1 Significance of Research

A fundamental challenge for the chemical sciences is to bridge the gap between our ability to study and manipulate matter at the atomistic scale and our desire to understand, and predict how this impacts on the emergent properties of systems at a macroscopic scale. Molecular simulation holds the keys to understanding this relationship.

Our team is currently working on the (NSF-funded) development of ExTASY (Extended Tools for Advanced Sampling and analYsis), a novel software framework to provide a step-change in the sampling of the conformational space of macromolecular systems. ExTASY has been explicitly designed to play to the strengths of current and next-generation state-of-the-art High Performance Computing facilities. A developing set of flexible, interoperable tools provide high-level abstractions of common molecular simulation tasks and permit the rapid development and deployment of novel data- and compute-intensive workflows to address the most challenging issues in the prediction and interpretation of biomolecular structure, dynamics, recognition, and function.

Despite recent successes, in general, the achievement of the sampling necessary to predict macroscopic properties from Molecular Dynamics (MD) simulations of large systems remains an enormous problem. It has motivated huge investments in specialized hardware to improve sampling by "brute force" increase in simulation speed [1], and software solutions based on "divide and conquer" approaches [2] where single, long simulations are replaced by large numbers of shorter, independent, ones. Even for a stable well-folded protein, the adequate sampling of its conformational dynamics–a prerequisite for the accurate prediction of thermodynamic parameters–can be a serious challenge. The use of ensemble simulation methods, supported by novel middleware solutions and distributed computational resources, has been proved critical for achieving a useful time-to-solution for these problems.

Within the ExTASY project, we have recently showed that, by interleaving large ensembles of short molecular dynamics simulations with novel analysis tools to direct the sampling on-the-fly, we can speed up the sampling of macromolecular systems at least three orders of magnitudes [3]. Here we are seeking the computational resources to move beyond these proof-of-principle studies on model systems and fully exploit the advantages offered by our approach for the characterization of large macromolecular motions in the study of a system of major current interest. Specifically, we will focus on the characterization and understanding of the desensitization process in the *N*-methyl-D-aspartate receptor (NMDAr), a member of the glutamate receptor family of proteins.

This application requires simulations with timescales and system sizes well beyond any that have previously been undertaken. We believe that access to the computational resources of Blue Waters will provide us with the capabilities to study such a system in the ExTASY framework. Beside the importance of the NMDAr *per se* (as discussed in the next section), this study will open the way to a conceptually different approach to study of large conformational changes in complex macromolecules, as the same methodology can in principle be applied to a large number of biomedically relevant systems.

We plan to generalize the solutions that will be developed to support large ensembles of NMDAr simulations and make them available to the community.

The second track of this project is concerned with development, scaling and optimization of the cyberinfrastructure required to support the scientific objectives of this project. Specifically it will cover: (i) the optimization of the underlying ExTASY tools to enable ensemble runs at large-scales, (ii) improving the scaling of RADICAL-Cybertools [4] including the implementation of advanced workload management systems, and (iii) the development of SPIDAL (Scalable Parallel Interoperable Data Intensive Libraries) and MIDAS (Middleware for Data-intensive Analysis and Science). SPIDAL and MIDAS are the two primary components of the HPBDS [**?**] (high-performance big data stack) project, part of the recently awarded DIBBS project (ACI 1443054) which will aim to provide the libraries and middleware to support data-intensive analytics on existing and future NSF's high-performance platforms.

Our team combines the expertise of domain and computer sciences/cyberinfrastructure: Clementi and Laughton have an extensive track record in developing and using different tools to define collective variables for biomolecular systems from large data samplings, while Jha has been responsible for supporting many scalable applications, as well as both conceptual advances and the development & optimal deployment of extensible computational tools on leadership-class HPC resources. Fox is a world leader in parallel and data-intensive computing.

1.1 NMDA receptor's ligand binding domain

Glutamate receptors are biomedically important as they mediate cation transport, primarily calcium ions, in and out of the cellular membrane in neurotransmitters [5]. The NMDAr is unique among the glutamate receptors in that it forms hetero-tetrameric complexes using two glycine binding GluN1 monomers, and two other monomers, either the glutamate binding monomer GluN2 or the glycine-binding GluN3 monomer [6]. NMDArs are believed to be involved with a number of neurological diseases including schizophrenia, Parkinsons, and Alzheimers [7–9]. Each monomer of the NMDAr is comprised of four main domains, the extracellular N-terminal domain, the ligand binding domain (LBD), the intracellular C-terminal domain, and the transmembrane domain [6, 10]. The LBD is comprised of a top and bottom region connected by a hinge region forming a cleft in which the agonists bind (see Fig. 1 inset) [10]. In order to precisely control ion flow, the NMDAr, like the other glutamate receptors, performs its function via a cycle of concerted conformational changes. First, in the "activation" step, a ligand binds to the LBD and initiates a conformational change that opens the transmembrane channel. After a brief time, a second conformational change occurs that closes the channel in what is called the "desensitization" step, because it occurs even when the "activating" ligand remains bound. For normal function, the receptor needs to be able to respond quickly to the sudden presence of the signaling molecules, but somewhat paradoxically be allowed to relax back to a closed channel conformation even while still bound to the agonist which induced the open channel conformation in the first place, in order to avoid over-stimulation in the synaptic cleft [11]. The NMDAr desensitization pathway is known to be complex and still not well understood. We believe that understanding the desensitization mechanism will help solve how we might induce or inhibit the process in future drug design.

The current model of NMDA activation, the "Venus fly trap" model, posits that activation occurs when the two lobes of the LBD clamp down on the ligand. However, this hypothesis contradicts the finding that the binding of partial agonists, which are known to lead to sub-optimal activation, also close the LBD cleft completely [12, 13]. Indeed, recent simulation studies to determine the dynamics of the LBD cleft of the NMDAr have shown that the energy landscape of the protein is very broad, and potentially multistate even in the bound form of the primary agonist, indicating that there are many conformational states that are weakly stable [14]. Because of the size and complexity of the molecule, and the timescales ($\mu s - s$) involved in the opening/closing of the cleft (and associated conformational changes), previous computational studies have been limited to the characterization of the bound form of the primary agonist, and have employed aggressive advanced sampling techniques focused on the projections of the configurational space on specific sets of order parameters. The approach proposed below is expected to avoid the pitfalls of conventional molecular dynamics and insufficient sampling. In addition we have an ongoing collaboration (see collaboration letter) with the experimental group of Dr. Christy Landes (Rice University), who's currently studying the dynamics of this system by single molecule spectroscopy, as discussed below. The combination of our simulation with the experimental data from Landes' group will allow us to completely characterize and understand the NMDAr activation and desensitization processes.

Up until now, structural analysis of this system has been performed via either crystallographic or NMR methods, allowing for a full picture of only the most stable conformation to be obtained [10]. This information is static and must be paired with indirect measure of the conformational states using electrophysiological mea-

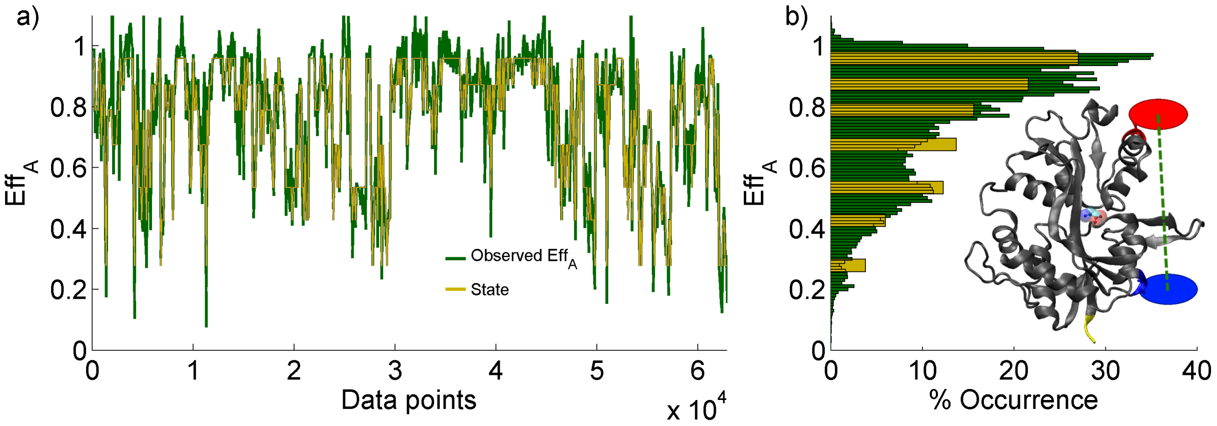

Figure 1: Several single molecule FRET trajectories were combined to provide a large enough basis to accurately locate the states, a downsampled portion of which is shown in (a). Measured single molecule FRET distribution for the NMDAr with glycine bound is shown in (b). Inset shows the structure of the NMDAr GluN1 isolated LBD bound to glycine based on the 1PB7 crystal structure. The protein strand was mutated at T193 (red marker) and S115 (blue marker) to cysteine in order to attach the acceptor and donor fluorophores. A His-tag was added to the protein at the N-terminus (yellow marker) to allow for immobilization. The FRET distance is indicated by the green dashed line and changes as the protein opens and closes the cleft. Measured distance of 34.2 Å from the crystal structure matches well with the calculated 32.2 Å distance based on the peak position of the single molecule histogram.

surements to monitor current flow through the channel. Recently, Landes’ group has used single molecule FRET (smFRET) experiments to observe fluctuations of conformational states of the NMDAr in real time, by recording the distance between two fluorescent dyes placed on either side of the cleft opening (see Fig. 1, unpublished results from Landes’ group). Performing smFRET experiments allows for the observation of individual kinetics as well as the ensemble distribution. A number of advances in the preparation and protection of the fluorescent tags used in single molecule measurements enables observations of individual proteins for multiple seconds, so that kinetics within the millisecond to second time scale can be captured [15].

