Proposal: CUDA Center of Excellence at LSU

■ [Proposals should be in PDF format and should consist of no more than 5 pages motivation and description including budget]

1 Introduction

2 TODO LIST

```
Rephrase and restructure the intro to have small paras about:
a) hooks and goals \\
     a1) crisis in code development: \\
         nvidia has won the hardware battle;
         someone needs to develop the
         complex frameworks, algorithms and runtime systems
         avalable to a large user base\\
     a2) crisis in skilled workforce
         Leading national initiatives for creating
         Workforce in computational sciences.

b) why we are qualified to do that \\
       b1) have large international(PIRE), national(BLUEWATERS)
           and state-wide(LONI and LI) partnerships\\
       b2) varied applications\\
           lasigma(applications for materials research),
           cactus, frameworks, stellar, auto-code generators,

       b3) staff and experience:
           we have made investments\\
           outreach efforts are unique in the nation\\

c) payoff to nvidia \\
       c1) open new application areas
           comp. materials, grav. phy., \\
       c2) graduate users from serial to heterogeneous\\

d) personnel involved and research highlights
       d1) cactus
       d2) lasigma
       d3) hpx
       d4) pluto

e) GPU computing at LSU: Past and the future
       e1) previous efforts: spider, gpu@hpc
       e2) future: tezpur replacement

f) outreach vision:
       f1) courses: current and past
       f2) boot camps

g) budget : what we are asking for?
       g1) cash: workshops
```

```
g2) hardware: research cluster
g3) travel support
g4) personnel to help conduct workshops
g5) cash: future upgrade of gpu cluster
```

2.1 Goals

■ [hooks and goals] GPU acceleration is fast turning out to be the default route to exascale computing. The next generation supercomputers are increasingly employing several multi-core CPUs and GPUs on each node. Nvidia has already won the hardware battle and established CUDA as the prime candidate for software development on GPUs. Unfortunately, development of efficient scalable codes on these complex hybrid topology architectures, is still a great challenge to larger scientific comminuty. This has led to a mid-term crisis in software development which can only be resolved by dedicated collaborative effort between domain scientists and computer engineers. It also requires a relook at all aspects of development including frameworks, runtime systems, algorithms and applications, which worked well for sub-petaflop systems, but are not expected to be as productive in the future. The effort requires susbtantial training initiatives at state and national level, to develop a generation of upcoming users and developers.

■ [Why are we qualified?] LSU, with LONI and CCT has such centers which excel equally in research and outreach activities. LSU has been leading research activities and providing critical support to various state-wide research and education programs. LONI, LONI Institue and the recently funded LA-SiGMA are successful state wide programs which continue to provide excellent hard and soft resources to all state researchers, unequivocally. The Center of Computation and Technology at LSU has played a key role in these programs. We are involved in extensive outreach and education program ranging from boots camp for high school kids to international workshops and conferences. The research activities of various groups in Louisiana are strongly coupled to the LONI Institute and LASiGMA, which have large number of renowned researchers. It is the goal of LASiGMA and LONI Institute, to help graduate its researchers to the next generation supercomputers. We have excellent staff with immense experience in maintaining resources and carrying out outreach activities. Our outreach program, with a strong collaborative atmosphere, is unique in the nation.

■ [Payoff to Nvidia] To accomplish our goal of providing cutting edge research environment throughout Louisiana, we are keen to partner Nvidia. A CUDA Center of excellence at LSU would pave the way for next generation software development. There are ample reasons why NVIDIA should be excited by the prospect of a CUDA Center of Excellence at LSU. With an extensive research and outreach infrastructure, already in place, we can catapult the use of CUDA in various fields ranging from physics, chemistry, material sciences and many engineering domains. We shall discuss the ongoing research activities in the next section.

3 Research highlights

3.1 The Cactus CaKernel

Cactus is an international effort developing a general framework for programming parallel scientific applications. Using the framework approach, methodologies and best practices are captured in modules called thorns, allowing efficient programming via building blocks. Once GPU support is enabled via a thorn, it can be used without explicit knowledge of GPU programming. Cactus is heavily used at LSU in the development of the Numerical Relativity Tool Kit, but is by no means limited to this field. For instance there is a thorn which provides an adaptive mesh refinement driver that can be used in CFD programs.

3.2 Pluto Compiler Advancement

Pluto is a compiler designed to generate effective code for heterogeneous systems. It functions as a source-to-source transformation system that can generate code for general-purpose multicore systems as well as for GPUs. The input source is basically standard C code with a few added directives. Pluto is being enhanced to handle heterogeneous environments with multiple GPUs and many-core processors. In addition, we are exploring the use of user directives to cguide optimizations.

3.3 ParalleX

A new systems paradigm is being developed by the STE||AR group using the ParalleX execution model and its implementation in the experimental runtime system HPX (High Performance ParalleX). It is intended to improve the reliability and programmability of data-intensive massively parallel computing platforms using GPUs in a more user friendly way. The ParalleX notion of percolation provides for a perfect high level abstraction of massively data-parallel execution using GPUs. All applications built on top of HPX will immediately benefit from the results.

