1 Theme Topic and Brief Description: Dynamic Distributed Data-intensive Programming Abstractions and Systems (3DPAS)

Many problems at the forefront of science, engineering, medicine, and the social sciences, are increasingly complex and interdisciplinary due to the plethora of data sources and computational methods available today. A common feature across many of these problem domains is the amount and diversity of data and computation that must be integrated to yield insights. Such data is increasingly large-scale and distributed, arising from sensors, scientific instruments, scientific simulations, and Internet data clouds. Applications need access to these disparate data-sets and to the methods that operate on them. One solution is to "pour" the data and methods into a single platform (e.g., a cloud) for integration and execution. However, for many complex data-intensive applications, moving the data may have restrictions due to ownership or size issues. Furthermore, different computational methods may run best on different platforms rather than within a uniform cloud, and applications are also increasingly integrating third-party services (e.g. GoogleEarth). Therefore, we believe data-intensive applications need first-class support for distributed heterogeneous platform execution.

There has been a lot of effort in managing & distributing tasks where computational loads are dominant. Such applications have after all, been historically the drivers of "grid" computing. There has however, been relatively less effort on tasks where the computational load is matched by the data-load, or even dominated by the data-load. For such tasks to be able to operate at scale, there are conceptually simple run-time trade-offs that need to be made, such as, (i) determining whether to move data to computational nodes, versus (ii) keep data localised and move computational tasks to operate on the data *in situ*, or (iii) possibly neither, and just regenerate data *on the fly*. Due to fluctuating resource operating-points, it is essentially not possible to make such decisions upfront; currently it is also very difficult to implement these *dynamic decisions* in a general-purpose and scalable fashion.

Although the increasing volumes and complexity of data will make many problems data-load dominated, the computational-requirements will still be high. In practise, data-intensive applications will encompass data-driven applications — which are the basis for much of the *Fourth Paradigm* [2] of science. For example, many data-driven applications will involve computational activities triggered as a consequence of independent data creation; thus it is imperative for an application to be able to respond to unplanned changes in data load or content. Therefore on-balance understanding how to support dynamic computations,is a fundamental, but currently a critical missing element in data-intensive computing.

Our efforts will therefore operate at the triple point of *dynamic* and *distributed* and *data-intensive*, (3D) attributes. We refer to the first attribute as the "Dynamic-Data" problem; the last aspect is referred to as the "Big -Data" problem. We will consider the distributed aspects where applicable. More often than not, we will focus on applications that represent the merger of the *Big-Data* problem **AND** with the need for supporting *Dynamic-Data* **AND** which may either be fundamentally distributed or need to be distributed.

1.1 Establishing the Fundamental Role of Dynamic-Data

We outline three simple, yet representative 3D applications that will help to highlight the fundamental role of Dynamic-Data in data-intensive computing: Before that we briefly characterise Dynamic-data as possessing one or more of the following elements:

- Real-time: data that is processed in real-time, or redistributed or partitioned in real-time
- On-demand: data that is (made) available on-demand and thus needs processing when available
- Adaptive: granularity of storage or processing is adapted, data that is either passed through different filters etc.

Application 1: Sensor Data-driven Computation: For large-scale applications such as LEAD[1] the data-stream from the sensors drives the computational execution. Often in response to a predicted, or phenomenologically interesting event, the data-source and stream itself needs to be adapted, e.g., sampling rates (step-up/down), changing resolution, etc. Additional elements of dynamic adaptation – compute and data, arise from spatio-temporal variations in data generation (sensors).

[1] https://portal.leadproject.org/

Application 2: Dynamical Source and Analysis: A very nice example of dynamic distributed data-intensive applications is provided by *ClimatePrediction.net* [20], which has two intrinsic computational models, one for generating the data, and one for analyzing the data. Associated with the former is the concept of *phases*, where a client run will return a usable piece of the result part way through the computation of its work unit. The use of distributed computing in the data generation phase allows a collaborative processing, where the ClimatePrediction.net team can run a large set of models without having to own all of the needed computing and storage.

The model for the analysis of the data is less well known. Here there are a variable number of highly distributed computing resources and data stores. The data is distributed across the data stores in a regular way, with complete results from individual models always residing on the same server. The data is typically too big to transfer to a single location, so analysis must be performed *in situ*. It is necessary to provide some form of abstraction from the changing number of distributed resources. This was provided through a data parallel workflow language, Martlet, that contains a set of abstract constructs with which to build analysis functions. Other solutions, such as Parallel Haskell, are able to handle the changing numbers of computational resources, but a unique feature of this model of computing is that the *number of data sources is also changing*. There is definitely a gap here to be filled by further abstractions, as the current constructs (including Martlet) are just simple prototypes and their is a need for extensions with more powerful and useful ones.

Application 3: In-transit Adaptation: In many dynamic applications processing of data often needs to take place "in-flight" (e.g, via streaming) to meet boundary conditions, and/or data-volume reduction may be required to effectively store/manage data. An application example is provided by the coupled-fusion simulations that aim to provide integrated predictive plasma edge simulation to support next-generation plasma experiments, such as International Thermonuclear Experimental Reactor (ITER). Here data has to be transformed while it is being streamed using a mesh interpolation module to satisfy the different formulations – domain configurations and decompositions. Similar to the LEAD example, depending upon the specific components being connected (which is time dependent), the in-flight processing has to change.