The experiments were performed on the NMDAr GluN1 isolated ligand-binding domain under glycine-bound conditions, thus allowing the observation of the conformational landscape that potentially comprises both the desensitization and denaturation dynamics. The experimental results are best described by a model of 7 states (see Fig. 1). Two main regions of states were observed in the ensemble smFRET distribution, those with high FRET efficiency and those with low FRET efficiency. The 4 states with high FRET efficiency (0.96, 0.87, 0.79 and 0.68) can be further divided into two categories, the 0.96 and 0.87 FRET efficiencies correspond to closed cleft conditions associated with an “activated” channel, and 0.79 and 0.68 FRET efficiencies that correspond to a more open conformation. This assignment is based on the known crystal structure of the antagonist bound NMDAr that has a cleft distance of 39 to 40 Angstroms, corresponding to a FRET efficiency of around 0.82. The states probed by the lowest FRET efficiencies (0.53, 0.43, 0.28) indicate that the protein is in a more open conformation than observed in the full receptor through bulk ensemble studies. This further opening of the LBD fragment is likely due to the lack of steric hindrance experienced by the isolated LBD structure that is used in the experiments. In the full-length NMDAr, neighboring segments of the tetrameric structure limit the degree of cleft opening. Access to the broad range of FRET efficiency values might also be associated with the complete folding/denaturation landscape of the NMDAr LBD. One of the primary goals of the proposed computational effort is to determine the exact cause of these low FRET states.

1.2 Extreme-scale “many simulation” workload management and Execution

There are many science and engineering applications that are characterized not by the time-to-solution of a single task, but by the make-span of a set of tasks. An important distinction between such logically distributed “many simulations” applications and “traditional” HPC applications [16], is in the characteristics and requirements of an application's execution (run-time) phase. For example, HPC applications typically have a fixed resource requirement over their lifetime and thus acquire a static set of resource(s) at the start of the application; in contrast, most logically distributed applications should be flexible in their resource utilization, so as to be able to effectively utilize a changing resource pool [17, 18].

Resource utilization and workload execution decisions however, have been mostly statically determined, and decisions in one realm have typically been decoupled from the other. This has resulted in inefficient resource utilization, limited scalability, as well as inflexible execution strategies [19]. Breaking free of the static execution and resource utilization mode [17, 19] is a necessary condition to reach extreme-scales [20, 21]. Even though the requirements and world view of applications has changed significantly since [22–24], most supercomputing centers have their user and runtime environments tuned to support mostly single-job oriented workloads. In essence, the supercomputing environment still presents a very batch-queue, job-centric view, reminiscent of the inception days of high- performance computing.

The effective mapping of large-scale, logically distributed "many simulation" (where a simulation could be anything from a traditional large-scale MPI application to a small data-intensive processing task), to resources is a multi-level and integrated problem, which historically has been conceptually thought of, and implemented as either a monolithic, or un-integrated single-level problem. There is a critical need to evolve from these point solutions (i.e., approaches that try to solve only the resource utilization problem, or just the application execution problem) towards more comprehensive resource management capabilities [16].

The need to integrate applications and resources with different levels of granularity, control, dynamism etc., imposes a set of challenges. From an analysis [22–27] of application requirements, we find that there is a need for workload management solutions that support resource selection, dynamic execution decisions, and performance trade-offs using both application characteristics and resource (static and dynamic) information.

Given the fundamental importance of "many simulations" , the pilot abstraction concept is a natural starting point for exploring solutions. The P* Model of pilot abstractions [28]) has provided a firm theoretical foundation to the pilot concept. The most common representation of the pilot abstraction is in the form of Pilot-Jobs [29–34].

As effective as the Pilot-Job approach is, its scalability is limited by the fact that all tasks can't be treated equally in the execution phase; some task-specific information is required in order to manage the workload execution. Per se, the pilot abstraction provides a passive approach to the execution of "many simulations". Overcoming the limitations of a passive approach, we will deliver an "active" management system, viz., an integrated workload management and execution capabilities based upon the Pilot abstraction for a range of workload types and granularity (large MPI simulations to processing heavy data-intensive workloads, short-running to long-running). This will form the execution engine of Version 3 of the ExTASY toolkit; it will also lead to a stand-alone capability – the Overlay-based Workload Management System, which will be a component in the suite of Radical-Cybertools.

1.3 HPBDS: High-Performance Big Data Stack

Many scientific problems depend on the ability to analyze and compute on large amounts of data. This analysis often does not scale well i.e. its effectiveness is hampered by the increasing volume, variety and rate of change (velocity) of big data. The HPBDS project will design, develop and implement building blocks that will enable a fundamental improvement in the ability to support data intensive analysis on a broad range of cyberinfrastructure including that are supported by NSF for the scientific community. The project will integrate features of traditional high-performance computing, such as scientific libraries, communication and resource management middleware, with the rich set of capabilities found in the commercial Big Data ecosystem. The latter includes many important software systems such as Hadoop available from the Apache open source community.

The aim of HPBDS is to aim for the performance of HPC and the breadth and productivity of Apache Big Data stack (ABDS). The resultant integrated architecture is targeted at both production high-end computing platforms, as well as (commercial) cloud computing. As part of HPBDS, we propose two fundamental building blocks, Middleware for Data-Intensive Analytics and Science (MIDAS) and the Scalable Parallel Interoperable Data Analytics Library (SPIDAL).

The SPIDAL libraries will have the same beneficial impact on data analytics that scientific libraries such as PETSc, MPI and SCALAPACK have had for supercomputer simulations. These libraries will be implemented to be scalable and interoperable across a range of computing systems including clouds, clusters and supercomputers.

The HPBDS project will design and implement a Middleware for Data-Intensive Analytics and Science (MIDAS) that will enable scalable applications via the use of SPIDAL to have the performance of HPC (High Performance Computing) and the rich functionality of the commodity Apache Big Data Stack. In conjunction with SPIDAL, MIDAS will support new programming and execution models for data-intensive analysis in a wide range of science and engineering applications.

A specific focus of HPBDS will be the support of trajectory (data) analysis for biomolecular simulations. Computer simulations of biomolecular systems have been steadily increasing in length and spatial complexity, enabled by steady advances in computing hardware and algorithmic developments. However, analysis of the resulting raw data (trajectories with millions of time steps and up to millions of coordinates) has not kept pace with data production and is becoming a bottle-neck for researchers in the field due to two main reasons: (1) Current analysis tools and libraries do not generally utilize the parallelism provided by modern platforms, thus with increasing volumes of data the time for analysis can become a substantial fraction of the time to generate the data, and (2) most domain scientists do not have the computational background to leverage current state-of-the-art algorithms nor the tools to do so, and thus much time is spent in implementing and re-implementing ad-hoc solutions instead of pursuing the scientific question.

The situation is exacerbated on Blue Waters, where an individual scientist utilizing up to O(10,000) cores could generate 0.1 GB per nanosecond simulated for each replica, thus 10 3 GB are generated per 10 microseconds simulated, or 106 GB for O(1000) replicas, i.e., 1 PB for each typical science problem! The absence of parallel analysis techniques thus renders scientific gains arising from the speed of simulations ineffective. There is at the top-level the need to develop parallel renderings of different trajectory analysis algorithms that are transparent to users who are not expert on high performance data analysis. In HPBDS the ultimate goal is to leverage both the middleware capabilities and analytics libraries to make analysis and execution more scalable.

HPBDS is a 5 year (2014-2019) $5M NSF-ACI DIBBS project. This project is led by Fox (Indiana) with Jha co-leading the middleware (MIDAS) design and development of the project, as well as leading the development and integration with bio-molecular sciences.