3.4 LA-SiGMA

LA-SiGMA is a statewide NSF funded computational materials science project involving nearly 50 faculty at 7 Louisiana campuses. LA-SiGMA has several computational teams developing a variety of computational materials science applications of which the most active team is the GPU team. Concentration is being placed on computational quantum chemistry approaches to modeling novel materials and their properties. It will also make use of existing community codes that are GPU-enabled, such as NAMD and LAMMPS.

4 Requested Support:

At this time, support is requested for:

- Approximately $250,000 for upgrade to the “Kepler” GPUs on Tezpur in 2014 to keep the GPUs on the production system at the state-of-the-art level.
- A small research cluster, code named Aragog, with GPUs available only to NVIDIA researchers to allow us to keep our researchers, trainers and their codes current on developing technology trends. Our codes and trainers will then be ready when we upgrade to the newest GPUs prior to development and testing on Shelob.
- Instructor support and funding for a multi-day workshop for researchers and LSU and LONI trainers on NVidia’s development technology so they can use Aragog
- Workshop support for a 5-day computational workshop each year for undergraduate students on Aragog. This would include NVIDIA trainers and funding support for catering and t-shirts.

5 Budget

The following table summarizes the support request for a 5-day workshop to be held on the LSU campus, based on 35 attendees:

Item (Cost/Person)	Day 1	Day 2	Day 3	Day 4	Day 5	Total
Breakfast ($8.00)	$280.00	$280.00	$280.00	$280.00	$280.00	$1,400.00
Morning Break ($4.50)	$157.50	$157.50	$157.50	$157.50	$157.50	$787.50
Lunch ($13.00)	$455.00	$455.00	$455.00	$455.00	$455.00	$2,275.00
Afternoon Snack ($4.50)	$157.50	$157.50	$157.50	$157.50	$157.50	$787.50
Total ($30.00)	$1,050.00	$1,050.00	$1,050.00	$1,050.00	$1,050.00	$5,250.00

Table 1: Workshop Support Budget

6 LSU History with GPUs

The researchers in LSU's CCT HPC group, members of Cactus, Pluto, STE||AR, and LA-SiGMA projects, and the investigators in the proposal are well qualified to meet these goals. The CCT has administered a (now five years old) GPU cluster, Spider, with 32 NVIDIA Tesla GPUs since 2005. The LSU HPC group administers 16 CPU clusters totaling over 110 TF including the TeraGrid Queen Bee resource, in addition to a 2-node GPU cluster with 6 Fermi M2070 GPUs. The CCT events team includes 4 full time staff who host a variety of computational sciences workshops. Last summer, the CCT hosted 12 workshops each lasting from several days to a week. The Cactus project provides a general parallel framework for a variety of scientific applications with short ranged correlations. This request will enable Cactus to add elements to the framework which enable GPU acceleration, aiding all who use Cactus to quickly derive performance benefits.

7 Outreach

■ [Brief intro to courses and workshops before you discuss them] HPC@LSU is the organization which provides services to the LSU, LONI (Louisiana Optical Network Initiative), and NSF XSEDE (formerly TeraGrid) HPC user community. In addition to operational support for 16 clusters at 7 sites around the state, it also provides practical training via tutorials, workshops, and a variety of on-line resources. The LSU Center of Computation and Technology (CCT), provides tier 3 level support from expert users, and member faculty teach formal courses in computational science.

7.1 Description of courses

Distance learning graduate level classes are shared via synchronous video and are available to students throughout the state. They are, for the most part, specialized graduate courses that emphasize computational methods that could not be taught at the individual institutions due to the lack of a critical mass. Last year, nine graduate courses were offered, in many cases involving faculty from different locations:

- *Computational Physics*: Computing for Petascale Systems by Karen Tomko (Ohio Supercomputer Center) and Juana Moreno (LSU).
- *Simulations of Quantum Many-Body Systems* by Matthias Troyer (ETH Zurich) and faculty at the Ludwig-Maximilians-Universität, Munich; École normale supérieure, Paris; Max Planck Institute for the Physics of Complex systems, Dresden; University of Massachusetts, Amherst; University of Wyoming; and Louisiana State University (Co-PI Jarrell).
- *Advanced Solid State Physics with Computation* by Co-PI Mark Jarrell (LSU).
- *Computational Solid State Physics* by Rongying Jin (LSU) and Cyrill Slezak (Hillsdale College, Michigan).
- *Introduction to Scientific Computing* by Gabrielle Allen (LSU, NSF), and Frank Löffler (LSU).
- *GPU Programming* by SI David Koppelman (LSU), Programming GPUs for graphics and non-graphical computations using OpenGL, OpenGL Shader Language, and CUDA,
- *GPU Microarchitecture* by SI David Koppelman (LSU), Low-level design and programming of GPUs, including organization, memory hierarchy, and native instruction sets, with examples.
- *Performance Optimization of Applications on GPUs* by Co-PI Ramanujam (LSU): Optimization of science applications, GPUs, heterogeneous systems, CUDA, OpenCL
- *Program Parallelization* by Co-PI Ramanujam (LSU): Compiler optimizations, source-to-source transformations. Programming paradigms and support for HPC architectures.