The first application makes the case for agile execution and control of computing and data; the third reinforces the need to couple dynamic activity to data streams. As 3D applications become pervasive, the importance of dynamic placement, management and scheduling of data and data-sources/sinks will increase, as illustrated by all three examples. But there are several limitations in the current understanding and handling of Dynamic-Data, some of which carry-over from constraints in the way we handle the Big-Data challenge. For example, to address the challenge of Big-Data, several programming models, such as MapReduce & variants, have been developed. And even though the solution-space for Big-Data is not complete (See §1.2), most existing programming models (and associated tools and services) typically assume that the underlying data-set is "static", i.e., work-load assigned to a worker does not change during execution. Thus, performance, deployment and execution decisions, once made, are typically assumed to be valid throughout the life-cycle/execution of the application. This situation is analogous and reminiscent of the first-generation of distributed applications that inherited the static execution models of legacy cluster applications. It is only with the right tools, abstractions and run-time support that a subsequent class of distributed applications have been able to break-free of the static (resource) usage model. For traditional distributed applications, the ability for dynamic resource utilisation and optimisation has lead to a concomitant performance enhancement. *Thus, any vision and plan for managing a data-intensive future should include a strategy for supporting such dynamic and distributed aspects of data-intensive computing.*

1.2 Programming Paradigms for Data-Intensive Science

The different approaches to programming simulations are well understood although there is still much progress to be made in developing powerful high level languages. Today OpenMP [**?**] and MPI [**?**] dominate the runtime used in large scale simulations and the programming is typically performed at the same level in spite of intense research on sophisticated compilers. One also uses workflow to integrate multiple simulations and data sources together. This coarse grain programming level usually involves distributed systems with much research over last ten years on the appropriate protocols and runtime.

We can ask what the analogous programming paradigms and runtime are for data intensive applications?

We already know that many of the distributed system ideas will carry over as workflow has typically used dataflow concepts and integrated data and simulations. However as data processing becomes a larger part of the whole problem either in terms of data size or data-mining/processing/analytics, we can anticipate new paradigms becoming important. For example most data analytics involves (full matrix) linear algebra or graph algorithms, and not the particle dynamics and partial differential equation solvers characteristics of much supercomputer use. Further storage and access to the data naturally involves database and distributed file systems as an integral part of the problem. It has also been found that much data processing is less closely coupled than traditional simulations and is often suitable for dataflow runtime and specification by functional languages. However we lack an authoritative analysis of data intensive applications in terms of issues like ease and effeciveness of programming, performance (real-time latency, CPU use), fault tolerance, and ease of implementation on dynamic distributed resources.

A lot of progress has been made with the MapReduce framework originally developed for information retrieval – a really enormous data intensive application. Many data analysis applications, including information retrieval, fit the MapReduce paradigm. In LHC or similar accelerator data, map functions consist of Monte Carlo generation or analysis of events while the reduction function is the construction of histograms by merging distributions from different maps. In the SAR[2] data analysis of ice sheet observations, map functions consist of independent Matlab invocations on different data samples. Life Sciences have many natural candidates for MapReduce including sequence assembly and the use of BLAST and similar programs. On the other hand, partial differential equation solvers, particle dynamics and linear algebra require the full MPI model for high performance parallel implementation.

In general, MapReduce programming models offer better fault tolerance and dynamic flexibility than MPI and so should be used in loose coupling problems in preference to MPI. The MapReduce programming model implies straightforward and efficient fault tolerance by re-running failed map or reduce tasks. MPI addresses a more complicated problem architecture with iterative compute–communicate stages with synchronisation at the communication phase. This synchronisation means for example that all processes wait if one is delayed or failed. This inefficiency is not present in MapReduce where resources are released when individual map or reduce tasks complete. MPI of course supports general (built-in and user-defined) reductions so MPI could be used for applications of the MapReduce style. In addition to greater fault tolerance, the latter offers a user-friendly, higher-level environment largely stemming from the coarse grain functional programming model implemented as side-effect free tasks. It is evident from the discussion that appropriate programming paradigms for data-intensive science have to be considered, amongst other things, in the context of the degree-of-distributedness and cyber-infrastructure – software implementations, physical infrastructure of the problem at hand.

1.2.1 Axes of Research Challenges for Programming Paradigms

Using MapReduce as an exmplar, we highlight representative challenges along the algorithmic, software and cyberinfrastructure axes associated with programming paradigms and data-intensive applications.

Algorithmic: There is a clear algorithmic challenge to design more loosely coupled algorithms that are compatible with the map followed by reduce model of MapReduce. This could lead to generalisations of MapReduce which are still compatible with the cloud virtualisation and provide fault tolerance. Research into extensions of MapReduce attempt to bridge these differences.

Software: There are many software challenges including MapReduce itself; its extensions (both in functionality and higher-level abstractions); and improved workflow systems supporting MapReduce and the linking of clients, clouds and MPI engines. We note there is also active work in the preparation, management and deployment of program images (appliances) to be loaded into virtual machines.

Infrastructure: Another critical area is file systems where clouds and MapReduce use new approaches that are not clearly compatible with traditional production Grid infrastructures [13, 14]. Support of novel databases such as Big Table across clouds and MPI clusters is probably important. The intrinsic conflict between virtualisation and the issues around locality or affinity (between nodes in MPI or between computation and data) needs more research.