2 Target Problems

2.1 Science Research Objectives and Milestones

The primary theoretical challenge to the characterization of the NMDAr LBD is formulating a simulation strategy that will thoroughly sample all conformational states relevant to receptor function. Once this is done, a kinetic model can be constructed to show how the states are dynamically linked and directly compared with experiment. This is a fundamental first step to understanding the conformational dynamics in their biological context. Since large sizes and long timescales of motion make conventional all-atom molecular dynamics prohibitively slow, a new approach is needed. The proposed methodology will combine the recently developed extended Diffusion Map directed MD (DM-d-MD) [3], that is in turn based on the Locally Scaled Diffusion Map (LSDMap) approach [35], and the COCO method [36] to rapidly and efficiently construct a detailed picture of the relevant regions of the protein conformational landscape. Details on the methodology are provided in section 2.2, while section 5.2 dicusses the computational requirments to achieve our goal.

As mentioned in the previous section, NMDAr forms obligate hetero-tetrameric assemblies [6] comprised of both glycine activated (GluN1 and GluN3) and glutamate activated (GluN2) subunits. The conformational dynamics of the individual ligand-binding domains is a key step in understanding their role in the larger NMDA complex [37]. While the dynamics of the ligand-binding domain are usually described as an open or closed clamshell (i.e., the "Venus fly trap" model), it has become clear that the underlying functional landscape contains a diverse range of conformations that will be key in understanding receptor function [38]. For example, experiments on the GluN1/GluN2 ligand-binding domains have shown that they have diverse signalling responses in the presence of ligands of varying size and stereochemistry [39]. Additionally, recent computational investigations indicate that the conformational landscape of the ligand-binding domain is much more complex that the one-dimensional clamshell picture [14].

Existing crystal structures will be used as initial conditions to a more complete sampling of conformational space. Existing structures of the isolated GluN1/GluN2 ligand-binding domain are very similiar (≤ 3 Å rmsd) to the ligand-binding domains of the recently solved crystal structure for the full NMDA complex [40]. All our simulations will be carried out in glycine-bound conditions, in order to make a direct comparison with the smFRET results obtained in the same conditions. However a set of different structures will be used as starting configurations for the sampling of GluN1 ligand-binding domain. In particular, we will use the apo (4KCC) and glycine-bound (1PBQ) conformations as these directly correspond to simultaneous work being carried out by Landes' group. Additionally starting conformations will be taken from other glycine-bound structures.

Once the configurational space has been thoroughly sampled, the next challenge is to identify a reaction coordinate that will effectively distinguish the states of interest. Reaction coordinates based on chemical intuition, such as angles or pairwise distances, are prone to fluctuations and likely to lump together states that should be separated. In order to identify a reaction coordinate that "naturally" captures the long timescale motion we will again apply LSDMap, which uses the intrinsic connectivity of the high-dimensional configuration data to determine optimal reaction coordinates for motion on long timescales [35, 41]. The LSDMap algorithm determines collective coordinates solely from the simulation data and without the need for intuition. This will be an essential tool as previous research has shown it to be effective at separating states that are indistinguishable by conventional coordinates [42]. These collective coordinates will allow us to identify the different free energy minima populated on the protein landscape, and the connections among them. While a direct kinetics analysis will not be possible as we use non-equilibrium methods to speed up the sampling, free energy barriers between states can be evaluated and will provide a picture of the connectivity among the different states.

These steps will paint the details of the conformational space and the barriers to conformational interconversion. The end result will be a quantitative model of LDB dynamics that can be used to make experimentally testable predictions. The theoretical model will be tested by predicting observables such as expected FRET efficiency histograms from different locations of the fluorescent chromophores as well as the optimal chromophore arrangement to measure the conformational kinetics. The prediction will be tested experimentally in Landes' lab and new experiments will be designed to corroborate the theoretical model. We expect this study will be able to shed light on the merits of the "Venus fly trap" model for the cleft closure of the ligand-binding domain of NMDAr, and the desensitization process. Understanding the dynamics of the NMDAr will have an important impact on the fundamental science of protein dynamics as well as practical impacts in the field of medical research and drug discovery. Demonstrating that large-scale conformational changes can be understood with a small number of collective coordinates will open the door to the study of larger systems and the organizing principles of living matter.

2.2 Methodology

As mentioned above, we propose to use a combination of the extended DM-d-MD and COCO methods to sample the conformational space of NMDAr LBD. The extended DM-d-MD method [3] is based on the computation of the locally scaled diffusion map (LSDMap) associated with molecular dynamics. LSDMap is a recently developed method [35, 41, 42] used to decouple molecular dynamics on different time scales in terms of a few collective coordinates, referred to as *diffusion coordinates* (DCs). In essence, extended DM-d-MD uses the information on the slowest directions of motion (that are provided by an on-the-fly LSDMap analysis of the space sampled), to direct the computational resources toward the sampling of rare events, such as crossing of large free energy barriers in a complex system as NMDAr LBD.

The algorithm has been designed (i) to enhance the sampling of MD trajectories, especially along high-energy barriers of macromolecular systems and (ii) to recover information about Boltzmann distributions directly from the biased trajectories. The former is made possible by periodically restarting MD trajectories by selecting the new starting points so as to obtain a uniform distribution of their first and second DCs. Points located along transition regions have very different values of their DCs whereas the DCs of points located inside a given minimum are very similar. Within these minima, regions associated with faster transitions can be identified from points having very different values of their 2^{nd} DC whereas points with similar 2^{nd} DC will account for local minima, and so on. Using a uniform sampling along the first two diffusion coordinates to select the new starting points thus appears to be a natural choice to cover the largest possible area of the configuration space, *i.e.*, to visit yet unexplored regions without being trapped within local minima.

Regarding Boltzmann distribution, *i.e.*, point (ii), it is based on the assignment of statistical weights to each trajectory. In order to save information associated with molecular dynamics, an appropriate reweighting scheme, based on nearest neighbor search, is used each time MD trajectories are restarted. The weights of each point are used self-consistently in order to compute LSDMap at the next iteration.

Similar in spirit but different in practice, the COCO (Complementary Coordinates) method also speeds up the sampling of rare events, by iteratively starting ensembles of simulations from "interesting" points in configurational space identified from analysis of the prior sampling [36]. While extended DM-d-MD uses LSDMap to analyze the space sampled, COCO uses Principal Component Analysis (PCA) in Cartesian space, and while the new points chosen by the DM-d-MD method come from the existing ensemble, COCO proposes new initial points for trajectories that correspond to regions in PCA space that have so far never been sampled. Because it explicitly explores beyond the limits of the current ensemble, COCO can enable a more rapid initial exploration of conformational space than DM-d-MD, however it has the disadvantage that as the structures generated by COCO are novel they a) may be unphysical and b) cannot be assigned a statistical weight and so preclude the regeneration of a Boltzmann-weighted ensemble in the way that DM-d-MD permits. Thus our approach is to use COCO as an initial tool to give a fast, approximate, sampling of conformational space and then DM-d-MD to further explore, refine, and properly weight the sampled configurational space to converge to a Bolzmann ensemble. The workflow is depicted in Figure 2.

2.3 Cyberinfrastructure Research Objectives and Milestones

An important general motivation of the CI research and development arises from the many instances of science & engineering workflows that depend upon the effective and efficient execution of "many simulations", and which seek to exploit high-performance resources, but are encountering multiple barriers. However, the CI research and development to be undertaken on Blue Waters, although funded and ongoing as part of other NSF funded projects, are motivated by the specific requirements of the science case presented earlier in §1 and §2.

These barriers can be categorized into several types: (P1) existing applications that need to support new workload characteristics, new resource types and/or execution modes, or (P2) applications that are unable

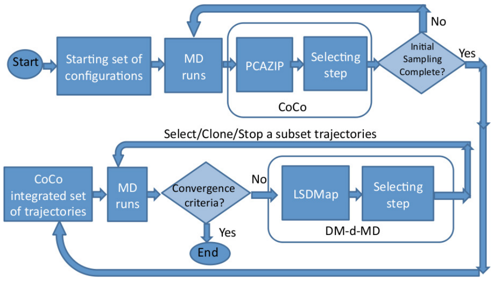
Start
Starting set of configurations
MD runs
PCAZIP
Selecting step
CoCo
No
Initial Sampling Complete?
Yes
Select/Clone/Stop a subset trajectories
CoCo integrated set of trajectories
MD runs
Convergence criteria?
No
Yes
End
LSDMap
Selecting step
DM-d-MD

Figure 2: Illustration of the workflow that will be adopted to sample the conformational space of NMDAr LDB. First the COCO method will be used to explore a large region of the configurational space. By using the configurations generated by COCO as starting points, the extended DM-d-MD approach will provide a Boltzmann ensemble of configurations spanning the system most relevant minima and the paths connecting them.

to scale to the requisite level and are thus throttled in their ability to use the computational capabilities that their science demands. Delivering these capabilities requires integration through multiple levels of application, middleware and resource management capabilities. Given the scope of the problem and solution space, improving the throughput of applications and ensuring their effective usage of large machines such as Blue-Waters requires *cross-layer and cross-cutting approaches*.

