniques (e.g. Web services [23] and SOA [24]) solves the problem of reliable transmission, but efficient support of ongoing acquisition of external data (e.g. from numerous environmental sensors), particularly the timeliness of its delivery (i.e. maintenance and adaptation of communication channels) is yet to be addressed.

The need for wider research on urgent data delivery has recently received some attention, as evidenced e.g. by the management of data flows in the LEAD environment [10], the Urgent Data Management Framework (UDMF) [6] or robust data placement for SPRUCE [25]. However, these solutions take advantage of pre-staged data, whereas real-time monitoring (e.g. wildfire control [13], dike condition governance [26], etc.) requires continuous delivery of up-to-date data. The cited environmental monitoring systems are based on sensor networks, specifically wireless ones [27], which have become a common solution in recent years. Ongoing environmental monitoring via a large number of nodes creates a collaborative output environment and dealing with such data streams demands appropriate solutions [28].

Advanced research on the efficiency and reliability of urgent data delivery is taking place. For instance, UDMF [6] introduces new capabilities for urgent computing infrastructures, including QoS and data policy management and monitoring. Among others, urgent storage and data management tools provide configuration of QoS and Service Level Agreements (SLAs) [29] for data services. The authors of [5] recognize the problem of maintaining transport SLAs for sensor data volume in order to avoid disturbing ongoing computations. Their work proposes a net-based architectural model for supporting QoS for multiple concurrent data streams.

However, the above-mentioned solutions focus on data delivery issues and do not take into account deeper correlations with the mechanisms of ongoing computation performed in the central system. They are based on unalterable configurations which aim at enforcing data delivery QoS, which is presumed to be optimal, but not collated with the operating environment issues (e.g. energy balance, current demand of processing subsystem, etc.).In contrast to the above, a holistic approach has been recognized in many domains of complex system study, including natural history, biology, education, etc. [30]. Unlike reductionism, the idea of holism incorporates the concept that a whole entity is more than the mere sum of its parts and it implies attention to a higher level. Systems and their properties should be viewed as synergic wholes, not as collections of parts [31]. In all aforementioned solutions, aspects of governance of data processing and external data delivery are mostly treated as two separate problem domains, however no mechanisms have been found that can leverage the overall synergy of individual system processes.

In the technology domain, the simplest interpretation of holism would be shifting control mechanisms from the local plane to the global one. The concepts of *Smart cities* [32] or *Software-defined Networks* (SDN) [33] provide examples of such approach. In both cases, the business intelligence is shifted from the infrastructure layer (e.g. a local traffic lights system or a network switching appliance) to the control layer (a management center or business applications supported by network controllers respectively). Thus, these systems take advantage of global knowledge and, instead of improving efficiency only locally, provide global optimization.

A corresponding approach is the cornerstone of the presented research. We have not found a model urgent computing system

Please cite this article in press as: B. Balis, et al., Holistic approach to management of IT infrastructure for environmental monitoring and decision support systems with urgent computing capabilities, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.08.007

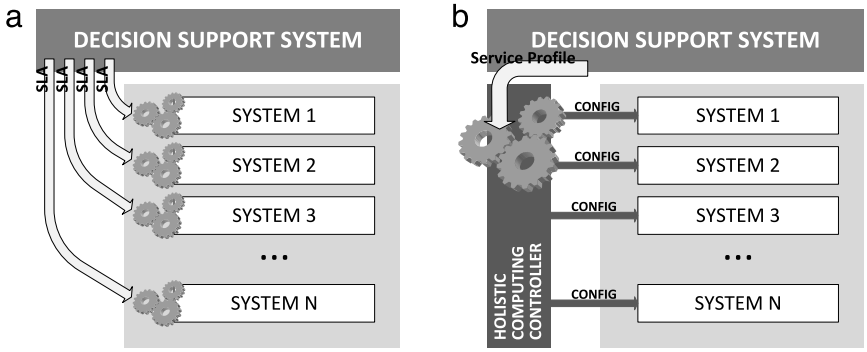

Fig. 3. The difference between (a) isolated and (b) holistic approaches to management of a complex system. In the isolated approach each subsystem is managed separately; in the holistic approach, a holistic computing controller manages the configuration of each subsystem taking advantage of global knowledge and relations between subsystems.

(based on reliable continuous delivery of data from remote environmental sensor networks) that implements mechanisms which guarantee quality of service not only within individual subsystems, but globally throughout the whole system, by taking into account the specific features of particular subsystems and means for their adjustment. Moreover, the aforementioned urgent system solutions mostly disregard the data acquisition and delivery stage, addressing only the top four layers of the IoT model. On the other hand, data delivery subsystems are unaware of the current needs of processing subsystems and operate on the basis of fixed QoS requirements, irrespective of changing requirements and restrictions. The paper introduces a method for such holistic system control, acknowledging all IoT layers, that allows for adaptation of subsystem configurations based on a global – rather than local – point of view.

3. Holistic approach to system management

The operation of an environmental monitoring and decision support system is driven by Service-Level Agreements (SLAs) specific for each subsystem and dependent on the operating mode (urgent or normal[2]). Each SLA contains a set of Quality of Service (QoS) requirements that need to be fulfilled. Such a system needs to be managed so as to adjust its behavior to changing conditions. In this case, management means *switching between configurations* in order to optimize the cost of operation, energy consumption, or quality of service requirements in emergency situations. Thus, it is understood in a limited scope and does not cover all the issues addressed by different IT service management frameworks, such as service lifecycle, software engineering, information security, etc. The management tasks realized within the proposed holistic approach comprise provisioning of optimization goals, selection of currently preferred configurations and their enactment within adequate subsystems.[3]

This section describes the difference between isolated and holistic approaches to system management, and introduces the concept of Holistic Computing Controller.

3.1. Isolated vs. holistic approach

Management of a complex system composed of multiple layers (subsystems) can be achieved on the basis of two complementary approaches, illustrated in Fig. 3:

- *Isolated approach*: each subsystem acts separately according to its own internal policy and Service-Level Agreement (SLA) requirements pertaining to this subsystem alone (Fig. 3(a)).
- *Holistic approach*: the execution policy of each subsystem is orchestrated by an external controller which coordinates the operation of the entire system, based on a *Service Profile* defining SLA requirements for the system as a whole (Fig. 3(b)).

In the first approach, management actions are performed by each subsystem separately. In the second approach, an additional component called the *Holistic Computing Controller* (HCC) is responsible for managing the configuration for each subsystem. HCC can take advantage of global knowledge concerning the system, in particular the mutual relations of its component subsystems. Details of HCC design are described in the following section.

3.2. Holistic computing controller

The architecture of the system which includes the Holistic Computing Controller is shown in Fig. 4. The operation of the system is defined by two operating loops:

- *Decision loop* implemented by the decision support system. In this loop, human decision-makers observe the status of environmental phenomena and objects in the monitored areas of interest and, depending on the current situation, set the operating mode for these areas to urgent or normal. Each operating mode has an associated *service profile* which specifies the SLA requirements for the system in that mode. The profiles are defined by *domain experts*.
- *Reconfiguration loop* implemented by the Holistic Computing Controller. The HCC makes decisions to reconfigure the IT infrastructure in such a way as to *fulfill functional requirements* and *optimize non-functional properties* of the system given the *current context*. The execution of this loop is triggered by a change of the system profile or its external context. Information characterizing the IT infrastructure, required for the HCC to perform the reconfiguration, is provided by *technical experts*.

In the paper, we focus on the reconfiguration loop implemented by the Holistic Computing Controller. The HCC operates in the following cycle:

1. *Monitoring*: observation of the system state and external context. Currently we assume that HCC makes decisions on the basis of the following runtime information: (1) operating mode (normal or urgent); (2) external conditions (such as weather). In a production system detailed monitoring information, such as performance metrics, would also be needed in order to monitor SLAs, detect their violation, and initiate re-configuration to enforce them. However, SLA monitoring and enforcement, while definitely a topic for future research, is currently out of scope of this paper.

[2] Here and throughout the paper, without the loss of generality we consider only these two modes. In real decision support systems there can be multiple operating modes corresponding to environmental threat levels.

[3] When compared to the ITIL framework, this remains in the scope of the Configuration Management System part of ITIL Service Transition [34].

Please cite this article in press as: B. Balis, et al., Holistic approach to management of IT infrastructure for environmental monitoring and decision support systems with urgent computing capabilities, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.08.007

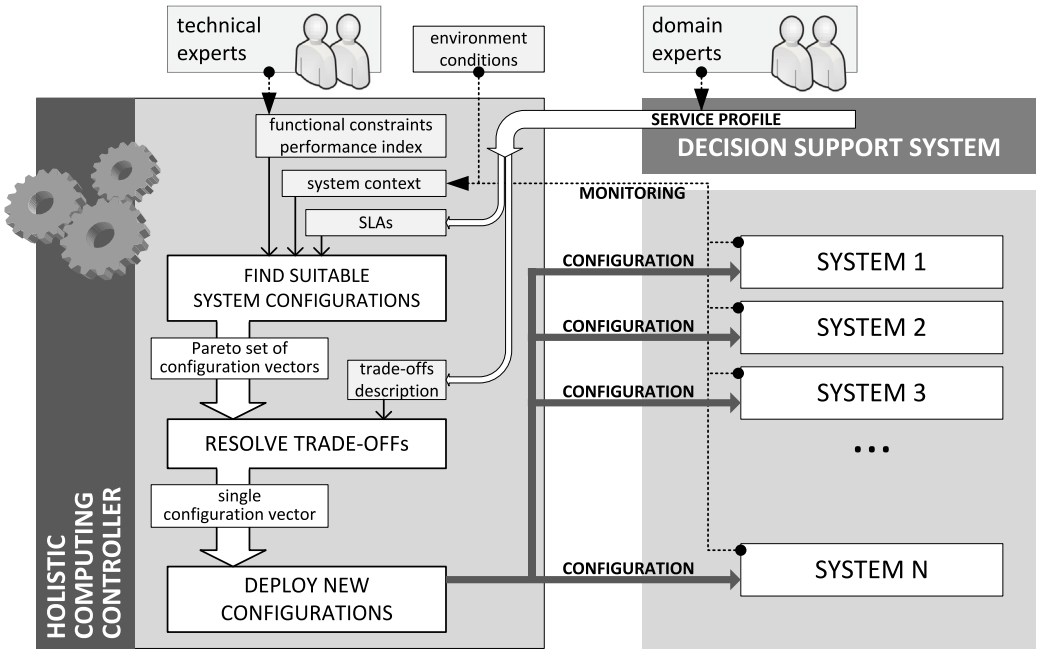

Fig. 4. Operation of a complex decision support system driven by the holistic computing controller. In response to change of the system profile or its external context, HCC calculates and deploys a new optimal configuration for all subsystems.