[2] http://www.asf.alaska.edu/sardatacenter

1.2.2 Research areas for Scientific Computing with MapReduce and Clouds

To appreciate the need to understand the landscape of programming paradigm — abstractions, software implementations, run-time and middleware services and the overall cyber-infrastructure, we discuss the the example of MapReduce and its use/support in clouds. There are multiple reasons to believe that clouds will form an increasingly important component in the cyber-infrastructure used for data-intensive. This is in spite of well known deficiencies in the current cloud paradigm and offerings, namely,

1. the centralised computing model for clouds runs counter to the concept of "bringing the computing to the data" and bringing the "data to a commercial cloud facility" may be slow and expensive,

2. the virtualised networking currently used in the virtual machines in today's commercial clouds and jitter from complex operating system functions increases synchronisation/communication costs.

Turning point 1 around on its head, we can say the "centralised" model of clouds just makes it more important to determine which data and when, should be centralised (and possibly static) versus the explicit support for distributed (and possibly dynamic) data. Point 2 is especially serious in large-scale parallel computing; indeed the usual (and attractive) fault tolerance model for clouds runs counter to the tight synchronisation needed in most MPI applications. There are other barriers to the whole-sale adoption of clouds, such as issues raised by security, legal and privacy, but we will not discuss them here as they are not typically determinants of performance for scientific data-intensive applications.

A more careful analysis of clouds versus traditional environments and other elements of the cyber-infrastructure to support a programming paradigm is needed to quantify the simplistic analysis given above. Also, any comprehensive analysis of programming paradigms for data-intensive science, must include MapReduce, but cannot stay confined to MapReduce.

1.3 Background Motivation, Intellectual Aims and Impact

As the report [1] from the recent Workshop on Data-intensive Research held at the e-Science Institute suggests, there exists a serious shortage of well understood and broadly applicable programming paradigms [18] for data-intensive applications. There exist some well known programming paradigms, such as MapReduce, and there has been a rush to reformulate many applications using MapReduce, whilst at the same time extending MapReduce to support new capabilities (e.g., iterative MapReduce). But in general, there was agreement on the lack of general-purpose programming paradigms for data-intensive application and sufficient understanding of the challenges thereof (See §5.5 of Ref.[1]). Along with the cross-cutting issue of programming paradigms for the different types & categories of 3D applications, we will also look at issues of abstractions and run-time systems for 3D applications.

We have observed important trends towards large-scale, dynamic and distributed applications, and identified limitations that make it important to understand the landscape of dynamic, distributed, data-intensive (3D) applications. This forms the bedrock of the intellectual aims and impact of the proposed 3DPAS theme, which can be captured in the following two statements:

- *Enhance our Understanding of the Landscape of Large-Scale Distributed Data-Intensive Applications:* We will analyse a broad range of data-intensive application, based upon which we will define a set of fundamental properties ("vectors"), which can be used to categorise distributed data-intensive applications in general. Having identified the "vectors", we will analyse existing & required approaches for the development of these applications. These will include, but not be limited to programming paradigms and abstractions, run-time execution and middleware services (REMS), as well as the cyber-infrastructure to support their effective deployment & execution.

- *Extending our Understanding to Large-Scale Dynamic Distributed Data-intensive (3D) Applications:* We will extend the understanding gained with distributed data-intensive applications to address the lack of our structured understanding of 3D applications. In some cases, the challenges of increasing scales – data volume, data sources, data users – and complexity of data, resources and analysis that define the landscape for *static* data, are simply magnified for *dynamic* and distributed data. For other cases, a fresh perspective is required. For example, abstractions and programming models are required to support the formulation of components and applications that are capable of correctly and consistently adapting their behaviors, interactions and compositions in real time in response to dynamic data. Which programming models are required, and how should they be offered?

Impact: The short-term impact will be represented by the publications and engagement with the e-Science Community (e.g., Software Sustainability Institute), possibly identifying best-practises and new usable abstractions (see "Plausible Outcomes"). The long-term impact, is more difficult to predict however, but based upon our experience with the DPA theme, we believe our work can become the authoritative reference for the status and survey of the field; thus we believe our long-term impact will be both pedagogical and research. Also, similar to the DPA theme we anticipate that our work will help bridge the gap between application development and infrastructure providers and developers [8].

2 Theme Objectives

The aim of the proposed 3DPAS Theme to understand the landscape – as defined by the *programming models and abstractions, run-time and middleware services and the computational infrastructure* – of dynamic distributed data-intensive (3D) applications. Understanding the landscape will serve as the critical first step in the gap analysis at each level. Providing the basis for our detailed analysis will be a set of real (read, in-production) applications and systems; epresentative 3D applications were discussed in §1; this will be vastly extended. We will validate our gap analysis with proposed solutions and abstractions that we will aim to demonstrate in publications inspired by and concurrent with the theme.

2.1 Specific Aims of the Theme

We observe that this is a diverse and active research area. However our efforts will be focused on attempting to understand the issues outlined in Big-Data and Dynamic-Data and the cross-cutting issues that bridge Big-Data and Dynamic-Data applications (and supporting infrastructure). Although we will attempt to consider the views of the wider research community working in this area, in order to focus the theme, we will attempt to consider possible answers to the following questions (and use these as a means to structure the interactions during the workshops):

1. What are the common, minimally-complete, characteristics of distributed data-intensive application? What are the representative application examples?