3 Description of the Computational Codes to be used

3.1 Simulation and Analysis Codes

We have selected **GROMACS** (http://www.gromacs.org) as the MD engine for the trajectory generation phase. GROMACS has an emphasis on high performance, and in particular contains optimised force evaluation routines for various CPU micro-architectures (including AVX), a mixed-mode MPI and OpenMP parallelisation, and support for GPU acceleration using CUDA. Thus it is well suited to take advantage of the hardware available on Titan.

For the analysis phase we will use **LSDMap** (http://sourceforge.net/projects/lsdmap/) and for initial configuration generation **COCO** (see https://bitbucket.org/extasy-project). Both are implemented in Python and parallelised using MPI (via the mpi4py package), and depend on the NumPy and SciPy libraries for efficient numerical operations. In addition, COCO depends on the MDAnalysis Python package. At present the analysis programs do not make use of GPU acceleration, but by construction they constitute a relatively small fraction of the overall computation cost.

The challenging component is not the scaling of Gromacs or the efficient execution of the analysis codes; the former is the perview of the Gromacs development team and the latter although our responsibility is not a major component of the computation (at or less than 1% of the time is spent in the analysis phase). The critical challenge is the concurrent support of O(1000) simulations and possibly upto O(10000) that must be made possible in order for the effectiveness of the analysis phase and thereby advanced sampling to become evident. The execution of large-scale ensemble of MD simulations coupled, as well as their coupling with analysis will be managed by the RADICAL-Pilot software which we now discuss.

3.2 RADICAL-Pilot

Many applications require multiple tasks (viz., simulations or jobs) to produce useful output data. A traditional way of performing these tasks would be to submit an individual job for each task. These jobs would normally stay in the batch queuing system and would not become active at the same time. A Pilot-Job however can both simplify and make more efficient this process.

A Pilot-Job can be thought of as a container job for many sub-jobs (Compute Units), with sophisticated workflow management to coordinate the launch and interaction of actual computational tasks within the container. A Pilot-Job acquires the resources necessary to execute the sub-jobs (thus, acquires the resources required to run the sub-jobs, rather than just one sub-job). If a system has a batch queue, the Pilot-Job is submitted to this queue. Once it becomes active, it can run the sub-jobs directly, instead of having to wait for each sub-job to queue. This eliminates the need to submit a different job for every executable and significantly reduces the time-to-completion.

RADICAL-Pilot (http://radical-cybertools.github.com/radical-pilot) is a Pilot-Job framework that allows users to reserve and utilize high-capacity resources on a target machine. Pilots provide a fundamental abstraction for task-level parallelism, by supporting concurrent execution of large number of tasks. The Pilot approach supports the decoupling of workload submission from resource assignment. This results in a flexible execution model, which in turn enables the scaling of applications on multiple and possibly heterogeneous resources. Many applications require multiple tasks (viz., simulations or jobs) to produce useful output data. A traditional way of performing these tasks would be to submit an individual job for each task. These jobs would normally stay in the batch queuing system and would not become active at the same time. A Pilot-Job however can both simplify and make more efficient this process.

RADICAL-Pilot is the specific Pilot-Job that will be utilized in this project. It is designed for scalability and provides a well-defined interface, usage and execution model, whilst retaining the flexibility to operate in different environments and being extensible. We will work to improve RADICAL-Pilot scalability and lower its coordination overhead, so as to effectively coordinate O(10^4-10^5) ensemble members concurrently; resources requested as part of T2 will help. RADICAL-Pilot builds upon important conceptual theoretical advances (P* Model [28]), an interoperability layer, and a well defined usage and programming model (and a simple interface exposing these). The interoperability layer used by RADICAL-Pilot is SAGA — which provides both a syntactic and semantic unification via a single interface to access multiple, semantically distinct middleware.

During the course of this project our research and experiences here and elsewhere, will result in formalization and developments to produce an Overlay based Workload Management System (OWMS) that will support the effective execution, placement and resource management of "many simulations" based applications. OWMS will be an important addition to the suite of tools comprised of SAGA [43] and RADICAL-Pilot [44] collectively called RADICAL-Cybertools [45]. RADICAL-Cybertools will collectively provide building blocks for the interoperable execution of different types of "many simulations" based applications on Blue Waters multiple high-end machines.

Although both a vehicle for research and the embodiment of research, it is important to point out that RADICAL-Cybertools already supports production grade science. RADICAL-Cybertools are currently used globally – on production DCI, Campus Clusters and departmental servers, by multiple science projects – from climate science modeling community [46, 47], to bio-molecular simulations [48] to high-energy physics (ATLAS) community [49] on DOE leadership class machines such as Titan and Hopper, to name just a few. On XSEDE they currently support 10M SUs per year for external science groups. In addition, RADICAL-Cybertools support another (at minimum) 10M CPU-hrs per year at other non-XSEDE sites, such as Campus Clusters.

3.3 ExTASY Toolkit

The ExTASY toolkit provides a well-defined library/API for the execution of different incarnations of the ensemble-based simulation-analysis patterns, in a way that is independent of the specific number of ensembles as well as the amount of resources available. This agnosticism to the "supply" and "demand" makes the execution of the simulation-analysis patterns simple and scalable.

The ExTASY Toolkit uses RADICAL-Pilot as the underlying execution substrate, to support the execution of a large number of ensemble simulations (tasks) coupled with an analysis phase. The coupled simulation-analysis execution pattern currently supports GROMACS (or AMBER, or even NAMD) as the simulation kernel, and LSDMap or COCO as the analysis kernel.

Each iteration of the pattern consists of a simulation phase and an analysis phase. Both the simulation and the analysis are carried out on the target remote machine (here Blue Waters). RADICAL-Pilot manages the job execution and data movement, simply and intelligently. The data movement during these phases between the local and the remote machine is handled by RADICAL-Pilot. RADICAL-Pilot also handles the proper setup of the environment to run the simulators and analyzers on different machines.

The "iterative simulation-analysis" workflow forms a chain of simulation-analysis patterns such that after the analysis phase, the configurations are updated and the simulations are started again with these updated values. Irrespective of whether the kernels toggle between simulation and analysis in a regular or irregular fashion, the kernels get the resources acquired by the Pilot and do not have to wait in queue. The frequency of simulation-analysis toggling is dependent upon the number of ensemble-members being investigated, as well as the ratio of computational cost of the simulation and analysis phase

4 Development Plan

4.1 Scaling RADICAL-Pilot to O(10000) concurrent tasks

The dominant use of RADICAL-Pilot involves the execution of large numbers of simulations, which in RP parlance is referred to as ComputeUnits (CU). In general, RADICAL-Pilot ability to scale depends on three main properties: (i) the performance of transferring CU information to the pilot agent on the target resource; (ii) the performance of staging required input data towards the target resource; and (iii) the performance of placing and starting the CUs on the target resource.

For the purposes of supporting O(1000) concurrent simulations, on Blue Water, the third factor is most important, which we discuss now: It is worth recalling that the rate at which a pilot agent can manage workloads is a determinant of resource utilization: if the processing of CUs is too slow and cannot keep up with CU completion, then the pilot agent will not be able to sustain the utilization of the allocated and available cores.

Additionally, the pilot has to work with a number of hard and soft system limits when spawning the CUs, such as the maximum number of concurrent processes, maximum number of open file descriptors, etc[1]. At the moment, RADICAL-Pilot is able to spawn about 3 CUs/second, for up to 4k concurrent CUs, per pilot instance[2]. Scaling beyond those numbers is feasable, and a roadmap to do so exists – but it requires solutions which are tuned toward specific resource properties (system limits, usage policies, host topologies) at large scales. For Blue Waters the pilot agent needs to be split into multiple components for scale-up, increasing the complexity of the agent.

4.2 ExTASY Development

Both COCO (and the underlying PCAZIP toolkit) and LSDMap have been entirely re-written during the ExTASY project in Python, replacing legacy Fortran implementation with an object-oriented toolkit approach which allows these to be easily composed into a larger workflow. While already parallelised (see 5.1) work is ongoing to scale these code further using MPI. For LSDMap, which processes only the final frame in each trajectory, this will enable the use of even larger ensembles of shorter trajectories, improving the overall rate of sampling.