Table 1
Examples of profile SLAs and the corresponding tradeoffs.

Profile	SLAs	Trade-offs
Normal	1. DMI $\leq$ 1 h 2. DPI $\leq$ 12 h 3. DTI $\leq$ DPI	1. Low OPC is *moderately preferred* over low DPI 2. Low OPC is *strongly preferred* over low EC 3. Low EC is *strongly preferred* over low DTI
Urgent	1. DMI $\leq$ 15 min 2. DPI $\leq$ 30 min 3. DTI $\leq$ DPI 4. SLT $\geq$ 10 days	1. Low DPI is *extremely preferred* over low OPC 2. Low DTI is *moderately preferred* over high SLT 3. Low DMI is *strongly preferred* over low DTI

2. *Optimization*: calculation of configurations that optimize the system's SLA objectives with respect to cost, energy efficiency and Quality of Service guarantees. Since these objectives are conflicting, a multi-objective optimization process is performed, resulting in a set of Pareto-optimal configurations.
3. *Trade-off resolution*: the Pareto-optimal configurations are ranked and the one regarded as best is selected for enactment.
4. *Reconfiguration*: the selected configuration is deployed in the system.

We assume the system has k configurable properties $P = (p_1, p_2, \ldots, p_k)$. A set of configuration options for these properties is also known: $O = (O_1, O_2, \ldots, O_k)$, where $O_i = (o^i_1, o^i_2, \ldots, o^i_{n(k)})$ $(i = 1 \ldots k)$ denotes possible configuration options for property p_i. We define *system configuration* s as a vector of configuration options chosen for each of the configurable properties:

$$s = (o_1, o_2, \ldots, o_k) \in S \quad \text{where } o_i \in O_i.$$

Here S denotes the set of all possible configurations of the system.

The remaining information that must be known to HCC is as follows:

- The current system context $c = (c_1, c_2, \ldots, c_l)$, where c_i denotes the current value of a particular context property (such as battery levels, weather conditions, etc.).
- $q_1(s, c), q_2(s, c), \ldots, q_m(s, c)$—non-functional properties of the system that need to be optimized (examples described in Section 4.2), expressed as functions of the system configuration and context.
- $e = \{e_1, e_2, \ldots, e_p\}$—functional constraints which must be satisfied in order for the system to fulfill its function, for example 'configuration options o^3_2 and o^5_4 are mutually exclusive'.
- $sla = \{sla_1, sla_2, \ldots, sla_r\}$—non-functional constraints imposed on the system's QoS properties. These are basically SLAs (Service-Level Agreements) that need to be fulfilled by the system, such as minimum system lifetime, or maximum data transmission interval. It is a set of inequalities restricting the non-functional objectives.
- $t = \{t_1, t_2, \ldots, t_s\}$—trade-offs which specify the relative importance of the non-functional objectives, given as a pairwise comparison matrix (e.g. 'q_1 *is strongly preferred over* q_2', see examples in Table 1).

Based on the above information, HCC calculates best configuration options for all subsystems. The algorithm executed by HCC can be described in the following steps:

1. Solve a multi-objective optimization problem in order to find the Pareto set of configuration vectors that optimize non-functional objectives and fulfill the functional constraints.
2. Apply trade-off resolution in order to select a single configuration vector out of the Pareto set.

3.2.1. Finding Pareto-optimal configurations

The HCC finds the set of Pareto-optimal configurations by solving the following multi-objective optimization problem:

$$\begin{aligned} \text{minimize} \quad & q = (q_1(s, c), q_2(s, c), \ldots, q_n(s, c)) \in Q \\ \text{subject to constraints} \quad & e \quad \text{and} \quad sla \\ \text{where} \quad & s \in S \\ & c = (c_1, c_2, \ldots, c_k) \\ & e = (e_1, e_2, \ldots, e_l) \\ & sla = (sla_1, sla_2, \ldots, sla_p) \end{aligned}$$

where s is a vector of *decision variables*, i.e. configurations for all configurable properties of the system; c is a vector of input (non-decision) variables; e and sla are vectors of functional and non-functional constraints, respectively; S is the *decision space*,

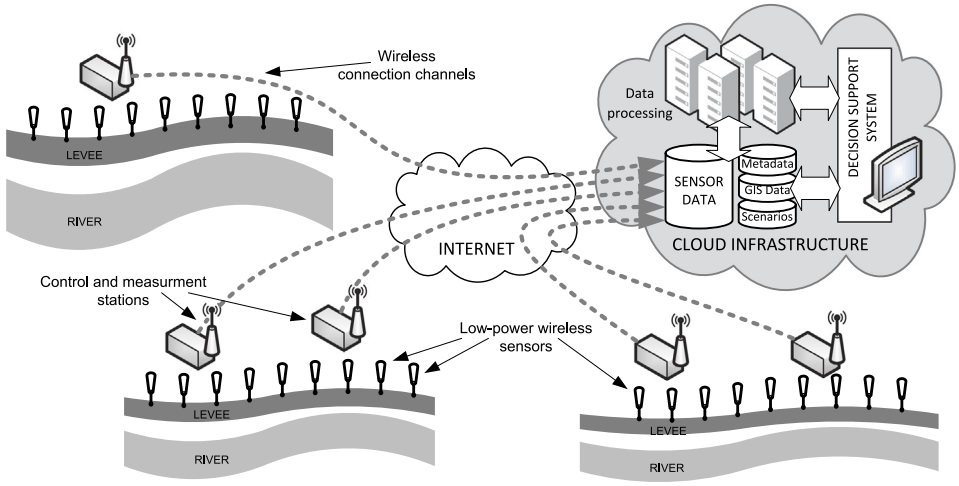

Fig. 5. Smart levee monitoring and decision support system: large-scale distributed deployment.

i.e. a set of all possible configurations of the system; while Q is the *objective space*. Let us note that there are two sets of constraints: the functional constraints that impose restrictions on decision variables, and non-functional constraints that restrict the objective space. We do not define these constraints formally due to their diversity (e.g. inequality relations or logical statements) that can be interpreted by HCC in order to reduce the decision and/or objective space. The output of the optimization is a set of feasible configuration vectors $S^{OPT} = (s_1, s_2, \ldots) \subset S$.

3.2.2. Trade-off resolution

As a result of the previous step we have set of Pareto-optimal solutions that, when applied to the system, will fulfill its functional requirements. Choosing any of these solutions will satisfy only some of the non-functional QoS requirements, leaving others unsatisfied. The decision should be driven by a trade-off that captures the relative importance of QoS objectives. HCC has to solve the following problem of reducing the multi-objective optimization to a single objective:

$$\begin{aligned} \text{minimize} \quad & q_1^N(s,c)k_1 + q_2^N(s,c)k_2 + \cdots + q_n^N(s,c)k_n \\ \text{where} \quad & s \in S^{OPT} \\ & c = (c_1, c_2, \ldots, c_k) \\ & k_1, k_2, \ldots, k_n \in [0;1] \\ & q_l^N(s,c) \text{ is a normalized } q_l(s,c). \end{aligned}$$

Providing explicit weights for each objective function is very difficult, so we propose to describe the trade-offs as a pairwise comparison matrix using the following scale: *extremely preferred*, *very strongly preferred*, *strongly preferred*, *moderately preferred*, *equally preferred*. The final weights are then obtained using the Analytic Hierarchy Process method [35].

The output of this step is a set S^{TR} that contains selected configurations for which the aforementioned function yields the smallest value. Several system configurations may correspond to an identical minimal value—in such cases additional logic should be provided to select the most appropriate solution. Typically, a random solution will be selected, but more advanced decision logic is also feasible—e.g. choosing the solution which introduces fewer changes in the configuration of the system.

4. Smart levee monitoring and decision support system

A typical disaster scenario which lies at the root of the presented research involves a flood wave passing down a river. The flood wave may last from a few hours up to several weeks, and affect a large area comprising hundreds of kilometers of levees. A flood will typically occur due to the failure of a levee resulting from its long-term infiltration.

The business requirements stemming from this emergency scenario dictate that the decision support system should provide regular flood threat assessments for the affected levees in a reliable and timely fashion. In terms of system requirements, it implies that a resilient infrastructure is needed, providing real-time acquisition and transmission of large amounts of sensor measurements, storage and retrieval of the collected data, as well as urgent (on-demand and deadline-driven) computing services. The architecture of such a system is presented in Fig. 5.

During operation the system needs to make trade-offs between conflicting optimization objectives: quality of service (which includes various aspects of data transmission and processing performance), cost of operation and energy consumption (expected system lifetime). How conflicts between these objectives are resolved is dictated by the current mode of operation. In the normal mode the optimization of the system's operating costs is given priority, while during an emergency situation non-functional properties related to reliability and performance are crucial.

The following sections describe the details of the smart levee monitoring and decision support system, including its architecture and subsystems (Section 4.1), non-functional properties which are the optimization objectives (Section 4.2), and service profiles for the normal and urgent operating modes (Section 4.3).

4.1. System architecture

The IT infrastructure of the smart levee and decision support system, shown in Fig. 6, consists of two main systems: the Computing and Data Management System (CDMS) and the Data Acquisition and Pre-processing System (DAPS).

The CDMS comprises the following subsystems:

Computing infrastructure contains physical computers, along with management software, such as a cloud middleware. The computing infrastructure provides the capability to dynamically allocate computing resources required for data processing, e.g. in the form of Virtual Machine (VM) instances.

Data management subsystem is responsible for reliable data storage and access to data required by the decision support system, in particular sensor data received from the Communication layer. The primary responsibility of the data management subsystem is to ensure data availability and data access quality (e.g. throughput and latency).

Please cite this article in press as: B. Balis, et al., Holistic approach to management of IT infrastructure for environmental monitoring and decision support systems with urgent computing capabilities, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.08.007

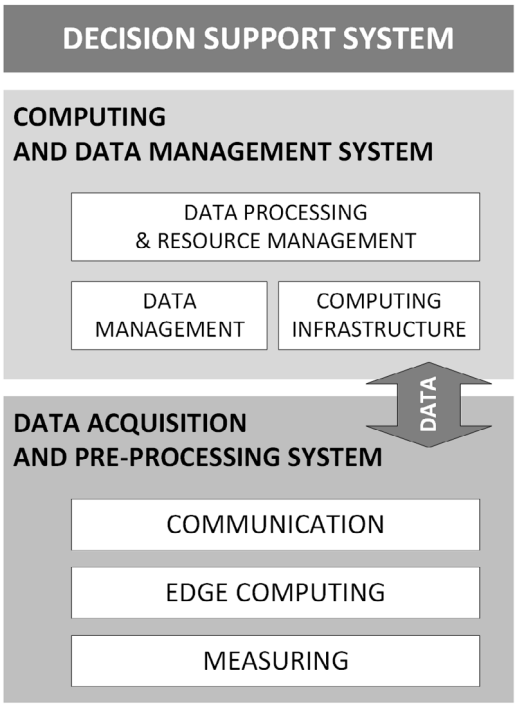

Fig. 6. Layered architecture of a smart levee monitoring and decision support system.