2. How are these applications currently developed, deployed and executed? What are the major barriers and scalability challenges?

3. What patterns exist within such applications, and are there commonalities in the way such patterns are used? In other words what abstractions exist or can be established to support the development, deployment and execution of distributed data-intensive applications?

4. What are the representative data-intensive applications (along with their characteristics) that can be used to define a benchmark suite of data-intensive applications?

5. Provide a Critical Perspective and a Gap Analysis of the programming abstractions and systems, tools and infrastructure that exist for applications that need to handle Dynamic Data and/or Big Data.

6. How can programming models, abstraction and systems for Big-Data applications be extended to support Dynamic-Data applications? What are the common programming models to support data-intensive applications?

7. What tools, environments and programming support exist to enable emerging distributed infrastructure to support the requirements of dynamic applications – including but not limited to streaming data and in-transit data analysis? Do these tools already exist, and to what extent are they being used at present?

Q1-3 outlined above focus on an understanding of the Big-Data problem and the cross-cutting issues associated with Big-Data challenge. Q7 is devoted to the Dynamic-Data problem. Q4-6 provide the connective tissue and common issues between the two primary sub-themes. We will revisit these question in §5 (Schedule and Workplan).

2.2 What are the plausible outcomes from the theme? Will they be entirely theoretical? Will there be some experiments/software produced?

The deliverables from the 3DPAS theme will be an interesting mix of theoretical work (in the form of research papers), working group reports (as an eSI Technical report) and a practical contribution.

Scope of Planned Publications:

At a theoretical level, we will publish an eSI Technical Report (TR) that presents and analyses the following high-level questions:

- Explore and analyse the landscape of 3D applications. Understand the current state of development, deployment and execution of such applications.
- Identify the design objectives of 3D applications? In the DPA theme we established IDEAS – **I**interoperability, **D**istributed Scale-Out, **E**xtensibilty, **A**daptivity and **S**implicity as design objectives for distributed applications.
- Identify possibly "new" abstractions and missing solutions: For example, the DPA analysis identified the ability to support (a range of) compute-data affinities for distributed data-intensive applications via the "Pilot-Data" abstraction as an important abstraction to support application requirements. What are the analogous cross-cutting design objectives for which general-purpose abstractions must be found?
- Analyse the supporting cyber-infrastructure for 3D applications. What tools exists? What programming systems are available? Understand effectiveness of programming paradigms for 3D applications. What are the current capabilities and what are the missing pieces?

Where relevant we will highlight missing gaps, recommend solutions and identify best-practises. The publication of a gap analysis will set-up potential research challenges and future directions. We will convert a condensed version of the TR into a peer-reviewed publication (see milestones section).

Scope of Practical Impact:

Interestingly, the 3DPAS "theoretical treatise" will inform the practical components too. The 3DPAs will have immediate and direct practical impact at least two different ways, as outlined below:

- The understanding gained from the mapping of the "application types" to "programming systems/tools and infrastructure" will inform the engagement component of the SSI. More precisely, the SSI is commited to providing and advising practical solutions for a broad range of scientific application. As no such mapping exists, the outcome of the data-intensive programming paradigm focus-study will be an asset.
- It is generally believed that the easy part of distributed computing was distributing the computing. This has been an important area of focus and advancement for SAGA in its initial formulation. But the impact of data-intensive science cannot be ignored in the evolution and development of SAGA. Hence, the understanding gained from broad-survey and in-depth analysis of application characteristics as well as requirements imposed on the programming systems & tools will motivate and guide the enhancement of SAGA towards a comprehensive programming system for distributed data-intensive applications.

3 Are there other similar projects to the proposed theme? What would be their relationship to/involvement in this programme?

3.1 3DPAS Synergy With Proposed DIR Theme

We are aware of theme proposal (TL: Atkinson, Szalay, Martin, van Hamert) that is being submitted for consideration as well. The 3DPAS theme, if funded, will be intellectually coupled by subject-matter, and co-operative in execution, but will however be independent in leadership, deliverables and finances. For example, the 3DPAS TL will work closely with co-TLs of the proposed DIR theme to refine scope & challenges in the broad data-intensive domain, possibly coordinate visitors and co-locate workshops where meaningful. The concurrent running of two data-oriented themes, with different approaches and core communities[3], but a similar objectives ("How to Improve our Use of Data") at heart, is an unique opportunity – not just for intellectual impact (measured by papers and reports), but also to possibly bring otherwise non-overlapping communities together.

The TL (Jha) and co-TL of the DIR theme (Atkinson) will try to attend workshops in the other team, and will also meet (physically or virtually) at least once per month to coordinate theme activities.

TL Jha will examine working with co-TL Szalay to propose funding to the US-NSF (see § on Leveraging other Financial Sources).

3.2 3DPAS – A Natural Evolution of the Ideas and Insight Gained from DPA

The DPA theme has explored and made important contributions in at least two domains: (i) Abstractions for Distributed Applications, and (ii) Autonomic Computational Science Applications. For example, in the former, as part of the DPA Theme Book, "Abstractions for Distributed Applications and Systems", we have developed on a chapter titled, "Abstractions for Data-Intensive Application and Systems", wherein we rigorously analysed multiple applications and systems and identified *affinity* as an important cross-cutting concern. There are many types of affinity, such as data-data affinity, data-compute affinity etc. We also identified Pilot-Data as a useful abstraction to support many of these affinities.