The analysis of the configurational landscape and the identification of the free energy minima and comparison with the experimental data will be performed after each set of COCO + DM-d-MD iterations. The value of

[1] For a good analysis and discussion of those system limits, see **??**

[2] The size of each CU does not factor in, ie. those CUs can each be larger MPI or OpenMP instances

different observables (such as relative differences in free energy between different states) will be monitored as a function of the iteration number to establish the convergence of the simulations [3]. We are working to improve the scaling of LSDMap. Currently optimal performance is achieved at 384 cores (24 nodes), with a factor of 2 enhancement easily achieved. LSDMap can be used with 20% slowdown uptil 1024 cores. However, further strong scaling will require effort to make it "optimal" at O(1000) cores. Whereas we are confident of reaching optimal efficiency at O(1000) cores in the 3-6 month window (mostly via use of efficient linear algebra/matrix libraries), in the interim we will employ the following "efficiency approach". For the sum total of (S) ensembles, we will start multiple tracks (T) of ¡ensemble-simulations, each of which will contain N ensemble-members, i.e., S = T x N. The value of N will be determined to be such that the functionality/performance of LSDMap phase will be acceptable, but will not slow down the overall progress of the simulation phase. This will be upto estimated typical values of T will be 2-8. RADICAL-Pilot easily supports such multi-track simulation-analysis execution.

4.3 SPIDAL-MIDAS

The aim of MIDAS is to provide a scalable runtime system and appropriate resource management abstractions enabling SPIDAL, and thereby Big Data applications. MIDAS provides the underlying resource management middleware and heterogeneous infrastructure access layer which will support SPIDAL libraries to work efficiently across these application types over a range of resources.

Figure 3 shows the architecture of MIDAS which has two primary design objectives: (i) Provide high-level abstractions (e.g. scalable data processing, inter-process communication and storage supporting both query and analysis), so as to hide details of different lower level implementations (e.g. for accessing data or resources via HPC schedulers such as SLURM, Big Data schedulers, such as YARN [50] or Cloud backends like Amazon and Google), (ii) provide a flexible middleware to support four key programming models: (PM1) Pleasingly parallel, (PM2) Search using Classic MapReduce, (PM3) Iterative MapReduce with Collectives and (PM4) Iterative Graph Processing. MIDAS will provide the scalable runtime system to support these programming models via appropriate execution-processing capabilities on different platforms.

Communication abstractions enable the coordination and exchange of data between tasks. In particular iterative MapReduce tasks need collective operations while graph processing largely needs point-to-point communication. We have already shown that using classic MPI techniques can provide a collective layer that outperforms existing (iterative) MapReduce approaches on both cloud and HPC environments [**?**, 51, 52].

As demonstrated by the evolution to YARN, there is an increasing need to provide application specific scheduling and resource management control; the Pilot abstraction has demonstrated such capabilities. Application-level scheduling as provided by the Pilot-layer will be an essential tool to integrate library resource usage modes (see processing patterns) with resource allocation/usage.

4.4 Support from Blue Waters Team:

We will use standard Gromacs compilations on Blue Waters. Given that each ensemble member will be run at 256 cores and physical models have been simulated before, we are confident that memory and I/O usage of individual ensemble members are well characterized and bounded. The stress to the systems arises from the number of ensembles that we we will be able to simulate. Given the target concurrent execution of O(10^3-10^4) Gromacs kernels, it will likely stress the I/O sub-systems [53] (if experiences with Kraken are any guide). A further major problem in scaling to larger number of ensembles will be the large load on the compute head-nodes; our existing approaches have resulted in unsustainable loads on Stampede. An unoptimized coordination mechanism for O(10^3-10^4) ensemble members on Blue Waters is unlikely to be acceptable. We will need to work with Blue Waters technical team to address these issues, as well as to profile techniques to ensure coordination costs/overhead are acceptable.

5 Resource Required

We begin by discussing the parallel scaling of primary components of the workflow; then we discuss how we will optimize scalable task-level execution as we improve the strong scaling properties of the analysis phase.

5.1 Parallel Performance

For benchmarking and CPU time estimation we have prepared a GROMACS input file consisting of a single NMDA protein (4639 atoms) in explicit solvent, for a total of 54652 atoms. The simulation runs Parrinello-Rahman NPT MD for 1 nanosecond, writing one frame of the trajectory every picosecond. Particle Mesh Ewald (PME) summation is used for long-range electrostatic forces. We have carried out strong-scaling experiments on ARCHER (Cray XC30, Intel Xeon 2.7 GHz E5-2697, 24 cores per node) using GROMACS 4.6.5 (table 1) with MPI only.

Nodes	Cores	Walltime (s)	Efficiency (%)	core-hours per ns
1	24	3363	100	22.4
2	48	1840	91	24.5
4	96	1079	78	28.8
8	192	685	62	36.5
16	384	535	39	57.0
32	768	375	28	80.0
64	1536	386	14	164.7

Table 1: Strong scaling of NMDA MD with GROMACS on ARCHER

To extrapolate this scaling data to expected performance on Blue-Waters we have carried out additional benchmarks on a local system with AMD Opteron 6276 ‘Interlagos’ 2.3 GHz processors and an Nvidia Tesla K20c GPUs. The CPU is identical apart from 0.1 GHz difference in clock speed and the GPU on Blue-Waters has one extra SMX processor (14 instead of 13) and an additional 1GB memory. Thus we believe this to be a reasonable proxy for the expected on-node performance on Blue-Waters. GROMACS requires a 1:1 ratio between GPUs and MPI processes, so the 16 cores per node are harnessed using OpenMP. On this system, using the GPU plus 16 OpenMP threads, the MD benchmark took 5929s, approximately 1.76x slower than a single node on ARCHER. Assuming similar scaling efficiency (a conservative assumption since the slower per-node performance typically leads to better scalability) we compute the predicted strong scaling behaviour on Blue-Waters in Table 2 running on 4 nodes is a reasonable compromise between efficiency and time-to-solution, and we would expect a cost of **33.7 core-hours per ns** for our system.

Nodes	Cores	Walltime (s)	Efficiency (%)	core-hours per ns
1	16	5918	100	26.3
2	32	3239	91	28.8
4	64	1898	78	33.7
8	128	1206	62	42.9

Table 2: Predicted scaling data for Blue-Waters

LSDMap takes as input the endpoints of an ensemble of trajectories and computes their coordinates in the diffusion coordinate space. We have carried out strong-scaling tests using different ensemble sizes of NMDA trajectories on ARCHER and summarise the results in Figure 4. While the strong scaling is limited, we can complete the analysis with up to 10,000 trajectory end-points in only 68 seconds using 384 cores. Based on tests which suggest the CPUs on Blue-Waters are around 2.5x slower per core than on ARCHER, this gives **170s per analysis using 24 nodes, or 18 core-hours**.

The main computational task in COCO is a Principle Component Analysis (PCA), implemented by the Python PCAZIP toolkit. We have tested this on ARCHER with a set of 1,000 1-nanosecond MNDA trajectories and the strong scaling results are summarised in table 3. The PCA is significantly cheaper than LSDMap, and even given that Blue-Waters is slower than ARCHER we can complete the analysis in **330s using 1 node, or 1.5 core-hours**.

Cores	Walltime (s)	core-hours
1	2831	0.8
2	960	0.5
4	368	0.4
8	197	0.4
16	132	0.6
24	125	0.8
36	104	1.0
48	101	1.4

Table 3: Strong scaling test data for PCAZIP on ARCHER

We have selected **GROMACS** (http://www.gromacs.org) as the MD engine for the trajectory generation phase. GROMACS has an emphasis on high performance, and in particular contains optimised force evaluation routines for various CPU micro-architectures (including AVX), a mixed-mode MPI and OpenMP parallelisation, and support for GPU acceleration using CUDA. Thus it is well suited to take advantage of the hardware available on Blue-Waters.

For the analysis phase we will use **LSDMap** (http://sourceforge.net/projects/lsdmap/) and for initial configuration generation **COCO** (see https://bitbucket.org/extasy-project). Both are implemented in Python and parallelised using MPI (via the mpi4py package), and depend on the NumPy and SciPy libraries for efficient numerical operations. In addition, COCO depends on the MDAnalysis Python package. At present the analysis programs do not make use of GPU acceleration, but by construction they constitute a relatively small fraction of the overall computation cost.

Each picosecond of molecular dynamics outputs 1.3 GB of data at a rate of around 1 MB/s. Given that 1000 ensemble members will run in parallel, this equates to a sustained aggregate write bandwidth of 1 GB/s.

The execution of large-scale ensemble of MD simulations coupled, as well as their coupling with analysis will be managed by the RADICAL-Pilot software.