Data processing and resource management subsystem is where data analyses are performed in order to assess the current levee state and forecast its future behavior. These analyses may include anomaly detection, simulation of future levee state, and computation of flood threat levels for individual levee sections.

DAPS is an autonomous entity. Energy for its operation is supplied from renewable energy sources, mainly sunlight, and stored locally in battery banks. Alternatively, the system can be powered by a wind turbine or other energy scavenging methods. DAPS has a multilayer structure, composed of three layers which perform the following operations:

Measuring Layer —comprising sensors which measure physical parameters of the environment. Values of these parameters are sent over a wired or wireless network to edge computing devices. These sensors are mostly wireless and battery operated.

Edge Computing Layer —a collection of many distributed computing resource-constrained devices which control Measuring Layer operation and perform data preprocessing (including compression, encryption and filtering). In more advanced systems this is the place where event processing could be effectively deployed. The operation of this layer is referred to as Fog Computing [36].

Communication Layer —providing bidirectional communication between Edge Computing devices and CDMS. This layer performs routing operations and selects the most suitable communication technologies and routes to transfer preprocessed data to the central system.

4.2. Non-functional properties (optimization objectives)

The system is characterized by a number of non-functional properties related to quality of service, cost of operation, and energy consumption. These properties are the objectives of optimization (minimization or maximization) during operation. The most important non-functional properties are as follows:

1. **Operating cost** (OPC): expenses (in currency units) required to maintain the operation of the system. The total system OPC is the sum of expenses for all individual subsystems, for example the cost of data transfer over the cellular network, the cost of renting computing resources from infrastructure providers, etc.
2. **Data measurement interval** (DMI): an interval (in seconds) which specifies how often sensor parameters are captured by the measuring subsystem. The lower the value of DMI, the more frequently measurements are captured and, consequently, the more accurate data analysis can become. However, low DMI also contributes to increased energy consumption.
3. **Data processing interval** (DPI): an interval (in seconds) specifying how often data analyses, such as the assessment of the current and future state of a levee, are conducted in the data processing subsystem.
4. **Energy Efficiency** (EE): an indicator (a value between 0 and 1) showing how energy efficient the system is. This parameter is especially important in the context of limited energy availability in the lower system layers, as it directly influences system lifetime. EE is only applied to lower layers because energy consumption usually cannot be controlled directly (through configuration) in the higher layers (in particular the computing infrastructures).
5. **Data processing time** (DPT): the time (in seconds) required to complete data processing for a given area of interest (e.g. all levees in a region affected by a flooding threat). Low DPT means that computation outcomes are available faster, which may be crucial for emergency decision-making. However, this also increases the associated operating cost.

   Other quality-related parameters that might be considered include:
6. **Data transmission interval** (DTI): an interval (in seconds) which determines how often sensor measurement data is transmitted by the communication subsystem to the data management subsystem, making it available for further processing. In the ideal case DTI is set to DMI, which means that data is transferred as soon as it is measured. $DTI > DMI$ means that data is temporarily buffered in the communication layer and transmitted in larger packets which saves energy but increases transmission delays.
7. **Data access time** (DAT): delay (in seconds) between data access request and its completion. For large areas of interest it can considerably influence the data processing time, as well as user experience.
8. **System lifetime** (SLT): the period (in seconds) during which the system is expected to maintain correct operation, given the current quality requirements and circumstances (such as weather conditions). System lifetime is affected mainly by the measuring layer where sensors have limited available energy. Because of the way EE is defined, the SLT function is directly proportional to EE, i.e. the higher the energy efficiency, the longer the system lifetime.
9. **System availability** (SAV): the time during which the system is available in, expressed as a fraction of total time, e.g. '99% monthly availability'.

4.3. Service profiles

The system has two service profiles defining its requirements and priorities with regard to non-functional objectives in the normal and urgent operating modes. The service profiles are defined in terms of two sets of statements, shown in Table 1:

- *Service Level Agreements* (SLAs) specifying the upper and lower boundaries for the non-functional properties.
- *Trade-off specifications* stating the relative importance of the optimization objectives.

Please cite this article in press as: B. Balis, et al., Holistic approach to management of IT infrastructure for environmental monitoring and decision support systems with urgent computing capabilities, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.08.007

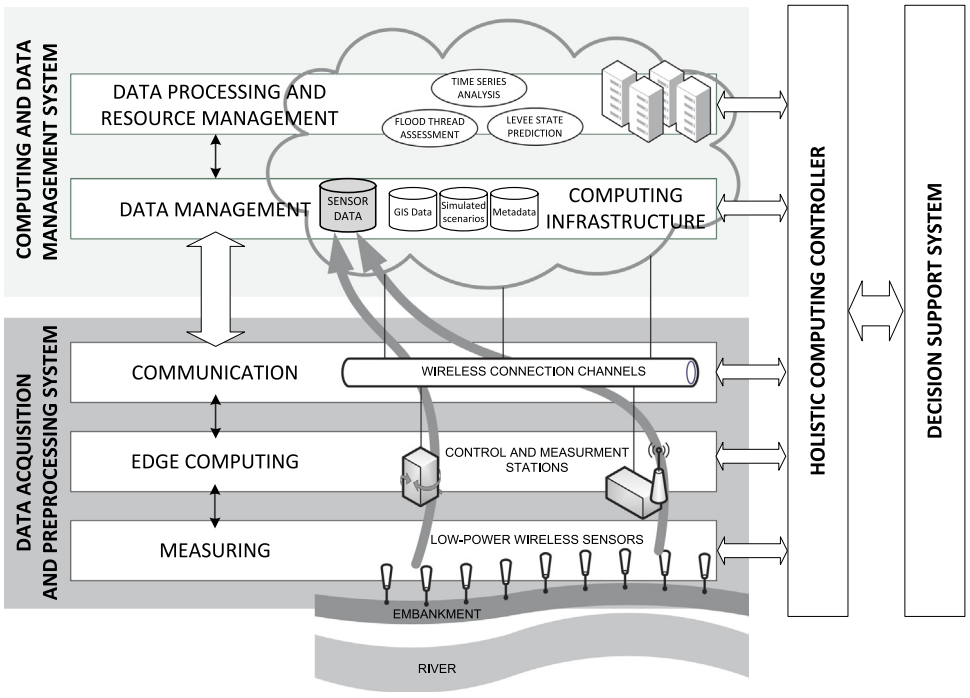

Fig. 7. The subsystems and information flow of the ISMOP flood decision support system.

As expected, in the *Normal profile*, the system is set to be cost and energy efficient while computations can be performed at a convenient time. The *Urgent profile*, on the other hand, calls for increased frequency of measurements and sets firm deadlines for computations.

5. ISMOP flood decision support system

We have developed ISMOP, a practical implementation of the smart levee monitoring and decision support system described in Section 4. To present its operation, the abstract architecture (Fig. 6) introduced in Section 4, is mapped to hardware and software subsystems of the ISMOP system, as presented in Fig. 7. The following sections describe implementation details of two main subsystems comprising ISMOP.

5.1. Data acquisition and preprocessing system

Fig. 8 presents the architecture of the Data Acquisition and Preprocessing System (DAPS) which acquires and transports data obtained from sensors to the Computing and Data Management System.

We have developed a functional prototype of the control and measurement station. The prototype consists of a specialized hardware platform and an embedded software solution. The control-measurement station hardware platform can be considered as *resource-constrained* in terms of energy harvesting and storage capabilities, computing power, on-board resources (e.g. operating memory), and communication channel availability [37]. The station's hardware platform is based on modern low-power ARM Cortex-M4 microcontroller unit (MCU). The station connects to the *Sensor concentrator* and acquires data from the sensors. The acquired data is preprocessed, serialized, and transmitted to higher layers of the system.

The control-measurement station uses multi-level power management features. Each time the MCU is idle, the onboard real-time operating system enters a reduced power consumption mode. This mode allows the system to achieve only limited power savings, but enables rapid resumption of the normal operating mode. The MCU can also enter a deep power saving mode, in which its microprocessor core is disabled. This mode is utilized periodically when the system has no operation to perform. The third power saving mechanism concerns the power supply of the peripheral modules, including sensing and communication subsystems.

The data processing capabilities of the control-measurement station can be implemented in two ways. The first method would be to embed C code in the control-measurement station which is an effective option but requires much work and low-level programming skills. We have also implemented a standard LUA interpreter [38] which runs in a separate task. This allows a software developer to run portable LUA code in the control-measurement station. The LUA interpreter provides a unified execution environment for different hardware and software platforms. Thanks to the utilization of LUA, the processing tasks can be migrated between different devices in the control-measurement network.

The station can transmit data using either of its two available interfaces. By default it uses its built-in GPRS connectivity to transmit data directly to the CDMS. The alternative path is based on XBee (a modified version of 802.15.4) communication [39]. XBee allows us to achieve very low-power and long-range mesh networking capabilities. In this scenario one station disseminates data to other stations until data reaches a station with sufficient cellular network coverage. Subsequently, data can be sent to CDMS via GPRS. Additionally, we have implemented another version of the control-measurement station core module which utilizes similar embedded hardware, but supports wired Ethernet connectivity. This setup, however, remains experimental and is not intended to be used in the planned final implementation (although it is useful in a testbed environment).

All of the developed prototype solutions support the MQTT protocol [40]. Data in the payload of the MQTT *PUBLISH* message [41] is packed simply as a string of ASCII characters. This allows us to use simple decoding procedures on the CDMS side and channel debugging. In the final version we intend to use data aggregation techniques [42] and end-to-end encryption with symmetrical block cipher, such as Advanced Encryption Standard (AES).