The DPA theme (See References [4]-[12] for publications engendered by the eSI DPA theme) provided a critical perspective of the landscape of distributed applications. It defined a representative set of approximately dozen production-grade applications, to motivate a broad but relevant methodology for understanding distributed applications and production-grid infrastructure. Specifically, we charted the landscape in the context of programming systems, tools and infrastructure support that exist for the development, deployment and execution of distributed applications. The methodology we propose to employ here, will be qualitatively similar: we will use a large number of applications as the basis for understanding the tools and programming systems, as well as the data-driven dynamic applications through their life-cycle, and thereby perform a gap analysis.

In the Autonomic Computational Science track of the DPA theme, we have identified two types of frameworks to support Autonomic Capabilities: frameworks that support the tuning of applications, and those that support the tuning by applications. The mapping of autonomic characteristics to either type is not "a closed, well-defined problem" but is dependent upon the application objectives, usage modes and the infrastructure available. Our analysis has focused on real-world applications (such as Montage), where autonomics can have beneficial performance (and other) advantages. It is important to extend and generalise this analysis to dynamic and data-driven applications. We are leading and editing a special issue on "Autonomic Computational Science", to be published by *Scientific Computing* (Editor Ron Perrot).

One implicit aim of the 3DPAS Theme is to bring the two DPA strands together and focus on the research issues that emanate from their convergence. There is a need to understand the "autonomics associated with data": where to place data, and how to change the placement, generation and transport of data. Although we will investigate autonomic approaches in detail, it will not be at the exclusion of other methodologies, and we will be analysing different approaches to the overall problems. For example, traditional ways to couple application components (e.g., database, file-repositories) need to be re-evaluated; and there are a significant and growing number of applications and data-integration schemes that will benefit from decomposition into computational components coupled via *data streams*.

[3] There are several common core members – Dobson, Chue-Hong, Rana, Blower etc.

4 Theme Leader, Working Group and Community

Building on the impact of DPA, the we are emboldened to propose the 3DPAS Theme. The core team will provide intellectual continuity will providing the opportunity for new skills and expertise to influence our thinking.

- Theme Leader: Shantenu Jha (Louisiana State University, USA & Visiting Fellow (2006-2012) University College London, UK). Shantenu is an Assistant Research Professor in Computer Science, and the Director of Cyberinfrastructure Development, at the Center for Computation & Technology (CCT) at Louisiana State University (LSU). Shantenu has a long term appointment at UCL, where amongst other research projects, he currently co-supervises one PhD student with Peter Coveney, and partly funds (via NSF grants) two post-doctoral level researchers in Coveney's group. Shantenu's interests are in Computational Science and High-performance and Distributed Computing, with an evolving emphasis on data-intensive computing. He is the lead of the SAGA project which is responsible for API specification & implementation, developing distributed applications and deployment over different infrastructure. He has won several international awards for developing and deploying novel distributed applications on distributed cyberinfrastructure. He serves on the User & Advisory Board for the $15M NSF-funded FutureGrid project which will provide a test-bed for implementing practical ideas generated from the proposed theme. Shantenu has been involved with the UK e-Science program since early 2002, has been an active participant in every AHM since, & been a PC member of the AHM since 2007. For further details and select publications, please see: http://www.cct.lsu.edu/~sjha & http://www.cct.lsu.edu/~sjha/select_publications

4.1 Working Group

Researchers and practitioners who have committed to staying engaged with the theme and contributing to the intellectual aims are listed below. The list will be strategically increased to include an expanded set of researchers so as to provide a balanced and broad set of interests, skills and expertise and thus to constitute a successful core working group.

- Prof. Omer Rana (Cardiff): Omer will be a co-lead of the working group. Omer is a Professor of Performance Engineering in the School of Computer Science at Cardiff University, and the Deputy Director of the Welsh eScience Centre. He holds a PhD in Computing from Imperial College, London, and works in the areas of high performance distributed computing, multi-agent systems and Data Mining. He has also recently been involved with the computational science in the development of scientific applications which can be deployed over Computational Grids. He co-chaired the "Service Management Frameworks" research group at the Global Grid Forum (2002-2005), and previously co-led the "Jini" working group at the GGF. Currently, he participates in the GRAAP and Semantic Grid groups. He participates on the Editorial boards of the "Concurrency and Computation: Practice and Experience" (Wiley), the "Scientific Programming" (IOS Press) and the "ACM Transactions on Autonomous and Adaptive Systems" (ACM Press) journals.

- Prof. Malcolm Atkinson (e-Science Institute, Edinburgh) Interests: As a long-term researcher into large and long-lived computing systems and their driving applications I seek to understand the evolving requirements for data-intensive computations and to deliver architectures and methods that facilitate the creation and use of those systems. I expect to influence future systems through bodies such as the UK e-Infrastructure Reflection Group and SSI, as well as through academic research and industrial liaison.

- Prof. Jim Austin, (York University and Cybula Ltd. http://www-users.cs.york.ac.uk/austin/). Jim is a Prof. of Neural Computing, and leads the Advanced Computer Architecture Group.