This fall we have added two more courses:

- *Condensed Matter Physics* by Adrienn Ruzsinszky (Tulane).
- *Statistical Mechanics* by SI Randall Hall (LSU) and Colin Wick (LATech).

7.2 Training activities

Training and education at all levels, from primary school through graduate school and beyond, is an essential component of this project. Programs are offered both to train a core set of users, who will develop codes for Tezpur and Aragog, and for the next generation of national level heterogeneous machines such as Forge and Titan. The intent is to develop an expertise pipeline to sustain CCOE@LSU in the future. The Aragog will be used for summer workshops to train students to use PGI GPU accelerator compilers and CUDA. GPUs will be introduced as components of the machines built and used by high school students in our long-running Beowulf bootcamp. To help establish a pipeline into our graduate programs, the Tezpur cluster will be used by undergraduate students in the CCT and LA-SiGMA REU programs. To establish a pipeline into our undergraduate programs, K-12 local teachers participating in LA-SIGMA summer RET programs will also use the cluster.

Beowulf/GPU Bootcamp. For the past four years the CCT has offered a week-long camp to introduce high school students to the field of supercomputing. In this camp students are given hands-on experience with all aspects of supercomputing, from the hardware to the applications level. They interact with research professors and learn about new career paths.

During the week, students take apart and reassemble compute nodes, install the operating system and system software, and learn to program their new cluster with Python + MPI. For many students, this will be their first exposure to programming, for others it is a chance to learn about parallel programming and its application to science. Students learn about different forms of parallelism at the bootcamp, and in part this activity is carried out with an interactive calculation of vector addition or the heat equation. Students also hear lectures from domain scientists, describing research of special interest to the Louisiana area, from hurricane forecasting to gravitational wave signals (i.e. the LIGO observatory in Livingston Parish).

This year we will update the program by including a discussion of GPUs and their importance in modern computing. The students will not only take apart and reassemble GPU hardware similar to what will be in Tezpur, but will explore simple data parallel programming of GPUs from within Python. Students will run their programs on Tezpur, and gain experience about modern scientific research. Students also will be taken on a tour of the machine room facilities and see Tezpur firsthand.

Research Experiences for Undergraduates (REU). Both LA-SiGMA and the CCT offer REU programs sponsored by the NSF. They are nine-week programs where students work collaboratively on a wide variety of computational science projects, and become familiar with interdisciplinary research. The LA-SiGMA REU program recently completed its first year. It involves 30 students every summer, 5 students at each of the following Louisiana institutions: Louisiana State University, Louisiana Tech University, Southern University, Tulane University, University of New Orleans, and Xavier University. The CCT REU is in its second year. During the summer of 2010 its NSF funding was complemented with additional funds from the Board of Regents to support 15 college students from all over the States. Another 10 students participated in the program this last summer. Three high school students also joined the REU programs last summer.

Students work with the faculty and research staff at CCT and LA-SiGMA, learning how to use the cutting-edge cyberinfrastructure on campus to examine various science phenomena such as gravitational waves that result from colliding black holes, explore new materials for energy storage or revolutionary electronic devices, or to design new kinds of physical interaction devices to extend computer visualizations. Since most research groups at the CCT collaborate with international researchers, REU students are exposed to how international collaborations work. Last year, nearly half of these students worked with the GPU team using the most current cyberinfrastructure tools with individually designed training sessions targeted to their specific degree of preparation. These students will be able to develop their codes on the Aragog GPU cluster, and participate in summer workshops (see below) which will employ the cluster to teach heterogeneous programming and parallel throughput. We hope that these students will return to LSU for graduate school so that they can continue to develop their codes and deploy them on future heterogeneous supercomputers.

The last two summers, the visit of Edward Seidel, Assistant Director for Mathematical and Physical Sciences at NSF and previous Director of the CCT, was a highlight of the program. Seidel participated in the final presentations of the REU students and addressed all REU students on campus in the evening both years.

Research Experiences for Teachers (RET) programs associated with the LA-SiGMA project are offered annually in New Orleans, Baton Rouge, and Ruston. For example, in the summer of 2011, three teachers were selected for the LSU's program, three for Southern University's, two for UNO's, and five for LA Tech/GSU's program. These teachers work at a mix of public and private schools. Three of the teachers are faculty members at the Louisiana School for Math, Science, and the Arts (LSMSA). LSMSA is the state's flagship high school, a boarding school for gifted and talented grades 10-12 Louisiana students. LSMSA sends the majority of its graduates to LA universities, mostly to LSU. Two of the LSMSA faculty were members and active participants in the GPU team last summer. They developed classroom demonstration modules, and they are using the present set of GPU workstations. These LSMSA faculty have committed to return next summer and help us to extend this program to other LA high school students.