5.2 Use of Resources Requested

As illustrated in Figure 2, our simulation methodology can be broadly understood as two stages (first COCO, then extended DM-d-MD), both of them consisting of iterations with two phases. In the *trajectory generation* phase a large number of relatively short MD calculations is performed starting from an initial set of configurations, and the resulting trajectories are stored. These are then fed into the *analysis* phase, which assesses the quality of sampling achieved so far by MD and generates a new set of configurations from which to restart MD.

To achieve the desired degree of sampling, we estimate 200 iterations will be required for the COCO stage and 20,000 iterations for the DM-d-MD stage. During the COCO state, at each iteration we will run 1000 MD ensemble members, each generating 1ns of trajectory data. In each iteration of the subsequent DM-d-MD stage we will run 10,000 ensemble MD calculations, each generating 10ps of trajectory data.

To balance the scalabality of the analysis phase, throughput of the iterations and available fraction of Blue-Waters, in the COCO stage we propose to use 4000 nodes (21% of the machine) running all 1000 ensemble members in parallel (see benchmark data in section 5.1 below). In each iteration, the 1ns-long trajectories will be completed in 1,900 seconds. As the analysis time in the COCO algorithm is negligable with respect to time required for the trajectory generation, the COCO stage requires **33,778 core hours per iteration**. In the DM-d-MD stage we propose to use again 4,000 nodes running 1,000 ensemble members in parallel. To complete all 10,000 10ps-long trajectories in the ensemble will take 190 seconds, and the analysis part takes 170 seconds on 24 nodes, for a total cost of 4,000 × 16 × 190 + 24 × 16 × 170, that is **3396 core-hours per iteration**.

Thus, to complete 200 COCO iterations + 20,000 generalized DM-d-MD iterations we request a total of **75 million core hours**. The walltime needed to complete the calculation is about 105 hours for the COCO stage and 2000 hours ($\sim$ 84 days) for the DM-d-MD stage.

We plan to iterate the procedure (a COCO exploration followed by extended DM-d-MD) twice, to fully exploit the complementary advantages of the two methods. That is, in the first year we will use half of the computational resources to run 100 COCO iterations followed by 10000 DM-d-MD iterations, for a total of $\sim$ 1,050 hours of walltime (about 37.5 million core hours of CPU time). We will then use the configurational space so obtained as a starting point to run another 100 COCO iteration and 10,000 DM-d-MD iterations.

Our work requires Blue-Waters because the total computational cost of our proposed work cannot be feasibly met by other resources. While each individual ensemble member will run on only a handful of nodes, the large number (1000s) of trajectories to be sampled concurrently could require around 21% of the entire machine. In addition, the computational demands and high frequency of the analysis precludes the use of traditional throughput computing.

Cyberinfrastructure R&D: Developing, testing and optimizing tools for execution and dynamic load-balancing, workload management systems and parallel data analysis techniques at extreme-scales requires access to the same extreme-scales as the production runs. The computing resources requested as part of the cyberinfrastructure of this proposal (5M SUs per year) represent the resources required to enable the R &D underpinning middleware and tool development for a large section of the simulations and modeling community that are important stakeholders in ensuring a capable and balanced cyberinfrastructure for the next level of scale. ***shantenu: Need to refine estimate for cyberinfrastructure development

6 Sources of Development Funding and Development Resources

As part of the PI's NSF CAREER Award (*NSF CAREER: Abstractions and Middleware for D3 Science on NSF Distributed Cyberinfrastructure*, ACI-1253644, 2013-2018) we are tasked to develop middleware to support tools and execution capabilities of logically (and physically) distributed resources that have "a dynamic resource utilization" on National Production CI. One class of applications that can exploit dynamic resources are those represented by "many simulations" class of applications. Our initial work has had some early success in enabling scale and addressing the aforementioned problems on XSEDE class of machines [54]; supporting "many simulations" on Blue Waters – which represents the top tier of the NSF's Branscomb Pyramid, is the need of the hour for a range of applications that we are currently supporting.

HPBDS is a 5 year 2014-2019 NSF-ACI DIBBS project. This project is led by Fox (Indiana) with Jha co-leading the middleware (MIDAS) design and development of the project, as well as leading the development and integration with bio-molecular sciences. ***shantenu: please fill in here

***shantenu: from solicitation: Source of Development Funding and Development Resources : If your application requires development to conduct the planned research at petascale, please identify the source, amount and duration of funding and computer resources that will be used to develop the necessary methods or algorithms.

7 Intellectual Merit

We propose here a new approach to characterize the conformational landscape of the NMDAr LBD in its different forms, by using novel sampling methodologies and workload management tools. Leveraging the sophisticated sampling methods provided by the ExTASY software toolkit and the compute power of Blue Waters's resource, our simulations will give us an unprecedented atomically-detailed picture of the different states the protein can adopt as a function of the ligands it interacts with (and their associated energies), and will also give us predictions of the kinetic pathways that link these states together. The result will be a conceptual framework to understand the sometimes perplexing experimental results and a springboard for the rational design of further experiments. Beside the importance of the NMDAr per se, this study will open the way to a conceptually different approach to study of large conformational changes in complex macromolecules, as the same methodology and computational infrastructure can in principle be applied to a large number of biomedically relevant systems.

This project will support several new classes of applications to utilize Blue Waters at scales that are scientifically needed but simply not possible elsewhere. The resources made available as part of this project, will enable the delivery of advanced software systems and data analysis tools that will support the end-to-end cyberinfrastructure requirements of these application. Any solution to support the "many simulations" application will need to move beyond building customized solutions that support only a specific workload type or usage mode. It might just be technically possible for large projects to develop their own customized software infrastructure for workload management. However, it is unproductive and unsustainable to do so for all major projects due to an excess of duplicated customization and development efforts. Furthermore, researchers in the long-tail of science possess neither the resources nor the person effort to develop custom software infrastructures. These domain scientists have a workload that requires many simulations, and have to prioritize getting the science done, while relying on externally provided software infrastructure. This project addresses the need to provide base capabilities, upon which it is possible to build a set of common and extensible set of higher-level services and capabilities. In addition by designing systems to "support as many scalable concurrent applications" as possible, a new scaling dimension emerges, viz., scaling the number of people that can use leadership resources.

Concomitant with advances in scalable "many simulations" there is a requirement to advance data-analytic capabilities for the prodigious volumes of data that will be produced. Given the typical global synchronization, intensive nature of communication and runtime cost (typically $O(N^2)$) of the most commonly used trajectory analysis algorithms, in the absence of parallel analytical approaches, existing trajectory analysis techniques will simply not scale to support the unprecedented volumes of data [55].

Broader Impact: Clementi and Jha are also PIs of the NSF ACI "SI2: Conceptualization of an Institute for Biomolecular Simulations", wherein a significant fraction of the spectrum of applications are "many simulations" based, e.g., the three "many simulations" application classes identified, make up a very large fraction (best estimate of up to 50%) of the total of biomolecular simulation on XSEDE. Enabling effective approaches to support the scalable execution of "many simulations" will thus be an important step forward in meeting the needs of the biomolecular simulation community.

Prior NSF Support Clementi.... ***shantenu: please complete

Geoffrey Fox was the PI of FutureGrid: An Experimental, High-Performance Grid Test-bed OCI0910812, 10/1/09-9/30/14 with with 8 partners. FutureGrid has been used in about 400 projects and several papers have been published by users and the FutureGrid team. Fox is the PI of HPBDS (ACI-1443054). ***shantenu: please complete

PI Jha has current award CAREER: Abstractions and Middleware for Distributed Dynamic Data-intensive Science on NSF Distributed Cyberinfrastructure (03/13-02/18). He is currently PI of RADICAL-Cybertools

(ACI-1440677) and a co-PI on HPBDS (ACI ACI-1443054). Jha serves as the cyberinfrastructure lead of the ExTASY project (CHE-1265788), Cyberenabled Discovery and Innovation project (Mapping complex biomolecular reactions with large scale replica exchange simulations on national production cyberinfrastructure, CHE-1125332) and “Collaborative Research: S2I2: Conceptualization of a Center for Biomolecular Simulation” (ACI-1331401). Important funding for SAGA was provided by the HPCOPS NSF OCI-0710874 (2007-2011) (Jha, co-PI) and Cybertools Project (NSF-RII EPS 0701491) (09/08-08/11), which led to over 50 publications and several open source community software packages.

References

[1] K. Lindorff-Larsen, S. Piana, R. O. Dror, and D. E. Shaw (2011) "How fast-folding proteins fold". *Science*, **334** (6055), 517–520.

[2] M. R. Shirts and V. S. Pande (2000) "Screen savers of the world, unite!" *Science*, **290** (5498), 1903–1904.

[3] J. Preto and C. Clementi (2014) "Fast recovery of free energy landscapes via diffusion-map-directed molecular dynamics". *Phys. Chem. Chem. Phys.*, (16), 19181.