- Prof. Margaret Bell CBE, Newcastle University. http://www.ceg.ncl.ac.uk/profiles2/margaret.bell Professor Bell was named Commander of the Order of the British Empire for services to sustainable transport in the Queen's 80th Birthday Honours List in 2006. Her main contribution to knowledge rests with queueing model for TRANSYT, ageing of traffic signal plans and pioneering research in assessment of impacts and management of traffic related environment, exposure and health. She has been instrumental in creating integrated research facilities including the instrumented City which boasts two decades of historic data and more recently with Professor Blythe initiated the North East Transport Observatory which forms the platform for assessing the performance of electric vehicles and delivers the modelling framework for investigating Regional Transport sustainabilty. Professor Bell is a member of the Research Council's Peer Review Colleges, Chair of the Smart Environment Interest Group of the ITS (UK) and given evidence at the Transport Select Committee on behalf of the IET. Other areas of research include: development of congestion measures, forecasting air quality and duration of incidents with neural networks and fuzzy logic, evaluation of demand management strategies and traffic control policy on the environment. In-depth studies of the effect of driver behaviour on tailpipe emissions, evaluation of the impact of fatal accidents on network delay, of illegal parking on vehicle emissions,the role of ITS in reducing carbon emissions, application of hazard models to understand changes in trip duration, logit modelling for assessment of the quality drivers of bus service satisfaction and development of decsion support system for public transport interchange.

- Dr. Jon Blower (Reading). Jon is the Technical Director of the Reading e-Science Centre (ReSC). His main interests lie in the application of e-Science methods to the environmental sciences.

- Dr Mario Cocommo (TGAC, Norwich): Mario is the Head of Bioinformatics, at The Genome Analysis Centre, Norwich Research Park. Amongst many other elements of Bioinformatics, his interests are in programmatic access and APIs for bioinformatic analysis and programming.

- Dr Neil Chue-Hong (Edinburgh). Neil is Director of the Software Sustainability Institute, a national facility for research software users and developers providing specialist software engineering skills to drive the continued improvement and impact of research software. Previously, he has been Director of OMII-UK, Technical Manager of NeISS, Project Manager of ENGAGE and Project Manager of OGSA-DAI. Neil in his role as the SSI director will bring a specific interest and contribution to this theme, viz., the opportunity to guide the theme towards tangible practical outputs so as to benefit the mission of the SSI. For example, the ability to advise end-users when they approach him with questions like, " Is there a classification of data-intensive applications that would help me to understand (simply) if Approach X were/were-not useful for my problem Y?"

- Prof. Simon Dobson (St. Andrews). Simon is a Professor of Computer Science at St. Andrews. His current research focuses on sensor networks and environmental sensing. These systems are notoriously hard to develop and analyse since they have to adapt to changes in their environment while continuing to deliver data and services reliably. Simon's work involves building formal mathematical models of adaptive sensor systems, and developing new ways of programming them to improve our ability to do sophisticated long-term experiments. His interest in 3DPAs will centre around programming and program analysis. Simon will help organise events, especially in programming paradigms for sensor networks.

- Prof. Keith Haines (Reading)

- Dr. Milo Thurston (Oxford), Milo is the Scientific Computing Specialist for Climateprediction.net. An important aspect of this role is cataloguing the large volumes of data returned from Climateprediction.net participants and making it available to climate researchers. His scientific background is in biology, bioinformatics and Linux system administration; he has a degree in microbiology and genetics from Dundee University, a D.Phil. in virology from Oxford University.

- Prof. Paul Watson (Newcastle)

Industry Affiliate and eSI Visitors:

- Dr. Roger Barga (Microsoft): Roger is currently principal architect for Technical Computing, a Microsoft initiative to partner with the scientific community to accelerate advances in science through computing. The Technical Computing team works across Microsoft to coordinate collaborative efforts with the scientific community worldwide. Previously, Dr. Barga held the position of researcher in the database group of Microsoft Research from 1997 through 2006.

- We will also explore the need/possibility of requesting long-term eSI visitors status for a core set of working group members. Our working group will be drawn from the set of science -specialists and other target groups outlined above. Prof. Parashar (NSF & Rutgers), Katz (UC) and Weissman (Minnesota) – members of the DPA Core Group, have also expressed interest and agreed to contribute and remain intellectually engaged.

4.2 Target Communities/Areas that we will engage and representative specific participants

We list some science-domains and groups that we will seek to engage directly (e.g., seek input from at the workshops) as well as representative communities that will benefit directly from our proposed extension.