Computational Sciences Workshops are run by the CCT continuously throughout the summer. Last summer, we held 12 workshops on a broad range of subjects. These included two workshops from the Virtual School of Computational Science and Engineering (one on GPU programming and the other in MPI/OpenMP), introductory workshops on ParalleX and HPX, a Shodor workshop on Computational Thinking from a Parallel Perspective, a week long hands-on workshop on DFT methods, two one-week workshop for middle school girls (Alice in Computerland), and several others. In addition, intensive training for the REU programs discussed above involves workshop in basic Unix/Linux, C and C++ programming, MPI, OpenMP, CUDA, an introduction to Cactus, etc. which are open to non-REU students.

HPC training is provided throughout the semester, open to anyone at LSU and across the state. It not only includes tutorials on HPC, MPI, OpenMP, Parallel programming concepts, debugging with TotalView and DDT, grid computing, the Cactus framework, CUDA programming, Shell scripting, but also science-specific tutorials, e.g., on "Electronic Structure Calculations in Quantum Chemistry" and "Scientific Workflow and Visualization in VisTrails". Aragog will have a huge impact on these tutorials, courses and workshops. All of the distance learning courses listed above involve computational projects, which will utilize the cluster. It will also be heavily utilized in the workshops. Furthermore, since Aragog will be under local control it can be "reserved" to meet the needs of these activities without disrupting the production systems.

Proposal: CUDA Center of Excellence at LSU Suppliment

8 GPU research activities at LSU

■ [up to 5 pages supplementary material detailing previous or envisioned research including references, and a biographical sketch or condensed curriculum vita, which must be no more than 2 pages per principal investigator.]

8.1 Cactus Framework on Heterogeneous Systems

Despite the growing trend to make GPUs available for supercomputing, there is a paucity of tools to fully exploit the hardware's potential. To address this issue, a team at LSU CCT, in collaboration with researchers from Poznan Supercomputing and Networking Center in Poland designed and implemented a programming framework, *CaKernel*. This framework supports development of large scale scientific applications which are primarily using stencil computations in their numerical kernels.

The Cactus computational framework is the foundation of CaKernel. Cactus [**? ? ?**], is an open-source, modular, highly portable, programming environment for collaborative research using high-performance computing. Cactus is distributed with a generic parallel computational toolkit providing parallelization, domain decomposition, coordinates, boundary conditions, interpolators, reduction operators, and efficient I/O in different data formats. There are at least 30 groups worldwide using Cactus for their research work in cosmology, astrophysics, computational fluid dynamics, coastal modeling, quantum gravity etc. The Cactus framework is a vital part of the Einstein Toolkit, an NSF-funded (CIGR) collaboration which enables large parts of the world-wide research in numerical relativity by providing the necessary core computational tools. Cactus is part of the software development effort for Blue Waters, and in particular the Cactus team is working with NCSA to produce development interfaces and paradigms for large scale simulation development. Through the LONI's HPCOPS (TeraGrid) award, the Cactus group is developing a Science Gateway which will include capabilities for simulation assembly, compilation and deployment across the NSF TeraGrid/XSEDE resources. Through the NSF EPSCOR CyberTools project, the Cactus team is involved in educational initiatives including an REU, K-12 education and graduate mentoring.

Automated Code Generation is an important technique to bridge the gap between a physics equation and its discretization on one hand, and an efficient implementation in a low-level language such as C, C++, CUDA or OpenCL on the other hand. We employ *Kranc* [**? ?**], which accepts a set of equations and a discretization specification in Mathematica syntax, and emits complete, self-contained Cactus modules. Kranc applies various transformations and optimizations (e.g. loop blocking, loop fission, multi-threading, loop unrolling, vectorization) as necessary for the target architecture. As Kranc's input describes the physics system at a very high level, it can perform optimizations that are inaccessible to a low-level language. In the Einstein Toolkit [**?**], Kranc is used to generate a highly efficient open-source implementation of the Einstein equations, a complex system of equations with more than 5,000 terms in its kernel. Current Kranc development includes using OpenCL as target language.