Please cite this article in press as: B. Balis, et al., Holistic approach to management of IT infrastructure for environmental monitoring and decision support systems with urgent computing capabilities, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.08.007

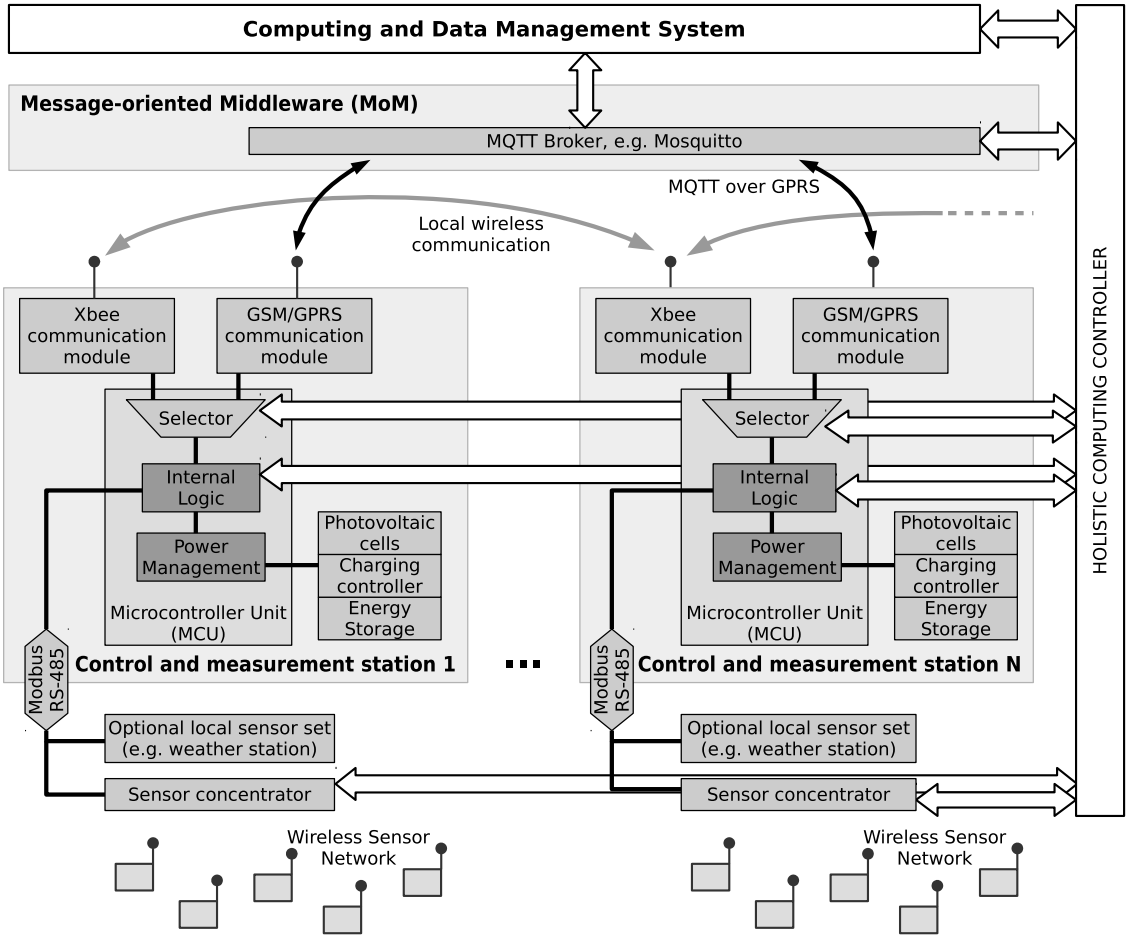

Fig. 8. Architecture of the data acquisition and preprocessing system (including block diagram of the control and measurement station hardware).

HCC can control selected aspects of the control-measurement station. First, HCC can suggest the preferred method of communication via its interface to the communication *Selector*. Access to the internal logic of the station allows HCC to control such aspects of the station's operation as its power management scheme and transmission security features. HCC can also control selected aspects of the sensor concentrator, mainly the measurement time resolution and accuracy.

Future improvements of the hardware platform would include providing better platform flexibility and computing power. In order to achieve better control over hardware and software configuration, the utilization of Field Programmable Gate Array (FPGA) technology can be considered. The increase in computing power in that case would be then achieved at the cost of worse energy efficiency compared to general-purpose MCU. The idea of employing FPGA to environment monitoring has already been described e.g. in [43,44]. The flexibility of configuration with HCC could be fully exposed if we consider not only software but also remote hardware reconfiguration feature. The use of FPGA can also positively impact EE and SLT values. First, the massively parallel implementation of common data processing algorithms leads to their faster execution or to reduction of hardware clock frequency. Either way, the system can potentially utilize less power in a shorter period of time. That approach, however, would be much more demanding for system programmers as it requires hardware-related skills (refer to [45] for more details).

5.2. Computing and data management system

Fig. 9 presents the architecture of the Computing and Data Management System (CDMS). The first subsystem of CDMS is DAP (Data Access Platform) which collates and stores data obtained from the DAPS system. Externally, the Data Access Platform presents a selection of RESTful interfaces enabling authorized users to register new sensors, alter their properties and query for measurements using a variety of filtering options.

Data analyses are activated as a result of user interactions in the Decision support system GUI. When a given area of interest is set to the urgent mode, an execution environment for data analyses for this area is deployed on-demand in the cloud infrastructure. The deployment is orchestrated by the *Execution planner* component which generates a workflow describing the computational jobs that need to be performed in a given data analysis and, based on the requested SLAs (such as the Processing interval), it calculates the initial number of Application VMs that need to be allocated in the cloud. Actual execution of the workflow is coordinated by two components: the *Workflow engine* HyperFlow [46] responsible for enacting the workflow graph, and the *Scheduler* which plans the allocation of workflow tasks to a number of Application VMs where the actual application programs are deployed. The *Executor* component installed on the application VMs is responsible for the execution of individual tasks, fetching input data from the DAP subsystem, and writing results back to DAP's database, using its RESTful API. As the execution proceeds, it is constantly monitored by the *Autoscaler* component which may decide to adjust the number of VM instances depending on the current workload.

The Holistic Computing Controller controls *VM allocation policy* of the Autoscaler, and the *Scheduling policy* of the Scheduler. These configurable properties are described in the following section.

Please cite this article in press as: B. Balis, et al., Holistic approach to management of IT infrastructure for environmental monitoring and decision support systems with urgent computing capabilities, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.08.007

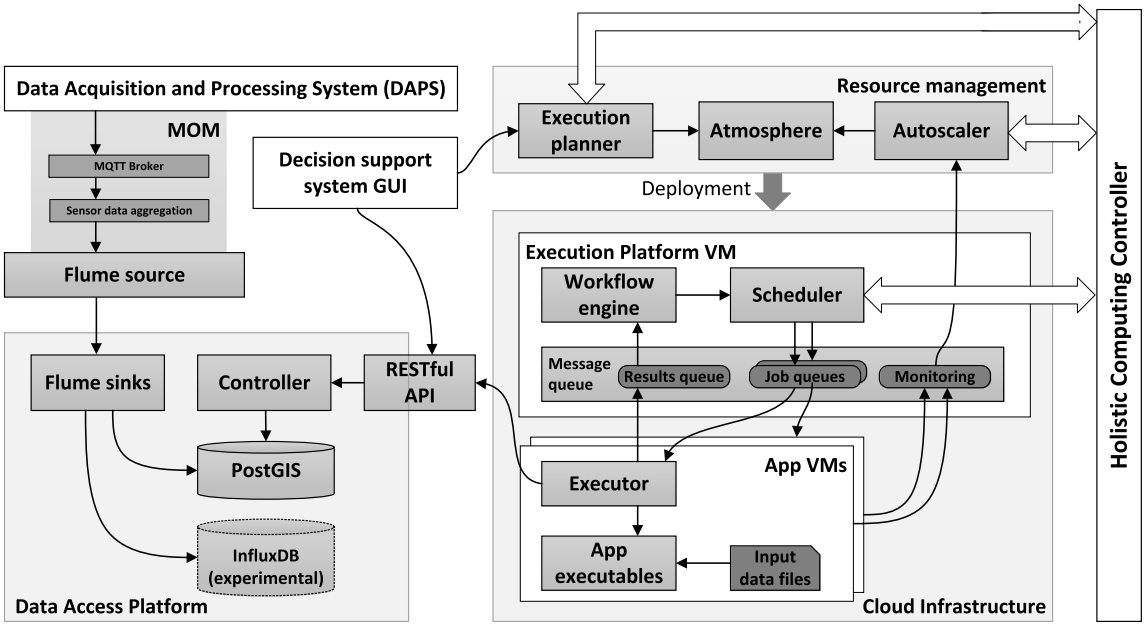

Fig. 9. Architecture of the computing and data management subsystem.

5.3. Configurable properties

Each layer of the ISMOP system comprises resources that can be managed through the following configurable properties:

- *Processing interval* indicates how often data analyses are performed in the data processing subsystem. Note that the objective function, DPI, is simply equal to the value of the *Processing interval*.
- *Scheduling policy* specifies whether the execution of data analyses should be optimized towards cost or turnaround time.
- *VM allocation policy* denotes circumstances under which VM instances will be allocated or deallocated. In the aggressive policy new VMs will be allocated earlier and deallocated later in comparison to the conservative policy, resulting in higher cost and better performance.
- *Transmission protocol order* states which communication protocols should be preferred in case of connectivity problems. Preferring XBee results in lower cost but less reliable connectivity, while preferring GPRS increases both cost and reliability.
- *Security* determines whether security features such as encryption should be turned on or off.
- *Data aggregation* determines for how long the sensor data collected by the measurement subsystem can be buffered in the edge computing subsystem before being transmitted to the data management subsystem. High aggregation time saves a considerable amount of energy by minimizing the activity of wireless network interfaces.
- *Measurement accuracy* indicates the precision with which data is collected from sensors. Increased accuracy (*High* option) is associated with greater energy consumption.
- *Measurement time* indicates how often the sensors perform their respective measurements. Note that the objective function, DTI, is simply equal to the value of *Measurement time*.

Each configurable property has an associated policy that is deployed in the appropriate subsystem in response to changing the value of the property. The policies are designed in such a way that their deployment does not disrupt the features of the system, e.g. its capability for data acquisition, transmission and processing. Table 2 presents possible values of configurable properties that can be assigned to specific resources in the ISMOP system.

6. Case study

In order to practically validate the proposed holistic approach to system management we have performed a series of experiments using prototype implementations of hardware and software components of the ISMOP IT infrastructure. The validation involved the following steps:

1. We have identified the decision space, i.e. key configurable properties for all subsystems of the ISMOP IT platform and their possible values.
2. We have identified the objective space, i.e. the objective functions, and developed their corresponding simplified models.
3. We have calculated the configurations that would have been adopted in the system managed according to the isolated approach.
4. We have implemented a prototype of the Holistic Computing Controller and subsequently defined constraints and trade-offs for the system operating in the normal and urgent modes. These constraints were then used to calculate optimal configurations of the system in the following variants: normal vs. urgent mode, with and without trade-off resolution.
5. We have compared the configurations obtained in different system management variants and operating modes.