- Autonomic Modelling of Traffic: Haibo Chen (Leeds)
- Autonomic Systems Science: Cecile Germain-Renaud (Paris)
- Advanced Programming Paradigms for Bioinformatics: Geoffrey Fox (Indiana)
- Biodiversity Informatics: Prof. Alex Hardisty (Cardiff University)
- Biological Sciences and Systems Biology: Prof. Pedro Mendes (Manchester)
- Climate Change: Prof. Martin Dove (National Institute for Environmental eScience, Cambridge)
- Medical Imaging: Prof. Richard McClatchey (UWE, Bristol), & neuGrid project (Dr Johan Montagnant, INRIA)
- Health Informatics – as part of EPSRC's "Grand Challenges in Information Driven Healthcare" programme, Richard Clayton and Rod Hose (Sheffield University)
- Earth Sciences and Programming Models & Systems for Dynamic applications – Roger Barga (Microsoft)
- Social Sciences: Mark Birkin (Leeds) and Dave de Roure (Southampton)
- Sensors-based Simulations and Cloud Computing Applications: Paul Watson (Newcastle)

4.3 Relation to e-Science

We have discussed some representative examples of applications in §1. To see a description of the application areas that would benefit, please see §4.1 and §4.2. However, the engagement of the 3DPAS theme, as well as its impact, will not remain confined to science domain. Reflecting the integrative (and practical) nature of the problems being addressed, the theme will cover the landscape — tools, systems and programming models/abstractions, required for distributed data-intensive applications. By virtue of our analysis, we will be able to directly address important over-arching questions, including but not limited to: What are the important classes and characteristics of 3D applications? What are the barriers to the effective development, deployment & execution of such applications. What are the major impediments to scalability?

Our work will thus also be of interest to the following groups, and similar to the DPA theme, will help bridge the gap between infrastructure (provider and developers) with the above mentioned application (classes):

- Tool and Infrastructure developers: We seek to discover what abstractions the tools to develop and support 3D applications should provide, and how frameworks to support the development, deployment and execution phases of such application can be designed and constructed. How should experimental and production infrastructure be designed to support data-driven dynamic instantiation, configuration and aggregation of distributed resources and varied data sources? In the same context, it is necessary to analyse the support infrastructure for 3D applications. What are the current capabilities and what are the missing pieces?

- Infrastructure and resource providers (such as production grid infrastructure provider and specialized science and data cloud providers): 3D applications are an important class of applications in domains from health informatics to climate, environment and Earth sciences, to a broad range of biological science problems, and especially as they move towards a "cloud-based" analytics and infrastructure. Clouds are as a consequence of their (i) the control over the software environment provided to the end-user, (ii) ability to provide illusions of immediate and unlimited resource access, important infrastructure for supporting novel formulations of dynamical applications [17]. For example, applications have to be reformulated to be 'dynamic', but so do many elements of emerging infrastructure need to support programming systems and services required for dynamic applications. The output from 3DPAS will help them address issues such as what capabilities and programming models should they support, as well as the interfaces they should expose in support. In general, what are the cross-cutting design objectives for which general-purpose abstractions must be found? How can existing Runtime Execution and Middleware Services (REMS) be extended to support these design objectives? How can REMS be developed and integrated with applications to preserve performance and resilience, yet not be tied to a specific infrastructure?

- Contribution from members of previous themes funded at the eSI will also be investigated, this include, in particular the Geo-Informatics theme on "Geographic Information Processes". The particular focus within this theme was on spatial semantics and data interoperability, however, this area also provides significant challenges for managing: (i) sensor data; (ii) dealing with large data sets.

5 Project Plan, Milestones and Resources Requested

Proposed Schedule and Workplan

- Duration: 01 June 2010 (start) - 31st August 2011 (end); (15 Months)

- We plan a total of 3 Workshops [September 2010, Spring 2011, May 2011] and 5 Working Group Meetings [quarterly]. A meeting of 3DPAS WG members that are present at the UK e-Science All-Hands (Cardiff; mid-Sep) will also be arranged.

- In general the WG meetings will be aimed at focused discussion, study and writing time to make progress — both in the shadow of, and the run-up to the larger Workshops.

- Q1-3 outlined in §2.1 will be the focus of Workshop I; these focus on an understanding of the Big-Data problem and the cross-cutting issues associated with Big-Data challenge. Q6-7 are dominated by the Dynamic-Data problem, and will be the focus of the Workshop II. Q5 and Q6 from §2.1 will be a common thread/overlap between both workshops. The final 3DPAS Workshop, as well as the final working group meeting will be aimed at compiling and synthesising a final report from workshops. This will be published as an eSI Technical Report The TR will be a live document and will evolve over the course of the 3DPAS theme.

Milestones

- July 15, 2010: Constitute Working Group. The TL will draw from the committed members and target communities to form a 3DPAS working group, akin to the DPA Authors/Core Group. However, this will be formed more rapidly and will be broader in expertise and interest.

- Nov 15, 2010: First draft of the 3DPAS eSI TR, with a focus on Programming Paradigms and Infrastructure for Big-Data

- April 15, 2011: Second draft of the 3DPAS eSI TR, with a focus on Programming Paradigms and Infrastructure for Dynamic-Data.

- July 31, 2011: At the last meeting of the 3DPAs WG, we will have a final version of the TR spanning Big-Data, Dynamic-Data and the cross-cutting issues associated with 3D applications.
- Aug 31, 2011: Convert eSI TR into a Journal Review/Survey article. Possible candidates include Scientific Programming, ACM Series and/or a special issue of Concurrency and Computation: Practise and Experience (eScience-A). Closing 3DPAS public-lecuture.