The CaKernel programming framework, although still in the development stage, addresses the following challenges facing the computational science community: portability, programmability, performance, and productivity. CaKernel has been designed and implemented in Cactus as a programming abstraction to develop highly efficient parallel scientific applications which can take advantage of heterogeneous architectures. Due to the flexibility and extensibility of the Cactus framework no changes to the Cactus flesh are necessary, guaranteeing that existing features and user-implemented modules are not adversely affected by this addition. In particular, the parallelization across GPUs is handled in Cactus via MPI without any modifications to the existing framework. Thanks to Cactus, the kernel code written in CaKernel are guaranteed to be portable on almost all supercomputers available for research. With the combination of a set of highly optimized templates and a kernel descriptor, CaKernel can automatically generate computational kernels in CUDA, OpenCL, C, and potentially other programming languages from developer's C-style numerical codes. The programmability of CaKernel is greatly improved via the automatic code generation enabled in CaKernel to reduce possible human errors in writing code, with the ever-growing complexity of scientific codes in mind. CaKernel not only makes it easy to write and execute computational kernels, but also makes it possible to optimize the kernel without changing the kernel code itself. The performance can be fine-tuned by modifying/swapping the templates or adjusting the kernel parameters. These highly optimized templates that are completely transparent to application developers hide the complexity of lower-level programming and optimization. This design enables software engineers or domain experts to contribute their expertise at the maximum scale, thus enhancing the overall programming productivity.

The Development of CaKernel has so far been carried out on a 8-node GPU cluster with 2 NVIDIA Tesla C1060 cards in each node. Due to complexity and wide scope of potential application codes, there is still a lot of research to be done before CaKernel reaches a level of maturity where good performance can be obtained on a regular basis with a minimum of effort. We are therefore in great need of a new quick turnaround GPU system that is big enough to enable routine performance and scalability tests. The new hardware architecture and development environment on the CCOE@LSU cluster will also enable us to prepare our researchers for the existing and planned national level GPU clusters, e.g., Forge and Titan.

8.2 LASiGMA: Louisiana Alliance for Simulation-Guided Materials Applications

The Louisiana Alliance for Simulation-Guided Materials Applications (LA-SiGMA) [**?**] is an NSF-sponsored virtual organization of seven state institutions of higher learning focusing on computational materials science. Members include nearly 50 faculty, 9 postdocs and nearly 70 students from Tulane, Xavier, University of new Orleans (UNO), LSU, Southern, Louisiana Tech, and Grambling. Three members (Xavier, Southern and Grambling) are historically black colleges or universities. This group is composed of three science drivers (SDs) using a set of common computational methods, combined with experimental studies, to study electronic and magnetic materials (SD 1), energy storage and generation materials (SD 2), and biomolecular materials for drug delivery (SD 3). The computer codes used range from those written by individual research groups to open source and commercial software packages. The goals of the electronic and magnetic materials SD are to develop and validate methods that enable the study of complex phenomena in correlated electronic and magnetic materials ranging from transition metal oxides to organic magnets. As a consequence of their exotic and diverse physical properties, transition metal oxides are considered to be the frontier of research on "emergent research device materials." The targets of the energy materials SD include electrode materials for supercapacitors, hydrogen storage materials, and catalytic process for the formation of biofuels. The goals of the biomolecular materials SD are to develop, apply, and validate experimentally multi-scale computational tools that will enable the design of novel drug delivery vehicles.

The glue that holds the three SDs together are the formalisms, algorithms and codes being developed in this project. The cybertools and cyberinfrastructure (CTCI) group of participants, led now by Co-PI Ramanujam, allows Alliance members to more efficiently utilize the next generation of 21st-century supercomputers. The SD teams work hand-in-hand with computer scientists, computer engineers, and applied mathematicians in CTCI to develop the formalisms, algorithms and codes needed to efficiently utilize the next generation of supercomputers. Members of the SDs together with CTCI are broken into several computational teams to address common codes and high-performance computing challenges.

The largest of these computations teams is the GPU team, composed of over 30 LA-SiGMA students, postdocs, and faculty from 4 LA-SiGMA campuses working together to develop GPU accelerated codes. The GPU team is developing a series of GPU-enhanced codes that are heavily used in the community, including Hirsch-Fye Quantum and continuous time Monte Carlo, Density Matrix Renormalization Group, parallel tempering for glasses and molecular systems, Monte Carlo Simulations in Open Ensembles, and Variational Monte Carlo. Each code release will be available on our web portal and be accompanied by documentation and completely worked examples. In addition, the LA-SiGMA GPU team is coordinating merging of continuum simulation codes with the LAMMPS molecular dynamics code. Highly tuned code elements that have more general applicability, such as the thin rectangular matrix product and update needed by some of the Monte Carlo codes, will be released on their own as separate code modules. The development versions of all the codes will be continuously available through anonymous SVN access. Last summer many of the LA-SiGMA and CCT REU students and 5 faculty and students from the Louisiana School for Math, Sciences and the Arts (LSMSA) worked with the GPU team (c.f. Sec. **??**).