The decision space and objective space were mapped out by technical experts during the system design phase. Service profiles are provided by end users of the system—the domain experts.

6.1. Configurable properties: decision space

Table 2 presents the most important configurable properties and their possible values (configuration options) for five different subsystems of the ISMOP IT infrastructure. Overall, there are 2880 possible configurations which meet functional requirements.

The size of the solution space can be initially reduced if we discard the configurations which have little or no significance for the system operation. The security mechanisms, which are implemented using hardware-accelerated techniques, have no significant effect on the overall system performance, including computation time, length of transmitted payloads and energy efficiency. Moreover, the sensor network inquiry timing accuracy

Please cite this article in press as: B. Balis, et al., Holistic approach to management of IT infrastructure for environmental monitoring and decision support systems with urgent computing capabilities, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.08.007

Table 2
Configurable properties of the ISMOP system.

Configurable property	Options	Policy	Resource
Data processing			
Processing interval (min)	5, 15, 60, 720, 1440	Deliver computation results every chosen interval	Scheduler
Scheduling policy	Cost-optimized Time-optimized	Use scheduling algorithm minimizing cost. Use scheduling algorithm minimizing turnaround time.	Scheduler
Computing infrastructure			
VM allocation policy	Conservative Aggressive	Allocate/deallocate VMs Allocate/deallocate VMs	Autoscaler
Communication			
Transmission protocol order	Preferred XBee Preferred GPRS	After 5 unsuccessful tries use the next technology in the following order: XBee, SMS, GPRS After 5 unsuccessful tries use the next technology in the following order: GPRS, SMS, XBee	MCU (Selector)
Edge computing			
Security	On Off	Switch on security features Switch off security features	MCU
Aggregation	High Low None	Enable data aggregation and send data in bulks just before processing time (possible only when processing time is known) Enable data aggregation and send data in bulk in the middle and just before processing time (possible only when processing time is known) Disable data aggregation and send data in bulk	MCU
Measurement			
Accuracy	High Low	Enable high accuracy in measurement network Disable high accuracy in measurement network	Sensor concentrator
Measurement time (min)	1, 5, 15, 60, 720, 1440	Measure temperature every chosen interval	Sensor concentrator

can also be set to a high value with little impact on the energy efficiency of the control-measurement station. Thus, it was decided to permanently enable the security mechanisms and set the acquisition timing accuracy to *High*. Those settings reduced the number of configurations to 720. The relation between acquisition sampling intervals and processing time in CDMS also allowed us to further reduce the number of configuration options. It is obvious that the data acquisition rate should be equal to or greater than the processing time in CDMS. This further reduces the number of options to 480.

6.2. Non-functional properties: objective space

We have chosen three most important non-functional properties as objective functions: operating cost (OPC), energy efficiency (EE), and Timeliness (TML). The latter property is an aggregated measure of the system's performance, responsiveness and capability to deliver timely results. TML is related to other more basic properties, described in Section 4.2: Data Transfer Interval (DTI), Data Processing Interval (DPI), and Data Processing Time (DPT). For the purpose of the optimization algorithm, each objective function has been normalized so that it maps a configuration vector to a value between 0.0 (minimum) and 1.0 (maximum).

The objective functions are calculated as follows. First, we have investigated the effect of configuration options described in Table 2 upon the values of the objective functions. The results of this investigation are summarized in Table 3. Each configurable property either has no effect on a given objective function (crossed-out fields), or it can contribute to its low (LOW = 0.0), medium (MID = 0.5) or high (HIGH = 1.0) value. The objective function is then calculated as a mean of contributing configuration properties, resulting in a value between 0.0 and 1.0. For example, OPC is calculated as a mean value of four contributing configuration options: Processing Interval (5 min = HIGH, 1440 min = LOW), Computing Schedule, VM allocation and Protocol order.

In addition, we have discovered that the effect on energy efficiency also highly depends on current weather conditions [47], hence there are two separate columns for this objective. This phenomenon stems from the fact that the control and measurement station utilizes solar cells for charging its batteries.

OPC is mainly influenced by CDMS configuration and the selected means of communication between the control-measurement station and CDMS. Utilizing the GPRS network carries much higher costs compared to e.g. XBee, which requires no commercial infrastructure. EE is determined mainly by the control-measurement station configuration. TML should be considered as holistic because it is influenced by all layers of the system. Finally, the *Weather* contextual factor (*cloudy* or *sunny*) tells us how much energy can be harvested using photovoltaic cells and has a significant impact on the autonomously-powered control-measurement stations. Table 3 presents the details on how each objective function is affected by different configuration settings. In the following two sections we will discuss the results of the system management using the isolated approach and the holistic approach respectively.

In many cases, interviews with experts or literature studies were sufficient to estimate the influence of configurable properties on the objective functions. However, in the case of the EE function, extensive experimental studies were conducted in order to measure energy consumption of sensor data transmission devices in various configurations. The summary of key findings from these experiments is shown in Fig. 10. We have found that the aggregation level (configurable property of the Edge computing layer) has a decisive influence on the energy consumption of GPRS-based transmission. When the power saving mode was enabled in the GPRS modem driver but the aggregation period was short, data transmission actually required considerably more energy compared to transmission without power saving. This is due to the fact that the GPRS modem requires an additional amount of energy for the wake-up procedure each time it leaves the sleep mode. Increasing the aggregation period extends the deep sleep periods thereby decreasing power requirements. Considering the parameters of the utilized hardware and software, the aggregation interval of 4 min is the critical value below which *disabling* power saving is always better than keeping power saving *enabled*.

Please cite this article in press as: B. Balis, et al., Holistic approach to management of IT infrastructure for environmental monitoring and decision support systems with urgent computing capabilities, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.08.007

Table 3
Objective space evaluation matrix. HIGH, MID and LOW values denote that the given option contributes to a high/moderate/low value of the given objective function.

Conf. property	Option	OPC (MIN)	EE (Sunny) (MAX)	EE (Cloudy) (MAX)	TML (MAX)
Processing interval (min)	5, 10, 60, 720, 1440	HIGH–LOW	–	–	HIGH–LOW
Computing schedule	Cost-optimized	LOW	–	–	LOW
	Time-optimized	HIGH	–	–	HIGH
VM allocation	Conservative	LOW	–	–	LOW
	Aggressive	HIGH	–	–	HIGH
Protocol order	Preferred XBee	LOW	HIGH	HIGH	LOW
	Preferred GPRS	HIGH	HIGH	LOW	HIGH
Security	On	–	–	–	–
	Off	–	–	–	–
Aggregation	High	–	HIGH	HIGH	LOW
	Low	–	MID	MID	MID
	None	–	LOW	LOW	HIGH
Accuracy	High	–	–	–	–
	Low	–	–	–	–
Measurement time (min)	1, 5, 15, 60, 720, 1440	–	LOW–HI	LOW–HI	HIGH–LOW

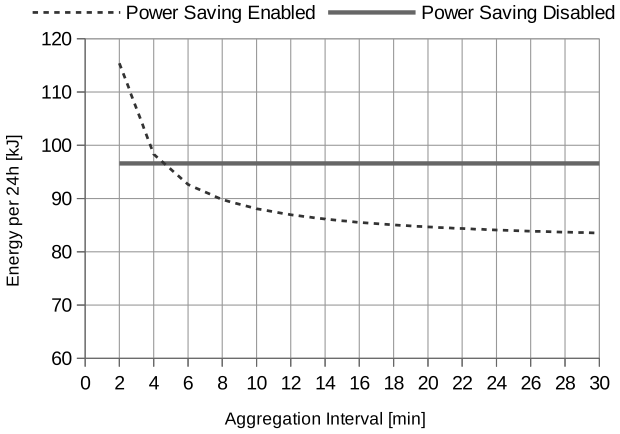

Fig. 10. Estimated energy requirements for GPRS-based transmission of sensor data over a 24-h period depending on the aggregation level.

6.3. Results: isolated approach

In the isolated approach all subsystems choose their configuration settings in isolation i.e. without knowledge of the options selected by other subsystems. For example, short measurement time and long processing interval imply that it is not necessary to transmit the measured data immediately following measurement. Unfortunately, when the Edge Layer is not aware of the processing time, it cannot make a decision to aggregate its data.

In the isolated approach the system may work in normal and urgent profiles defined by SLA. In order to investigate how the data aggregation in the Edge computing Layer influences the objective space, let us analyze the subset of solutions from S that minimize OPC (Table 4, 16 results). In the isolated approach only those solutions where Aggregation is set to *None* are possible. This introduces a bias in favor of solutions with greater timeliness and lower energy efficiency.

6.4. Results: holistic approach

In the holistic approach the optimal configurations are calculated by HCC using the algorithm described earlier (see Fig. 4). Figs. 11 and 12 visually present the Pareto sets found by the HCC in our case study example (for the normal and urgent modes, respectively, during cloudy weather conditions). The charts only show the values of the three objective functions: TML, OPC and EE, but not the configurations that led to these values. These are discussed in the following sections where the optimal solutions for *normal* and *urgent* profiles are presented. We also show what results would be obtained without trade-off resolution.

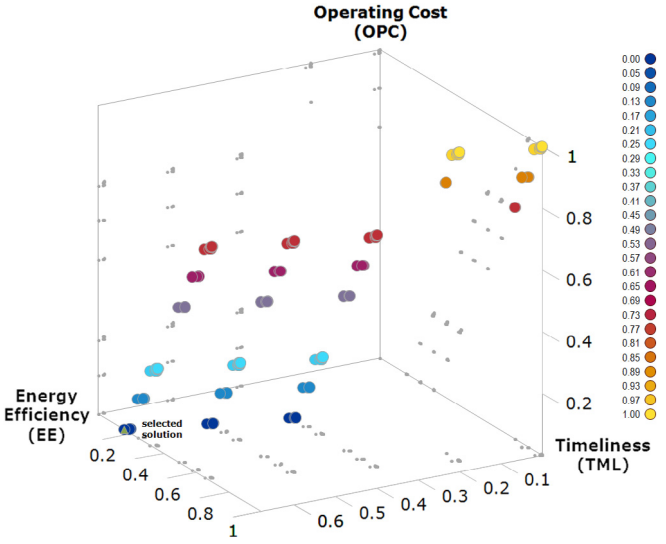

Fig. 11. Pareto-optimal solutions for the case study example found using the holistic approach (normal mode).

