For the problem of extracting semantics from instructional videos, we collected a large cooking video dataset, YouCook2, by querying YouTube. The videos were then annotated with sentence descriptions, as well as timestamps denoting the particular cooking styles, such as grilling and frying. As a first step, we generated an intermediate video representation by grouping similar pixels in space and time in a video using our streaming hierarchical video segmentation method [7]. We further extracted the semantic entities, such as objects and actions, from videos using random field modeling [8]. Our methods achieve much better results than the previous state of the art in the field.

For the problem of deep reinforcement learning, we have set up a simulation in the robot simulator Gazebo with wooden blocks and a robotic arm with two degrees of freedom. For the problem of accelerating DNN computation, we investigated custom data representations and ran simulations on how they can be used to speed up DNN computation through new hardware design.

WHY BLUE WATERS

Running DNNs and processing video require intensive parallel computation on both CPUs and GPUs. In order to perform thorough experiments, we need access to a large number of CPUs and GPUs. This makes Blue Waters **essential** for our research. At the peak of our usage, we were able to use a large number of CPUs and GPUs concurrently, which allowed us to process a large amount of video and explore a large design space of models. In addition, we received timely help from the Blue Waters team, which was also **indispensable** for our project.

NEXT GENERATION WORK

We hope to use Blue Waters to continue advancing video understanding, including more accurate understanding of human actions, deeper semantics from instructional videos, and faster DNN computation.

ALGORITHMS FOR EXTREME SCALE SYSTEMS

Allocation: Illinois/100 Knh
PI: William Gropp[1]
Collaborator: Luke Olson[1]

[1]University of Illinois at Urbana-Champaign

EXECUTIVE SUMMARY

Continued performance enhancement of large-scale computer systems will come from greater parallelism at all levels. At the node level, this is seen in the increasing number of cores per processor and the use of large numbers of simpler computing elements in general purpose graphics processing units (GPGPUs). The largest systems must network tens of thousands of nodes together to achieve the performance required for the most challenging computations. Successful use of these systems requires new algorithms. Over the last year, we have shown the benefit of lightweight intranode balancing on scalability and performance. We continue to explore alternative formulations of conjugate gradient that eliminate some of the strict barrier synchronization and better use memory hierarchy, ways to reduce the impact of communication on the scalability of algebraic multigrid, and algorithmic approaches to resilience that exploit the multilevel representation in multigrid methodology.

INTRODUCTION

At extreme scale, even small inefficiencies can cascade to limit the overall efficiency of an application. New algorithms and programming approaches are needed to address barriers to performance. This work directly targets current barriers for effective use of extreme scale systems by applications. For example, Krylov methods such as conjugate gradient are used in many applications currently being run on Blue Waters (MILC code is one well-known example). Developing and demonstrating a more scalable version of this algorithm would immediately benefit those applications. In the longer term, the techniques that are developed will provide guidance for the development of highly scalable applications.

METHODS & RESULTS

Early results with alternative Krylov formulations have revealed several performance effects that can provide a factor of two or greater improvement in performance at scale. Current work has been limited by the fact that the non-blocking MPI_Allreduce on Blue Waters is functional but does not provide the expected (or perhaps hoped for) performance, particularly regarding the ability to overlap the Allreduce operation with other communication and computation. However, even with this limitation, we have seen a benefit in using non-blocking collective operations regarding a reduction in the sensitivity of the application to performance jitter and other irregularities.

WHY BLUE WATERS

Scalability research relies on the ability to run experiments at large scale, requiring tens of thousands of nodes and hundreds of thousands of processes and cores. Blue Waters provides one of the few available environments where such large-scale experiments can be run. In addition, only Blue Waters provides a highly capable I/O system, which we will use in developing improved approaches to extreme-scale I/O.

NEXT GENERATION WORK

We expect the next generation systems to rely on many more cores per node and to use different network topologies compared to Blue Waters. Additionally, there are likely opportunities to exploit new network capabilities and provide algorithms and programming systems adapted to the new memory, processor, and interconnect architectures.

PUBLICATIONS AND DATA SETS

Kale, V., Low-overhead scheduling for improving performance of scientific applications, Ph.D. Thesis, Spring 2015.

Eller, P., and W. Gropp, Exploiting nonblocking collective operations in conjugate gradient, *Copper Mountain Meeting,* 2015.

Bienz, A., Analyzing the performance of a sparse matrix vector multiply for extreme scale computers, *SC15*, Austin, TX, Nov. 2015.

Eller, P., Non-blocking preconditioned conjugate gradient methods for extreme-scale computing, *SC15*, Austin, TX, Nov. 2015.