Collaborations between LA-SiGMA students and post-docs are enhanced through an innovative Computational Science Collaboratorium which brings together CTCI, GPU team, and members of all the SD's and computational teams into the same space with resources for the development of the next generation of codes. The collaboratorium is located in the CCT on the LSU campus. This space contains cubicles for 15 LA-SiGMA students or post-docs in computer science, physics, chemistry, mechanical engineering, electrical and computer engineering, etc. There is a separate space for faculty and staff across the hall. The collaboratorium has a meeting area with three white boards for discussion as well as modern teleconferencing and videoconferencing equipment so that the meetings may be shared with other campuses. Each desk in the collaboratorium has a high-performance Intel Westmere desktop computer, each with a NVIDIA Fermi GPU card. In addition there is a 1TF multi-core workstation, with a Fermi M2070 GPU. These desktops and workstation are being used for the development of GPU accelerated codes. In addition to the Intel compilers, MPICH, and OpenMP, each of these machines also has PGI GPU accelerator compilers installed. The PGI compilers simplify the development of GPU codes, by allowing the programmer to add OpenMP-like compiler directives to existing Fortran or C programs that control the code execution on the GPU. This allows codes to be developed and tested quickly without the need CUDA, at least in the initial stages of development.

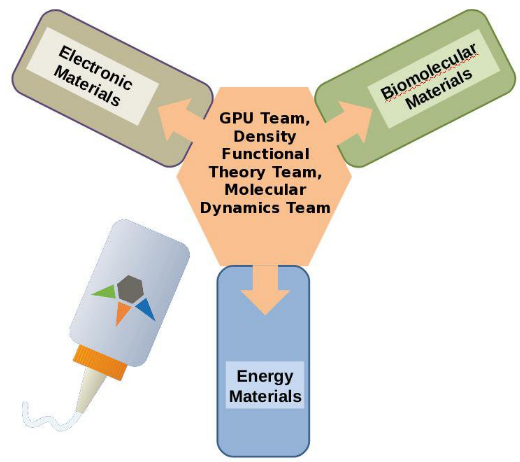

Figure 1: LA-SiGMA is composed of three Science Driver teams. The SD teams are glued together by the computational teams, the largest of which is the GPU team.

The impact of the CCOE@LSU project on the LA-SiGMA collaboration will be significant. One of the main goals of LA-SIGMA is to graduate its researchers from serial and parallel computing to the heterogeneous computing which will characterize the national leadership class capability computing in 2-3 years. The CCOE@LSU cluster will help us meet this goal, both as a machine we can use to design and test the codes being developed by the LA-SIGMA computational teams and as a machine which can be used in LA-SiGMA courses and workshops described in Sec. **??** Finally, and most significantly, the LA-SiGMA computational teams is developing a series of GPU-enhanced codes that are heavily used in the community. Each code release will be available on our web portal and be accompanied by documentation and completely worked examples.

8.3 Compilation Tehcnology—Pluto compiler

As mentioned earlier, the emergence of GPUs as cost-effective and powerful processors for general-purpose computing has raised great interest in their use for many scientific and engineering applications but further exacerbate the software development challenges. Developing high-performance applications for GPUs and accelerators is even more complicated than programming general-purpose multicore processors. Further, there is currently no way of writing portable code that can execute on an NVIDIA GPU or an AMD/ATI GPU or the Cell processor – each has its own programming model. The need for a portable programming model to program GPUs/Accelerators prompted an industry-wide consortium to come together and create the OpenCL language as an extension to C. The Pluto compiler builds on this approach, seeking to mask the difficulties of multi-core and GPU programming [**? ? ? ? ? ? ?**].

We plan to develop a number of new optimizations aimed at exploiting the different memories on the GPUs and aimed at optimizing communication between the CPU and the GPU. In addition, we will develop optimizations that allow the CPU and the GPU to work on partitioned computations rather than have the

CPU offload work to the GPU. The proposed work is motivated by recent advances in *polyhedral-based approaches* for powerful transformations of affine computations which have enabled the development of Pluto's automatic parallelization/optimization system [**?** **?**]. In addition, Pluto will be enhanced to explore automatic code generation for heterogeneous systems with multiple GPUs, where the GPUs can communicate directly (e.g., using the NVIDIA GPUDirect technology).

The proposed enhancements will be applicable over a broad range of scientific and engineering applications. For example, in doing these enhancements, we will work closely with application developers from the LA-SiGMA project. The CCOE@LSU cluster will be critical for compiler development, testing and training users to develop "performance programming" skills on next-generation heterogeneous systems. Such a system could result in significant productivity gains, and could improve the return-on-investment for existing and future hardware. As was done with the Pluto system, the OpenCL transformation system will be made publicly available.

8.4 The STE||AR Group

STE||AR stands for "Systems Technologies, Emergent Parallelism, and Algorithms Research" [**?**]. The centerpiece of this group's work is the ParalleX execution model and its implementation in the experimental runtime system HPX (High Performance ParalleX) [**?**]. The goal is to enable scientists and developers to write applications which expose more parallelism, scale better, and generally show improved performance compared to more conventional programming models such as MPI [**?**].