[4] RADICAL Cybertools. "RADICAL Cybertools GitHub Home".
http://radical-cybertools.github.io/.

[5] M. Mayer (2006) "Glutamate receptors at atomic resolution". *Nature*, **440** (7083), 456–462.

[6] S. Traynelis, L. Wollmuth, C. McBain, F. Menniti, K. Vance, K. Ogden, K. Hansen, H. Yuan, S. Myers, and R. Dingledine (2010) "Glutamate receptor ion channels: structure, regulation, and function". *Pharmacol. Rev.*, **62** (3), 405–496.

[7] H. Querfurth and F. LaFerla (2010) "Alzheimer's Disease". *New Engl. J. Med.*, **362** (4), 329–344.

[8] K. Johnson, P. Conn, and C. Niswender (2009) "Glutamate receptors as therapeutic targets for Parkinsons disease". *CNS Neurol. Disord. Drug Targets*, **8** (6), 475.

[9] P. Chen and D. Wyllie (2006) "Pharmacological insights obtained from structure–function studies of ionotropic glutamate receptors". *Br. J. Pharmacol.*, **147** (8), 839–853.

[10] H. Furukawa and E. Gouaux (2003) "Mechanisms of activation, inhibition and specificity: crystal structures of the NMDA receptor NR1 ligand-binding core". *EMBO J.*, **22** (12), 2873–2885.

[11] S. Myers, R. Dingledine, and K. Borges (1999) "Genetic regulation of glutamate receptor ion channels". *Annu. Rev. Pharmacol. Toxicol.*, **39** (1), 221–241.

[12] A. Inanobe, H. Furukawa, and E. Gouaux (2005) "Mechanism of partial agonist action at the NR1 subunit of NMDA receptors". *Neuron*, **47** (1), 71–84.

[13] M. Ylilauri and O. Pentikäinen (2012) "Structural mechanism of N-methyl-D-aspartate receptor type 1 partial agonism". *PloS one*, **7** (10), e47604.

[14] Y. Yao, J. Belcher, A. Berger, M. Mayer, and A. Lau (2013) "Conformational analysis of NMDA receptor GluN1, GluN2, and GluN3 ligand-binding domains reveals subtype-specific characteristics". *Structure*, **21** (10), 1788–1799.

[15] C. Landes, A. Rambhadran, J. Taylor, F. Salatan, and V. Jayaraman (2011) "Structural landscape of isolated agonist-binding domains from single AMPA receptors". *Nature Chem. Biol.*, **7** (3), 168–173.

[16] S. Jha, M. Cole, D. S. Katz, M. Parashar, O. Rana, and J. Weissman (2013) "Distributed computing practice for large-scale science and engineering applications". *Concurrency and Computation: Practice and Experience*, **25** (11), 1559–1585. doi:10.1002/cpe.2897.

[17] F. D. Berman, R. Wolski, S. Figueira, J. Schopf, and G. Shao (1996) "Application-level scheduling on distributed heterogeneous networks". In *Supercomputing '96: Proceedings of the 1996 ACM/IEEE conference on Supercomputing*. IEEE Computer Society.

[18] H. Casanova, G. Obertelli, F. Berman, and R. Wolski (2000) "The AppLeS Parameter Sweep Template: User-level middleware for the Grid". *Sci. Program.*, **8** (3), 111–126.

[19] F. Berman, R. Wolski, H. Casanova, W. Cirne, H. Dail, M. Faerman, S. Figueira, J. Hayes, G. Obertelli, J. Schopf, G. Shao, S. Smallen, N. Spring, A. Su, and D. Zagorodnov (2003) "Adaptive Computing on the Grid Using AppLeS". *IEEE Trans. Parallel Distrib. Syst.*, **14** (4), 369–382. doi:10.1109/TPDS.2003.1195409.

[20] A. Luckow, S. Jha, J. Kim, A. Merzky, and B. Schnor (2009) "Adaptive Distributed Replica–Exchange Simulations". In *Theme Issue of the Philosophical Transactions of the Royal Society A: Crossing Boundaries: Computational Science, E-Science and Global E-Infrastructure Proceedings of the UK e-Science All Hands Meeting*, volume 367.

[21] A. Thota, A. Luckow, and S. Jha (2011) "Efficient large-scale Replica-Exchange Simulations on Production Infrastructure". *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **369** (1949), 3318–3335.

[22] "NSF XSEDE Annual Report (2012)". https://www.xsede.org/documents/10157/169907/2012+Q2+Annual+Report.pdf.

[23] "Survey of cyberinfrastructure needs and interests of NSF-funded principal investigators". https://scholarworks.iu.edu/dspace/handle/2022/9917.

[24] "XROADS User Requirements". http://www.ci.uchicago.edu/research/papers/CI-TR-10-0811.

[25] S. Jha, D. S. Katz, and J. Weissman (2011) "White Paper: Enabling Distributed Data-Intensive Exascale Research: Applications, Middleware, and Cyberinfrastructure". Technical report, Rutgers University, University of Chicago, University of Minnesota. Submitted to DOE ASCR (Rich Carlson), August 2011, http://www.cct.lsu.edu/~sjha/select_publications/D3-exa_jha_katz_weissman.pdf.

[26] S. Jha, N. C. Hong, S. Dobson, D. S. Katz, A. Luckow, O. Rana, Y. Simmhan, and J. Weissman (2013) "3DPAS: Distributed Dynamic Data-intensive Programming Abstractions and Systems". http://www.cct.lsu.edu/~sjha/select_publications/3dpas.pdf.

[27] J. L. Gustafson (1988) "Reevaluating Amdahl's law". *Commun. ACM*, **31**, 532–533. doi:http://doi.acm.org/10.1145/42411.42415.

[28] A. Luckow, M. Santcroos, A. Merzky, O. Weidner, P. Mantha, and S. Jha (2012) "P*: A model of pilot-abstractions". *2012 IEEE 8th International Conference on E-Science*, pp. 1–10. doi:http://doi.ieeecomputersociety.org/10.1109/eScience.2012.6404423. Http://doi.ieeecomputersociety.org/10.1109/eScience.2012.6404423.

[29] J. Frey, T. Tannenbaum, M. Livny, I. Foster, and S. Tuecke (2002) "Condor-G: A Computation Management Agent for Multi-Institutional Grids". *Cluster Computing*, **5** (3), 237–246. doi:10.1023/A:1015617019423.

[30] E. Heymann, M. A. Senar, E. Luque, and M. Livny (2000) "Adaptive Scheduling for Master-Worker Applications on the Computational Grid". In *Proceedings of the First IEEE/ACM International Workshop on Grid Computing*, GRID '00, pp. 214–227. Springer-Verlag, London, UK, UK.

[31] A. J. Rubio-Montero, E. Huedo, and R. Mayo-Garcia (2012) "GWpilot: a Personal Pilot System". In *Proceedings of EGI Forum*.

[32] (2011). "ToPoS - A Token Pool Server for Pilot Jobs". https://grid.sara.nl/wiki/index.php/Using_the_Grid/ToPoS.

[33] P. Nilsson, J. Caballero, K. De, T. Maeno, A. Stradling, T. Wenaus, et al. (2011) "The ATLAS PanDA Pilot in Operation". In *Journal of Physics: Conference Series*, volume 331, 6, p. 062040. IOP Publishing.

[34] A. Luckow, L. Lacinski, and S. Jha (2010) "SAGA BigJob: An Extensible and Interoperable Pilot-Job Abstraction for Distributed Applications and Systems". In *The 10th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 135–144.

[35] M. A. Rohrdanz, W. Zheng, M. Maggioni, and C. Clementi (2011) "Determination of reaction coordinates via locally scaled diffusion map". *J. Chem. Phys.*, **134**, 124116.

[36] E. Sherer, S. Harris, R. Soliva, M. Orozco, and C. Laughton (1999) "Molecular Dynamics Studies of DNA A-Tract Structure and Flexibility". *J. Amer. Chem. Soc.*, (212), 5981–5991.

[37] M. L. Mayer (2011) "Emerging models of glutamate receptor ion channel structure and function." *Structure*, **19** (10), 1370–80. doi:10.1016/j.str.2011.08.009.

[38] D. M. Schauder, O. Kuybeda, J. Zhang, K. Klymko, A. Bartesaghi, M. J. Borgnia, M. L. Mayer, and S. Subramaniam (2013) "Glutamate receptor desensitization is mediated by changes in quaternary structure of the ligand binding domain." *Proc. Natl. Acad. Sci. USA*, **110** (15), 5921–6. doi:10.1073/pnas.1217549110.

[39] K. Vance and N. Simorowski (2011) "Ligand-specific deactivation time course of GluN1/GluN2D NMDA receptors". *Nature Comm.*, pp. 1–29. doi:10.1038/ncomms1295.Ligand-specific.