References

[1] Malcolm Atkinson, David De Roure, Jano van Hemert, Shantenu Jha, Bob Mann, Stratis Viglas, and Chris Williams. Data-Intensive Research Workshop Report. Technical report, e-Science Institute, 2010. http://bit.ly/cfMRn3

[2] The Fourth Paradigm: Data-Intensive Scientific Discovery http://research.microsoft.com/en-us/collaboration/fourthparadigm/default.aspx

[3] NSF DDDAS Workshop (2006) www.nsf.gov/cise/cns/dddas/2006_Workshop/wkshp_report.pdf

[4] L. Dagum, R. Menon, and S.G. Inc. OpenMP: an industry standard API for shared-memory programming. *IEEE Computational Science & Engineering*, 5(1):46–55, 1998.

[5] M. Snir and S. Otto. *MPI-The Complete Reference: The MPI Core*. MIT Press Cambridge, MA, USA, 1998.

[6] Abstractions for Distributed Applications and Systems, submitted to ACM Computing Surveys (under review) http://www.cct.lsu.edu/~sjha/dpa_publications/dpa_surveypaper.pdf

[7] Self-Adaptive Architectures for Autonomic Computational Science, to appear in post-proceedings publications of Workshop on Self-Organizing Architectures (SOAR) 2009, volume of LNCS Series (2010) http://distrinet.cs.kuleuven.be/events/soar/2009/contents/papers/rana-soar09.pdf http://www.cct.lsu.edu/~sjha/dpa_publications/dpa-soar09.pdf

[8] Fresh Perspective on Developing and Executing DAG-Based Distributed Applications: A Case-Study of SAGA-based Montage, accepted for IEEE Conference on eScience 2009, Oxford http://www.cct.lsu.edu/~sjha/dpa_publications/saga_montage_G09.pdf

[9] Autonomic Approach to Integrated HPC Grid and Cloud Usage, accepted for IEEE Conference on eScience 2009, Oxford http://www.cct.lsu.edu/~sjha/dpa_publications/cloud-grid-autonomics.pdf

[10] Critical Perspectives on Large-Scale Distributed Applications and Production Grids (Best Paper Award), Proceedings of the 10th IEEE/ACM International Conference on Grid Computing (GRID09), 2009 http://www.cct.lsu.edu/~sjha/dpa_publications/dpa_grid2009.pdf

[11] Investigating Autonomic Behaviours in Grid-Based Computational Science Applications, In Proceedings of the IEEE 6th ICAC, Workshop on Grids Meets Autonomic Computing (Barcelona, Spain, June 15 - 15, 2009) http://www.cct.lsu.edu/~sjha/dpa_publications/dpa-autonomics-gmac.pdf

[12] Developing Autonomic Distributed Scientific Applications: A Case Study From History Matching Using Ensemble Kalman-Filters, In Proceedings of the IEEE 6th ICAC, Workshop on Grids Meets Autonomic Computing (Barcelona, Spain, June 15 - 15, 2009) http://www.cct.lsu.edu/~sjha/dpa_publications/lazarus_gmac09.pdf

[13] Programming Abstractions for Data Intensive Computing on Clouds and Grids, pp. 478-483, 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, 2009 http://www.cct.lsu.edu/~sjha/dpa_publications/saga_data_intensive.pdf

[14] Developing Scientific Applications With Loosely-Coupled Sub-Tasks, International Conference on Computational Science (ICCS) 2009, Vol. 5544Springer (2009), p. 641-650 http://www.cct.lsu.edu/~sjha/dpa_publications/saga_loosely_coupled.pdf

[15] S Sehgal, A Merzky, M Erdelyim and S Jha*, "Understanding Application Level Interoperability: Scaling-Out MapReduce From High-Performance Grids to Web-Scale Resources", *short-listed for* Future Generation Computing Systems, as special issue of selected papers from Grid and Pervasive Computing 2009. http://www.cct.lsu.edu/~sjha/select_publications/2010_fgcs_mapreduce.pdf

[16] C Miceli, M Miceli and S Jha, "Understanding Performance of Distributed Data-Intensive Applications", Accepted for Philosophical Transactions of the Royal Society of London A (2010) http://www.cct.lsu.edu/~sjha/select_publications/saga_ptrsa10.pdf

[17] Exploring Adaptation to Support Dynamic Applications on Hybrid Grids-Clouds Infrastructure, Accepted for ACM ScienceCloud, workshop held in conjunction with HPDC 2010 http://www.cct.lsu.edu/~sjha/dpa_publications/hybrid-autonomics-sciencecloud.pdf

[18] http://www.nsf.gov/pubs/2010/nsf10050/nsf10050.jsp?org=NSF http://www.nsf.gov/pubs/2010/nsf10050/nsf10050.jsp?org=NSF

[19] S Jha, D S Katz, A Luckow, A Merzky, K Stamou, Understanding Scientific Applications for Cloud Environments, submitted to book on Cloud Computing, Edited by Raj Kumar Buyya, to be published by Wiley http://cct.lsu.edu/~sjha/select_publications/cloud_book_chapter.pdf

[20] Progamming Paradigms Track of the DIR Workshop.
http://wiki.esi.ac.uk/Programming_Paradigms
http://www.nesc.ac.uk/action/esi/download.cfm?index=4437 &
http://www.nesc.ac.uk/action/esi/download.cfm?index=4478

[21] Dynamic Distributed Data-Intensive Track of the DIR Workshop.
http://wiki.esi.ac.uk/Dynamic_Distributed_Data-Intensive_Applications

[22] ClimatePrediction.Net – The World's Largest Climate Forecasting Experiment for the 21st Century http://climateprediction.net/