ParalleX is a new (and still experimental) parallel execution model. It's purpose is to overcome the limitations imposed by current hardware and software development practice. ParalleX synthesizes several ideas such as lightweight synchronization semantic constructs supporting message-driven computation in a global address space. The idea is to mitigate overheads, expose parallelism, and provide runtime adaptive scheduling and load balancing. It targets two types of applications. The first type includes those requiring excellent strong scaling, allowing for a dramatic reduction of execution time for fixed workloads. The second type covers those needing the highest levels of sustained performance through massive parallelism. Such applications are presently unable (through conventional practices) to effectively exploit more than a relatively small number of cores in a multi-core system. Such applications are unlikely to utilize high-end computing systems with the hundreds of millions of cores expected by the end of this decade.

Four factors are inhibiting these two forms of scalability: 1) starvation – the insufficiency of available useful work either globally or locally; 2) latency – the distance measured in time (e.g., cycles) for a remote access or service request; 3) overhead – the critical-time work required to manage parallel resources and concurrent tasks which would not be required for pure sequential execution; and 4) waiting – caused by contention or delays due to conflicts for shared physical or logical resources. It is recognized that each of these is a consequence of the fundamental strategy implicit in the execution model.

HPX (High Performance ParalleX) is an experimental, modular, and performance oriented representation of the ParalleX execution model targeted at conventional architectures (currently, Linux based systems, such as SMP nodes and conventional clusters) [**?**]. The most important design objective of HPX is to create a state-of-the-art parallel runtime system providing a solid foundation for highly scalable applications while remaining as efficient, as portable, and as modular as possible.

The overarching goal of using HPX in this project is to conduct fundamental research that will result in formal techniques and tools to enable the development of correct programs for data-intensive parallel computing. A major goal is to dramatically accelerate data intensive applications on future platforms of unprecedented scale, comprising multi/many core processor components, deep memory hierarchies and GPU based heterogeneous architectures. While we do not expect to provide all answers to the crisis looming in data-intensive/parallel computing, we plan to address the following questions in this project:

1. How can we reliably program data-intensive computing platforms based on GPUs to exploit massive parallelism and to serve best the varied tasks that may be executed on them?
2. How can we express high-level parallelism at this scale in a natural way for users?

3. What new programming abstractions (including models, languages and algorithms) can accentuate these fundamental capabilities?
4. How can programming systems for data-intensive computing platforms using GPUs be designed to support extremely high levels of concurrency, reliability, efficiency, and availability?

The ParalleX execution model exposes the interaction with and integration of accelerator hardware such as GPUs through one of its intrinsic models - percolation. Percolation utilizes the work queue model to minimize the number of system memory accesses which would have to be otherwise initiated by the GPUs to obtain new work with related operand data. This alleviates unnecessary processor stalls and enables the GPUs to utilize their internal execution resources optimally. Percolation allows to fully overlap and synchronize the computations in the GPU and the main cores by combining code and data into special ParalleX parcels (messages), which are being sent to the GPU for execution. This high level abstraction allows to integrate and fully parallelize the accelerated computation with the message driven, split-phase transaction execution on the main cores.

Our choice of ParalleX as the target platform and HPX as the implementation framework for our planned research is motivated by the key challenges of scalability, efficiency, power consumption, and reliability as well as programmability. We expect these challenges to dominate data-intensive systems towards the end of the decade, distinguishing them from the conventional platforms and enabling ultra-scale performance and capacity.

9 Other development efforts

9.1 Coastal Modeling

Coastal engineers and scientists have utilized GPU's in modeling the impacts of storm surges and water waves on civil infrastructures, such as levees, bridges, piers, and oil and gas drilling rigs. Applications include the 3D CFD model using the volume of fluid method for free surface flows and waves in the Cactus computational framework, and the Smooth Particle Hydrodynamic model for wave and structure interaction [**?**]. The LSU Department of Civil and Environmental Engineering has also developed Lattice Boltzmann models for shallow water flow and mass transport associated with dam break or levee breach using GPUs [**?** **?**]. The CCOE@LSU cluster will enable coastal researchers at LSU to address rich and urgent scientific problems pertinent to Louisiana hurricane protection and coastal restoration.

9.2 Next-Generation GPU Project

Since the release of the NVIDIA G80 in 2006 [**?**] and with the rapid acceptance of CUDA with which the G80 could be programmed for non-graphical purposes, the needs of the scientific computing community have joined those of 3D graphics developers in driving the designs of GPUs. Subsequent GPU generations have increased their usefulness for scientific computing by refining useful features such as higher double-precision throughput, more flexible synchronization capabilities, and greater memory hierarchy flexibility [**?**]. These newer features focus on kernel performance itself, other new features of recent designs help improve GPU utilization by allowing multiple active kernels. Many proposed GPU enhancements are based on a system view in which the CPU plays only a supporting role [**?** **?**]. Still, there are some problems which have some elements that perform well on GPUs, and other elements which execute more efficiently on CPUs. When the interaction between these elements is fine-grained, efficiency on hybrid GPU/CPU may suffer from the costs of data movement. Next generation GPUs will need to mitigate such problems.