Note that in the normal mode it is possible to find solutions that maximize TML or minimize OPC, but the maximum achievable EE is 0.67. In the urgent mode, in turn, the lowest possible OPC is 0.24. The reason for this is that some solutions have been eliminated by the SLA restrictions.

6.4.1. Normal profile

Under *normal* conditions (no flood risk) the SLA assumes DMI of up to 1 h and DPI of up to 12 h. After the first step of the HCC algorithm, the SLA requirements leaves us with 408 solutions from which 128 solutions for cloudy weather and 102 solutions for sunny weather are Pareto-optimal. Without trade-off resolution one of these solutions would have to be chosen based on simple criteria. We have decided that in the normal profile the most important factor is the operating cost (OPC), followed by energy efficiency (EE) and – least importantly – timeliness (TML). Table 6 shows a solution for the normal profile with values of OPC, EE and TML equal to 0.0, 0.68, 0.16 respectively.

In the second step of HCC algorithm, using trade-off resolution, we can calculate the importance of all non-functional properties via pairwise comparison. The system should operate with the lowest possible operating costs (OPC) and with high energy efficiency (EE)—thus we assume trade-offs between OPC, EE, and TML:

Please cite this article in press as: B. Balis, et al., Holistic approach to management of IT infrastructure for environmental monitoring and decision support systems with urgent computing capabilities, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.08.007

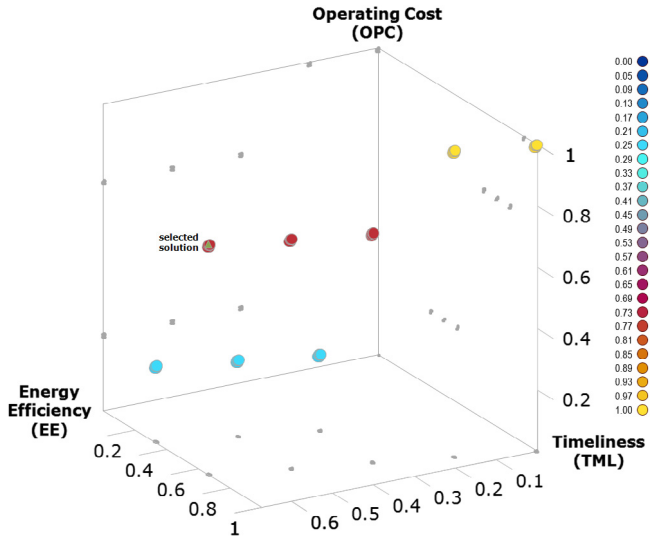

Fig. 12. Pareto-optimal solutions for the case study example found using the holistic approach (urgent mode).

- OPC is slightly more important than EE;
- EE is much more important than TML;
- OPC is much more important than TML.

After processing trade-offs using the AHP method we assign objective weights to each trade-off: 0.61 for OPC, 0.30 for EE, and 0.09 for FT (Fig. 13). By applying these weights to all 128 potential solutions we obtain one best solution, which happens to be the same for both sunny and cloudy weather. In this case, the final configurations are identical with and without trade-off resolution, as presented in Table 5.

6.4.2. Urgent profile

The SLA requirements for the *urgent* profile involve DMI of not more than 15 min and DPI of not more than 30 min. After the first step of HCC algorithm, the number of solutions that meet the SLA is 192 from which 64 solutions for cloudy weather and 48 solutions for sunny weather are Pareto-optimal. Without trade-off resolution one of these solutions would have to be chosen based on simple criteria. We have decided that in the urgent profile the most important factor is timeliness (TML), followed by energy efficiency (EE) and operating cost (OPC). Table 6 presents a solution for the urgent profile where the values of OPC, EE and TML are 1.0, 0.33, 1.0 respectively. The selected solution minimizes measurement and processing time—with values much lower than the acceptable values specified in the SLA.

In urgent profile, excess timeliness might not improve the quality of levee breach prediction as it does not materially affect the underlying mathematical models. Moreover, the system should preserve some energy as it might be impossible to obtain solar power during a natural disaster. The trade-off resolution, carried out as the second step of the HCC algorithm, will help choose the best solution based on various criteria. The trade-offs are as follows:

- EE is slightly more important than TML;
- TML is much more important than OPC;
- EE is much more important than OPC.

We apply the following objective weights to the trade-offs: 0.09 for OPC, 0.62 for EE, and 0.30 for TML (Fig. 13). We then obtain one best solution for each weather context. During sunny days the system is able to harvest more energy than during cloudy weather. Consequently, HCC select data transmission technology based on the weather conditions. The final energy efficiency objective remains unchanged at 0.67, while TML and OPC objectives are deemed less important. Nevertheless the TML objective in the urgent profile is better than in the normal profile, as shown in Table 5.

6.5. Discussion

The obtained results confirm that, thanks to the holistic approach, HCC can perform global optimization of the system's configuration and achieve better configuration settings than would have been possible had all subsystems been configured in isolation, based on SLAs pertaining only to these subsystems.

In the presented case only a holistic view of the system justifies setting the *Data aggregation* property of the Edge computing subsystem to *high* – i.e. at least 12 h – in the normal mode (Table 5), even though the *Measurement interval* is much lower—only 60 min. As shown, HCC can take into account the constraint imposed on the *Processing interval* (a property of a completely different subsystem) which, in this case, was much higher than the Measurement interval. This, along with the rule that $Aggregation\ time < DPI - DPT$, enabled finding the configuration which maximizes EE (and increases system lifetime). Without the HCC, the aggregation property would have been set to *none* by the Edge computing subsystem because there would have been no indication that sensor data was not required immediately by the upper subsystems. This, in turn, would have resulted in greater power consumption at the edge node as communication would have to be initiated more frequently. Consequently, without HCC (isolated approach), the values of OPC, EE and TML would have been equal to 0.0, 0.34, and 0.32, respectively. Given that in the normal mode EE is more important than TML, clearly it would have been a worse configuration compared to the one achieved with the holistic approach (OPC = 0.0, EE = 0.68, TML = 0.16).

The second observation proves that trade-off resolution can further improve the achieved configuration. Table 6 shows the solutions produced by the HCC for normal and urgent profiles, where instead of the trade-off resolution, a simple importance hierarchy among the objective functions was assumed. Thus, the final configuration chosen from the Pareto set was simply one that produced the best value of the most important objective. When several such configurations existed, the second most important objective was taken into account, etc. The assumed importance hierarchy depended on the mode of operation: in the normal mode the corresponding order was (from most to least important): OPC, EE, TML; while in the urgent mode it was: TML, EE, OPC. Trade-off resolution proved to be a better decision-making mechanism that led to different configurations depending on the context (weather), as shown in Table 5. Without trade-off resolution, a configuration was chosen that simply maximized TML, but at the cost of very low energy efficiency (0.33 in sunny and 0.00 in cloudy weather conditions). Introducing the trade-off resolution algorithm resulted in more balanced configurations in which TML was still high, but the system lifetime was also much higher due to much better energy efficiency (0.67). Moreover, the model accurately captured the fact that during cloudy weather the importance of energy efficiency rises (because the solar cells become less efficient). Consequently, in the configuration selected for cloudy weather, the XBee protocol is chosen, which allows sustaining high EE at the cost of TML.

Finding the objective functions that map system configuration and context onto its non-functional properties is by far the most difficult task. We have adopted a simplified approach to obtaining the models of the objective functions which proved sufficient for research purposes. However, in a real system more accurate models might be necessary. Such models can be diverse for different objective functions. For example, it can be relatively easy to find an accurate analytical formula for the cost function, but for other functions different approaches might be more effective, such as a simulation model, a model based on machine learning techniques, etc.

Please cite this article in press as: B. Balis, et al., Holistic approach to management of IT infrastructure for environmental monitoring and decision support systems with urgent computing capabilities, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.08.007

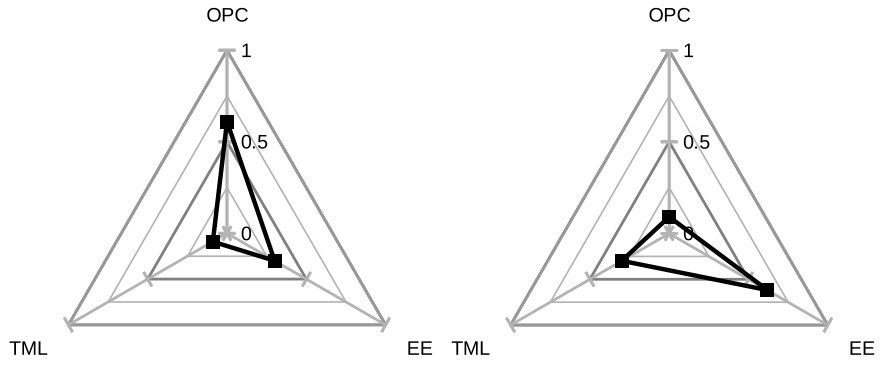

Fig. 13. Comparison of quality objective weights for trade-off resolution in: (a) normal mode, (b) urgent mode.

Table 4
All solutions (system configurations) minimizing OPC. In the isolated approach only suboptimal configurations with Aggregation set to *None* are possible.

Resources						Context	OPC	EE	TML
Processing interval	Computing schedule	VM allocation	Prot. order	Aggregation	Meas. time				
1440	Cost-opt.	Conservative	XBee	None	1	Sun./Cloudy	0.0	0.33	0.33
1440	Cost-opt.	Conservative	XBee	Low	1	Sun./Cloudy	0.0	0.50	0.25
1440	Cost-opt.	Conservative	XBee	High	1	Sun./Cloudy	0.0	0.67	0.17
1440	Cost-opt.	Conservative	XBee	None	5	Sun./Cloudy	0.0	0.33	0.33
1440	Cost-opt.	Conservative	XBee	Low	5	Sun./Cloudy	0.0	0.50	0.25
1440	Cost-opt.	Conservative	XBee	High	5	Sun./Cloudy	0.0	0.67	0.17
1440	Cost-opt.	Conservative	XBee	None	15	Sun./Cloudy	0.0	0.34	0.33
1440	Cost-opt.	Conservative	XBee	Low	15	Sun./Cloudy	0.0	0.50	0.25
1440	Cost-opt.	Conservative	XBee	High	15	Sun./Cloudy	0.0	0.67	0.16
1440	Cost-opt.	Conservative	XBee	None	60	Sun./Cloudy	0.0	0.34	0.32
1440	Cost-opt.	Conservative	XBee	Low	60	Sun./Cloudy	0.0	0.51	0.24
1440	Cost-opt.	Conservative	XBee	High	60	Sun./Cloudy	0.0	0.68	0.16
1440	Cost-opt.	Conservative	XBee	None	720	Sun./Cloudy	0.0	0.50	0.25
1440	Cost-opt.	Conservative	XBee	Low	720	Sun./Cloudy	0.0	0.67	0.17
1440	Cost-opt.	Conservative	XBee	High	720	Sun./Cloudy	0.0	0.83	0.08
1440	Cost-opt.	Conservative	XBee	High	1440	Sun./Cloudy	0.0	1.0	0.00

Table 5
Optimal solutions (system configurations) for normal and urgent profiles obtained using tradeoff resolution.