[40] E. Karakas and H. Furukawa (2014) “Crystal structure of a heterotetrameric NMDA receptor ion channel”. *Science*, **344** (6187), 992–997. doi:10.1126/science.1251915.

[41] W. Zheng, M. A. Rohrdanz, M. Maggioni, and C. Clementi (2011) “Polymer reversal rate calculated via locally scaled diffusion map”. *J. Chem. Phys.*, **134** (14), 144109.

[42] W. Zheng, B. Qi, M. A. Rohrdanz, A. Caflisch, A. R. Dinner, and C. Clementi (2011) “Delineation of folding pathways of a β-sheet miniprotein”. *J. Phys. Chem. B*, **115** (44), 13065–13074.

[43] The SAGA Project, http://www.saga-project.org.

[44] “SAGA-Pilot”. http://radical-cybertools.github.io/saga-pilot/index.html.

[45] “RADICAL Cybertools”. http://radical-cybertools.github.io/.

[46] Advancing hydrometeorological prediction capabilities through standards-based cyberinfrastructure development: The community WRF-Hydro modeling system, David Gochis and Antonio Parodi and Rick Hooper and Shantenu Jha and Ilya Zaslavsky http://www.researchgate.net/publication/258775852_Advancing_hydrometeorological_prediction_capabilities_through_standards-based_cyberinfrastructure_development_The_community_WRF-Hydro_modeling_system.

[47] Standards-based publication and sharing of time series information in the DRIHM project: A EU-US collaboration, Richard Hooper and Ilya Zaslavsky and Antonio Parodi and David Gochis and Shantenu Jha and Thomas Whitenack and David Valentine and Olivier Caumont and Ljiljana Dekic and Marija Ivkovic and Luca Molini and Tatiana Bedrina and Peter J. A. Gijsbers and Erik de Rooij and Nicola Rebora, http://www.researchgate.net/publication/258782496_Standards-based_publication_and_sharing_of_time_series_information_in_the_DRIHM_project_a_EU-US_collaboration.

[48] B. K. Radak, M. Romanus, E. Gallicchio, T.-S. Lee, O. Weidner, N.-J. Deng, P. He, W. Dai, D. M. York, R. M. Levy, and S. Jha (2013) “A Framework for Flexible and Scalable Replica-exchange on Production Distributed CI”. In *Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery*, XSEDE ’13, pp. 26:1–26:8. ACM, New York, NY, USA.

[49] PanDA Beyond ATLAS : A Scalable Workload Management System For Data Intensive Science, M.Borodin and K.De and S.Jha, and D.Golubkov and A.Klimentov and T.Maeno and P.Nilsson and D.Oleynik and S.Panitkin and A.Petrosyan and A.Schovancova and A.Vaniachine and T.Wenaus, Invited Session on Big Data, International Symposium on Grids and Clouds (ISGC) 2014, Taipei, Taiwan.

[50] Hortonworks (2014). “Apache Hadoop YARN is a sub-project of Hadoop introduced in Hadoop 2.0”.

[51] J. Qiu and B. Zhang (2013) *Clustering Social Images with MapReduce and High Performance Collective Communication*. IOS Press.

[52] J. Qiu and B. Zhang (2014). “Harp: a runtime for efficient in-memory communication.”

[53] R. Mukherjee, A. Thota, H. Fujioka, T. C. Bishop, and S. Jha (2012) “Running many Molecular Dynamics Simulations on many Supercomputers”. In *Proceedings of the 1st Conference of the Extreme Science and Engineering Discovery Environment: Bridging from the eXtreme to the Campus and Beyond*, XSEDE ’12, pp. 2:1–2:9. ACM, New York, NY, USA.

[54] J. A. Smith, M. Romanus, P. K. Mantha, Y. El Khamra, T. C. Bishop, and S. Jha (2013) “Scalable Online Comparative Genomics of Mononucleosomes: A BigJob”. In *Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery*, XSEDE ’13, pp. 23:1–23:8. ACM, New York, NY, USA.

[55] T. Tu, C. Rendleman, D. Borhani, R. Dror, J. Gullingsrud, M. Jensen, J. Klepeis, P. Maragakis, P. Miller, K. Stafford, and D. Shaw (2008) “A scalable parallel framework for analyzing terascale molecular dynamics simulation trajectories”. In *High Performance Computing, Networking, Storage and Analysis, 2008. SC 2008. International Conference for*, pp. 1–12. 10.1109/SC.2008.5214715.

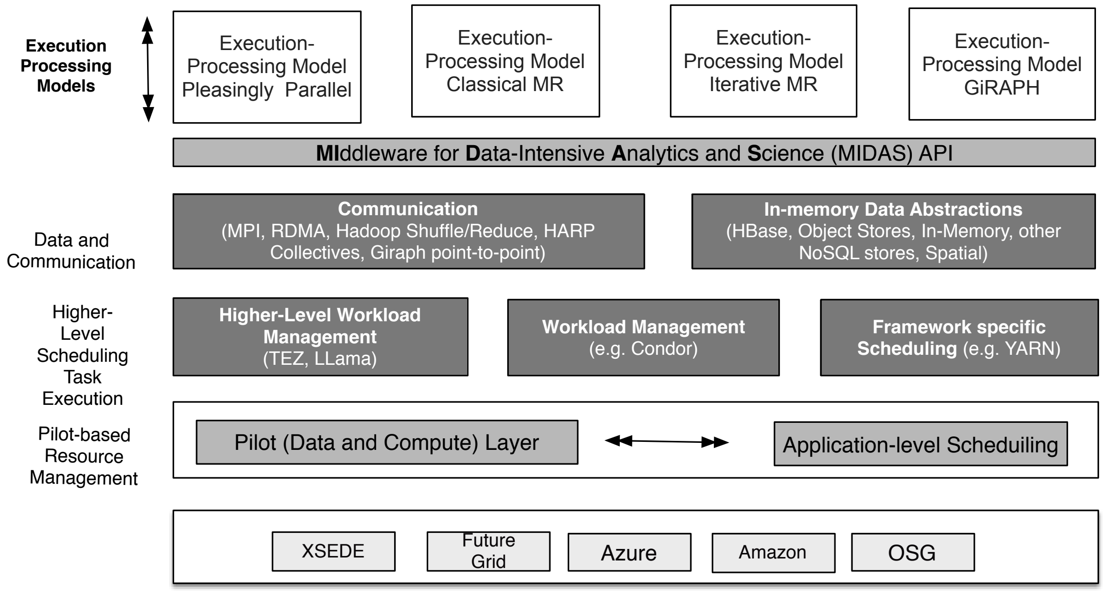
Execution Processing Models
Execution-Processing Model Pleasingly Parallel
Execution-Processing Model Classical MR
Execution-Processing Model Iterative MR
Execution-Processing Model GiRAPH
MIddleware for Data-Intensive Analytics and Science (MIDAS) API
Data and Communication
Communication (MPI, RDMA, Hadoop Shuffle/Reduce, HARP Collectives, Giraph point-to-point)
In-memory Data Abstractions (HBase, Object Stores, In-Memory, other NoSQL stores, Spatial)
Higher-Level Scheduling Task Execution
Higher-Level Workload Management (TEZ, LLama)
Workload Management (e.g. Condor)
Framework specific Scheduling (e.g. YARN)
Pilot-based Resource Management
Pilot (Data and Compute) Layer
Application-level Scheduiling
XSEDE
Future Grid
Azure
Amazon
OSG

Figure 3: MIDAS Layered Architecture View: The Pilot-Layer provides the basis for higher-level MIDAS abstractions supporting e.g. access to heterogeneous compute and data resources and in-memory caching for the iterative MapReduce programming model.

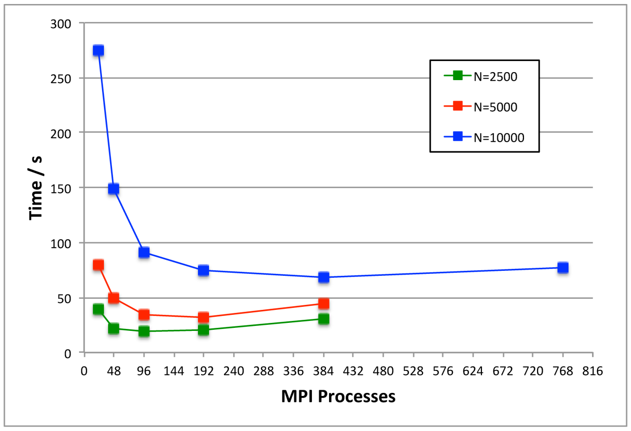

Figure 4: Strong scaling test data for LSDMap on ARCHER