The Next-Generation GPU Project led by SI Koppelman will investigate GPU and CPU designs which support very tight coupling between GPU and CPU to support hybrid execution. The starting point for this investigation will be a set of well tuned codes with deft GPU/CPU work divisions running on state-of-the-art production-scale hardware. This exactly describes the CCOE@LSU cluster and the codes to be developed for it. The Next-Generation GPU Project will leverage this experience by developing candidate next-generation GPU designs.

The project will look closely at which GPU and CPU features are useful for tightly coupled hybrid codes.

Early programming techniques initially concentrated on shifting work to the GPU with little consideration given to CPU participation beyond shuttling data [? ? ?]. The emphasis is now shifting to problems that require, or at least can benefit from, a more careful division of work between the devices, see for example [?]. For some codes work division is complex as it must balance the benefit of execution efficiency on the preferred processor with the overhead of data migration. It is important to understand how the division is influenced by such things as data layout, synchronization, etc. Achieving this understanding will not just produce better tuned hybrid codes, but it will also provide hints for features to include in follow-on generations of GPUs.

The list of candidate features for such a study includes memory organization and bandwidth provisioning, interaction with low-level caches, and GPU scheduling flexibility. There are complementary issues for the CPU, including performance and capabilities of the vector (SIMD) unit and cache organizations. To go beyond these features one needs extensive experience with tightly integrated GPU/CPU codes, such codes are currently being developed by the LA-SiGMA, CaKernel (Cactus), and STE||AR (HPX) projects, and development would move to the CCOE@LSU machine described in this proposal.

With the help of domain and computational experts the codes will be factored and tuned for execution on CCOE@LSU. In the process coding alternatives will be developed that exercise some of the system's features, and their performance will be carefully measured. This exploration process will provide us with well-tuned codes and the know-how for porting additional codes. It will also inevitably lead to ideas on how next-generation GPU/CPU compute clusters might be designed, as we find places where small additional or modified capabilities would lead to much better performance. These ideas will be pursued with simulation and other studies of next-generation GPUs.

Also, by exploring these issues in depth on the CCOE@LSU cluster, programming methodologies can be developed which will contribute to the other elements of the project. Exposure to these issues will raise the experience levels of participating students and directly help support future systems.

9.3 Geometric and Visual Computing

The CCOE@LSU cluster will be very useful to the geometric and visual computing group in LSU led by SI Li, especially on the digital archiving, modeling, and processing of massive geometric spatial and temporal data, such as urban scanning, architectural, archaeological, forensic scan data, and medical imaging data. The proposed heterogeneous computing environment can provide efficient data processing toolkits in facilitating and augmenting many important late-stage large scale geometric data processing tasks in information modeling and analysis.

The computational cost of processing large-scale geometric data is so great as to be prohibitive in many cases. We are studying efficient domain decomposition techniques that partition the mapping and modeling problems into solvable sub-domains, and we are also developing processing algorithms and optimization solvers accordingly. The CCOE@LSU cluster will play an important role in speeding up several tasks whose realtime or interactive response are critical.

A few closely related projects include:

1. Massive Heterogeneous Volumetric Data Mapping for Urban Modeling and Environment Exploration: efficient mapping and matching the raw data collected by vehicles/robots equipped with laser scanners in a dynamical environment;
2. Digital Forensic Facial Reconstruction: automating the assembly, completion, modeling of excavated damaged/fragmented skeletons/skulls (and other archaeological data) in a digital environment with real-time semantic manipulations/feedback from domain experts;
3. MRI-guided Four-dimensional Lung Tumor Motion Modeling and Radiotherapy Management: tumor movement prediction and temporal parametric motion model refinement in real-time during radiotherapy delivery enabling correct targetting of the tumor by the radiation beam, and minimizing the damage healthy tissue.

All these projects require efficient parallel processing and can greatly benefit from the proposed CUDA center of excellence at LSU.

Personnel Involved in Project:

1. Honggao Liu; Louisiana State University; PI
2. Mark S. Jarrell; Louisiana State University; Co-PI
3. Hartmut Kaiser; Louisiana State University; Co-PI
4. Steven Brandt; Louisiana State University; Co-PI
5. Jagannathan Ramanujam; Louisiana State University; Co-PI
6. Qin Chen; Louisiana State University; SI
7. Randall W. Hall; Louisiana State University; SI
8. James A. Lupo; Louisiana State University; SI
9. David Koppelman; Louisiana State University; SI
10. Xin Li; Louisiana State University; SI
11. Juana Moreno; Louisiana State University; SI
12. Mayank Tyagi; Louisiana State University; SI
13. Jian Tao; Louisiana State University; SI
14. Peter Diener; Louisiana State University; SI
15. Frank Löffler; Louisiana State University; SI
16. Ravi Paruchuri; Louisiana State University; SI
17. Bhupender Thakur; Louisiana State University; SI

References