Profile	Context	Resources						OPC	EE	TML
		Processing interval (min)	Computing schedule	VM allocation	Prot. order	Aggregation	Meas. time			
Normal	Sunny or cloudy	1440	Cost-optimized	Conservative	XBee	High	60	0.00	0.68	0.16
Urgent	Sunny	15	Time-optimized	Aggressive	GPRS	High	15	1.00	0.67	0.83
	Cloudy	15	Time-optimized	Aggressive	XBee	High	15	0.75	0.67	0.66

Table 6
Optimal solutions (system configurations) for normal and urgent profiles obtained without tradeoff resolution.

Profile	Context	Resources						OPC	EE	TML
		Processing interval (min)	Computing schedule	VM allocation	Prot. order	Aggregation	Meas. time			
Normal	Sunny or cloudy	1440	Cost-optimized	Conservative	XBee	High	60	0.00	0.68	0.16
Urgent	Sunny	5	Time-optimized	Aggressive	GPRS	High	1	1.00	0.33	1.00
	Cloudy	5	Time-optimized	Aggressive	GPRS	High	1	1.00	0.00	1.00

7. Conclusion

We presented a holistic approach to management of an IT infrastructure for environmental monitoring and decision support systems based on the Internet of Things and cloud computing technologies. We introduced the Holistic Computing Controller, a component complementing the IoT-Cloud stack, which calculates and deploys a globally optimal configuration for all subsystems based on knowledge of the system as a whole. The approach was experimentally validated using the hardware and software components developed in the ISMOP system for smart levee monitoring and flood decision support. The results confirm that the holistic approach produces a better configuration of the system in comparison to what can be achieved with the so-called isolated approach, where the configuration of all subsystems is managed in isolation.

Future work involves further experiments aimed at providing more precise objective functions, including an experimental study of additional configurable properties of the system, and their interplay across different subsystems. Another prospective research direction is development of prototype components of the ISMOP system, in particular an efficient solver for multi-criteria optimization within the Holistic Computing Controller.

Please cite this article in press as: B. Balis, et al., Holistic approach to management of IT infrastructure for environmental monitoring and decision support systems with urgent computing capabilities, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.08.007

Acknowledgments

This work is partially supported by the National Centre for Research and Development (NCBiR), Poland, project PBS1/B9/18/2013; AGH statutory research Grant No. 11.11.230.124 is also acknowledged. The authors are grateful to Prof. Robert Meijer (UvA and TNO, The Netherlands) and to the entire ISMOP project team for fruitful discussions and suggestions.

References

[1] A. Botta, W. de Donato, V. Persico, A. Pescapé, Integration of cloud computing and Internet of things: a survey, Future Gener. Comput. Syst. 56 (2016) 684–700.
[2] S.H. Leong, D. Kranzlmüller, Towards a general definition of urgent computing, Procedia Comput. Sci. 51 (2015) 2337–2346.
[3] CISCO, The Internet of Things Reference Model, 2014. Available online: http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.pdf (accessed 14.03.16).
[4] S.H. Leong, A. Frank, D. Kranzlmüller, Leveraging e-infrastructures for urgent computing, in: ICCS, in: Procedia Computer Science, vol. 18, Elsevier, 2013, pp. 2177–2186.
[5] R. Tolosana-Calasanz, J. Banares, O. Rana, C. Pham, E. Xydas, C. Marmaras, P. Papadopoulos, L. Cipcigan, Enforcing quality of service on opennebula-based shared clouds, in: 2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), 2014, pp. 651–659. http://dx.doi.org/10.1109/CCGrid.2014.50.
[6] J. Cope, H. Tufo, Supporting storage resources in urgent computing environments, in: 2008 IEEE International Conference on Cluster Computing, 2008, pp. 348–353. http://dx.doi.org/10.1109/CLUSTR.2008.4663794.
[7] R. Brzoza-Woch, M. Konieczny, B. Kwolek, P. Nawrocki, T. Szydlo, K. Zielinski, Holistic approach to urgent computing for flood decision support, Procedia Comput. Sci. 51 (2015) 2387–2396. proceedings of the International Conference on Computational Science, ICCS 2015.
[8] A. Pieta, J. Bala, M. Dwornik, K. Krawiec, Stability of the levees in case of high level of the water, in: 14th SGEM Geoconference On Informatics, Geoinformatics and Remote Sensing–Conference Proceedings, Vol. 1, 2014, pp. 809–815.
[9] B. Balis, M. Kasztelnik, M. Malawski, P. Nowakowski, B. Wilk, M. Pawlik, M. Bubak, Execution management and efficient resource provisioning for flood decision support, Procedia Comput. Sci. 51 (2015) 2377–2386. proceedings of the International Conference on Computational Science, ICCS 2015.
[10] K.K. Droegemeier, V. Ch, R. Clark, D. Gannon, S. Graves, M. Ramamurthy, R. Wilhelmson, K. Brewster, B. Domenico, T. Leyton, D. Weber, A. Wilson, M. Xue, S. Yalda, Linked environments for atmospheric discovery (lead): A cyberinfrastructure for mesoscale meteorology research and education, in: 20th Conf. on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology, 2004.
[11] Y. Cui, R. Moore, K. Olsen, A. Chourasia, P. Maechling, B. Minster, S. Day, Y. Hu, J. Zhu, A. Majumdar, T. Jordan, Enabling very-large scale earthquake simulations on parallel machines, in: Computational Science - ICCS 2007, in: Lecture Notes in Computer Science, vol. 4487, Springer, Berlin, Heidelberg, 2007, pp. 46–53. URL http://dx.doi.org/10.1007/978-3-540-72584-8_7.
[12] P.S. Bogden, T. Gale, G. Allen, J. MacLaren, G.A. , G. Creager, J. Bintz, L.D. Wright, H. Graber, N. Williams, S. Graves, H. Conover, K. Galluppi, R. Luettich, Architecture of a community infrastructure for predicting and analyzing coastal inundation, Mar. Technol. Soc. J. 41 (1) (2007) 53–71.
[13] J. Mandel, J.D. Beezley, L.S. Bennethum, S. Chakraborty, J.L. Coen, C.C. Douglas, J. Hatcher, M. Kim, A. Vodacek, A dynamic data driven wildland fire model, in: Computational Science - ICCS 2007, in: Lecture Notes in Computer Science, vol. 4487, Springer, Berlin, Heidelberg, 2007, pp. 1042–1049. URL http://dx.doi.org/10.1007/978-3-540-72584-8_137.
[14] R. Strijkers, W. Toorop, A. van Hoof, P. Grosso, A. Belloum, D. Vasuining, C. de Laat, R. Meijer, Amos: Using the cloud for on-demand execution of e-science applications, in: 2010 IEEE Sixth International Conference on e-Science (e-Science), IEEE, 2010, pp. 331–338.
[15] Open Geospatial Consortium, OpenGIS SWE Service Model - Implementation Standard, version 2.0 (3 2011).
[16] P. Beckman, S. Nadella, N. Trebon, I. Beschastnikh, Spruce: A system for supporting urgent high-performance computing, in: Grid-Based Problem Solving Environments, Springer, 2007, pp. 295–311.
[17] K. Yoshimoto, D. Choi, R. Moore, A. Majumdar, E. Hocks, Implementations of urgent computing on production hpc systems, Procedia Comput. Sci. 9 (2012) 1687–1693.
[18] B. Balis, M. Kasztelnik, M. Bubak, T. Bartynski, T. Gubała, P. Nowakowski, J. Broekhuijsen, The urbanflood common information space for early warning systems, Procedia Comput. Sci. 4 (2011) 96–105.
[19] V. Krzhizhanovskaya, G. Shirshov, N. Melnikova, R. Belleman, F. Rusadi, B. Broekhuijsen, B. Gouldby, J. Lhomme, B. Balis, M. Bubak, A. Pyayt, I. Mokhov, A. Ozhigin, B. Lang, R. Meijer, Flood early warning system: design, implementation and computational modules, Procedia Comput. Sci. 4 (2011) 106–115. http://dx.doi.org/10.1016/j.procs.2011.04.012.
[20] B. Balis, T. Bartynski, M. Bubak, G. Dyk, T. Gubala, M. Kasztelnik, A development and execution environment for early warning systems for natural disasters, in: Cluster, Cloud and Grid Computing (CCGrid) 2013, IEEE, 2013, pp. 575–582.
[21] K.V. Knyazkov, S.V. Kovalchuk, T.N. Tchurov, S.V. Maryin, A.V. Boukhanovsky, Clavire: e-science infrastructure for data-driven computing, J. Comput. Sci. 3 (6) (2012) 504–510.
[22] S.V. Kovalchuk, P.A. Smirnov, S.V. Maryin, T.N. Tchurov, V.A. Karbovskiy, Deadline-driven resource management within urgent computing cyberinfrastructure, Procedia Comput. Sci. 18 (2013) 2203–2212.
[23] R. Brzoza-Woch, Ł. Czekierda, J. Długopolski, P. Nawrocki, M. Psiuk, T. Szydło, W. Zaborowski, K. Zieliński, D. Żmuda, Implementation, deployment and governance of SOA adaptive systems, in: Advanced SOA Tools and Applications, Springer Berlin, Heidelberg, Berlin, Heidelberg, 2014, pp. 261–323. (Chapter) URL http://dx.doi.org/10.1007/978-3-642-38957-3_6.
[24] K. Zielinski, T. Szydlo, R. Szymacha, J. Kosinski, J. Kosinska, M. Jarzab, Adaptive SOA solution stack, IEEE Trans. Serv. Comput. 5 (2) (2012) 149–163.
[25] J. Cope, N. Trebon, H. Tufo, P. Beckman, Robust data placement in urgent computing environments, in: IEEE International Symposium on Parallel Distributed Processing, 2009. IPDPS 2009. 2009, pp. 1–13. http://dx.doi.org/10.1109/IPDPS.2009.5160914.
[26] V. Krzhizhanovskaya, G. Shirshov, N. Melnikova, R. Belleman, F. Rusadi, B. Broekhuijsen, B. Gouldby, J. Lhomme, B. Balis, M. Bubak, A. Pyayt, I. Mokhov, A. Ozhigin, B. Lang, R. Meijer, Flood early warning system: design, implementation and computational modules, in: ICCS, in: Procedia Computer Science, vol. 4, Elsevier, 2011, pp. 106–115.
[27] M. Konieczny, Enriching WSN environment with context information, Comput. Sci. 13 (4) (2012) 101. http://dx.doi.org/10.7494/csci.2012.13.4.101, URL http://journals.agh.edu.pl/csci/article/view/47.
[28] A. Pięta, M. Lupa, M. Chuchro, A. Piórkowski, A. Leśniak, A model of a system for stream data storage and analysis dedicated to sensor networks of embankment monitoring, in: Computer Information Systems and Industrial Management, in: Lecture Notes in Computer Science, vol. 8838, Springer, Berlin, Heidelberg, 2014, pp. 514–525. URL http://dx.doi.org/10.1007/978-3-662-45237-0_47.
[29] J. Kosinski, P. Nawrocki, D. Radziszowski, K. Zielinski, S. Zielinski, G. Przybylski, P. Wnek, SLA monitoring and management framework for telecommunication services, in: Fourth International Conference on Networking and Services, 2008. ICNS 2008. 2008, pp. 170–175. http://dx.doi.org/10.1109/ICNS.2008.31.
[30] R. Levins, R. Lewontin, The Dialectical Biologist, Harvard University Press, 1985, URL https://books.google.es/books?id=DKK-xiZKeoC.
[31] F.J. Varela, On being autonomous: The lessons of natural history for systems theory, in: G.J. Klir (Ed.), Applied General Systems Research, in: NATO Conference Series, vol. 5, Springer, US, 1978, pp. 77–84. URL http://dx.doi.org/10.1007/978-1-4757-0555-3_5.
[32] C. Harrison, B. Eckman, R. Hamilton, P. Hartswick, J. Kalagnanam, J. Paraszczak, P. Williams, Foundations for smarter cities, IBM J. Res. Dev. 54 (4) (2010) 1–16. http://dx.doi.org/10.1147/JRD.2010.2048257.
[33] ONF, Software-defined networking: The new norm for networks, Tech. Rep., Open Networking Foundation, 2012, URL https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf.
[34] Great Britain. Cabinet Office, ITIL Service Transition, AXELOS - global best practice, TSO, 2011.
[35] T. Saaty, Decision making with the analytic hierarchy process, Int. J. Serv. Sci. 1 (2008) 83–98.
[36] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the Internet of things, in: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, MCC '12, ACM, New York, NY, USA, 2012, pp. 13–16. http://doi.acm.org/10.1145/2342509.2342513.
[37] T. Szydlo, S. Gut, B. Puto, Smart applications: Discovering and interacting with constrained resources IPv6 enabled devices, Prz. Elektrotech. (2013) 221–226.
[38] R. Ierusalimschy, L.H. de Figueiredo, W. Celes, The evolution of lua, in: Proceedings of the Third ACM SIGPLAN Conference on History of Programming Languages, HOPL III, ACM, New York, NY, USA, 2007, pp. 2-1–2-26. http://doi.acm.org/10.1145/1238844.1238846.
[39] R. Faludi, Building Wireless Sensor Networks: With ZigBee, XBee, Arduino, and Processing, first ed., O'Reilly Media, Inc., 2010.
[40] P. Nawrocki, M. Jakubowski, T. Godzik, Analysis of notification methods with respect to mobile system characteristics, in: 2015 Federated Conference on Computer Science and Information Systems (FedCSIS), 2015, pp. 1183–1189. http://dx.doi.org/10.15439/2015F6.
[41] A. Banks, R. Gupta, Mqtt version 3.1.1, Tech. Rep., Organization for the Advancement of Structured Information Standards - OASIS, 2015, URL http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/errata01/os/mqtt-v3.1.1-errata01-os-complete.html.
[42] T. Szydlo, P. Nawrocki, R. Brzoza-Woch, K. Zielinski, Power aware MOM for telemetry-oriented applications using GPRS-enabled embedded devices – levee monitoring use case, in: Proc. 2014 Federated Conference on Computer Science and Information Systems, in: Annals of Computer Science and Information Systems, vol. 2, IEEE, 2014, pp. 1059–1064. URL http://dx.doi.org/10.15439/2014F252.
[43] R. Brzoza-Woch, J. Długopolski, P. Nawrocki, K. Zieliński, Application of fpga integrated circuits for acquisition and providing information in compliance with web service model, Prz. Elektrotech. 89 (7) (2013) 340–347.
[44] P. Bachara, R. Brzoza-Woch, J. Długopolski, P. Nawrocki, A. Ruta, W. Zaborowski, K. Zieliński, Construction of hardware components for the Internet of services, Comput. Inform. 34 (4) (2015).
[45] R. Brzoza-Woch, P. Nawrocki, Fpga-based web services – infinite potential or a road to nowhere? IEEE Internet Comput. 20 (1) (2016) 44–51. http://dx.doi.org/10.1109/MIC.2015.23.

Please cite this article in press as: B. Balis, et al., Holistic approach to management of IT infrastructure for environmental monitoring and decision support systems with urgent computing capabilities, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.08.007

[46] B. Balis, Hyperflow: A model of computation, programming approach and enactment engine for complex distributed workflows, Future Gener. Comput. Syst. 55 (2016) 147–162. http://dx.doi.org/10.1016/j.future.2015.08.015, URL http://www.sciencedirect.com/science/article/pii/S0167739X15002770.
[47] T. Szydlo, R. Brzoza-Woch, Predictive power consumption adaptation for future generation embedded devices powered by energy harvesting sources, Microprocess. Microsyst. 39 (4–5) (2015) 250–258.

Bartosz Balis, Ph.D. in Computer Science, is an Assistant Professor at the Department of Computer Science, AGH University of Science and Technology. He is co-author of above 75 international publications including journal articles, conference papers, and book chapters. His research interests include environments for eScience, scientific workflows, grid and cloud computing. He has participated in international research projects including EU-IST CrossGrid, CoreGRID, K-Wf Grid, ViroLab, Gredia, UrbanFlood and PaaSage. He has served as a member of Program Committee for conferences: e-Science 2006, ICCS 2007–2016, ITU Kaleidoscope 2013–2016, SC16 Workshops, SIMULTECH 2016.

Robert Brzoza-Woch studied Electronics and Telecommunication with major in Sensors and Microsystems. He received his M.Sc. degree in 2009 and Ph.D. in computer science in 2013 from the AGH University of Science and Technology in Krakow, Poland. Currently, he works as an assistant professor at the Department of Computer Science, at the same university. He has broad experience in hardware and software design for distributed embedded systems, wireless sensor networks, telemetry systems, and home automation. His work is centered around embedded systems based on microcontrollers and FPGAs with particular emphasis on ARM cores and FPGA-based microprocessor systems.

Marian Bubak has an M.Sc. degree in Technical Physics and Ph.D. in Computer Science. He is an adjunct at the Institute of Computer Science and ACC Cyfronet AGH University of Science and Technology, Kraków, Poland, and a Professor of Distributed System Engineering at the University of Amsterdam. His research interests include collaborative environments, parallel and distributed computing, and eScience. He is the author of about 230 papers in this area, co-editor of about 30 proceedings of international conferences and the member of editorial boards of 3 journals. He served key roles in series of EU-funded projects, including CrossGrid (the Architecture Team leader), K-Wf Grid (the Scientific Coordinator), CoreGRID (member of the Monitoring Committee), and ViroLab, GREDIA, UrbanFlood, MAPPER and VPH-Share (WP leader).

Marek Kasztelnik graduated from the University of Science and Technology in Kraków, Poland, where he received his M.Sc. in Computer Science. In 2005 he joined Technologies Group in Research and Development Department in ComArch (2005–2010) where he was responsible for designing and developing solutions for telecommunication management and meta-modeling. In 2006 he has joined DICE team at Academic Computing Centre Cyfronet AGH where he has been involved in the EU-funded projects. He is author of more than 40 scientific articles. His current work addresses areas such as distributed resource discovery, distributed cloud based application design and implementation.

Bartosz Kwolek has received M.Sc. degrees in Computer Science and Architecture. He works for the Department of Computer Science at the AGH University of Science and Technology, Krakow, Poland, where he gained Ph.D. degree in 2014. His interests focus on computer networks, multimedia systems and web application technologies as well as user interface design. He is an active Cisco CNAP trainer. He has participated in several EU research projects (6WINIT, ProAccess) and national projects (IT-SOA, NGOSS, ISMOP).

Piotr Nawrocki, Ph.D., is an Assistant Professor in the Department of Computer Science at the AGH University of Science and Technology, Krakow, Poland. His research interests include distributed systems, computer networks, mobile systems, mobile cloud computing, Internet of Things and service-oriented architectures. He has participated in several EU research projects including MECCANO, 6WINIT, UniversAAL and national projects including IT-SOA and ISMOP. He is a member of the Polish Information Processing Society (PTI).

Piotr Nowakowski received M.Sc. in Computer Science from the AGH University of Science and Technology. He works at the Academic Computer Center AGH in Kraków as a scientific programmer, delivering custom IT solutions for various branches of e-science. He is involved in many research projects, including CrossGrid, ViroLab, VPH-Share and PL-Grid. His primary research interests focus on managing and optimizing access to large datasets processed with HPC resources.

Tomasz Szydlo received Ph.D. in Computer Science from the University of Science and Technology in Kraków, Poland in 2010. Since 2011 he is assistant professor at the Department of Computer Science at AGH-UST. His interests focus on Internet of Things as well as mobile and SOA systems. He has participated in several EU research projects including CrossGrid, Ambient Networks and UniversAAL and national projects including IT-SOA and ISMOP.

Krzysztof Zielinski is a full professor and head of Department of Computer Science at the AGH University of Science and Technology, Krakow, Poland. His interests focus on networking, mobile and wireless systems, service-oriented distributed systems engineering and Cloud computing. He is author of over 200 papers in this area. He served as an expert with Ministry of Science and Education. He is a member of IEEE, ACM and Polish Academy of Sciences, Computer Science Chapter. He served as a program committee member, chairman and organizer of several international conferences including MobiSys, ICCS, ICWS, IEEE SCC and many others.

Please cite this article in press as: B. Balis, et al., Holistic approach to management of IT infrastructure for environmental monitoring and decision support systems with urgent computing capabilities, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.08.007