EFFICIENT, SCALABLE AND FAULT TOLERANT GENOMICS PIPELINE

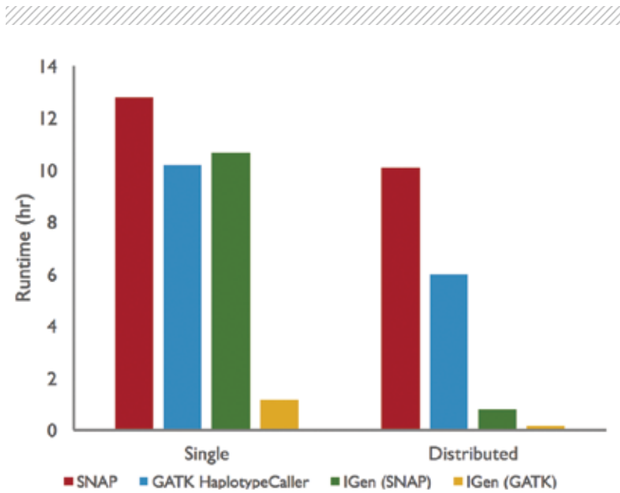
ALLOCATION: Illinois/50.0 Knh
PI: Ravishankar K. Iyer[1]
Co-PI: Zbigniew Kalbarczyk[1]
Collaborators: Saurabh Jha[1], Subho Banerjee[1], Phuong Cao[1], Valerio Formicola[1], and Hao Jin[1]

[1]University of Illinois at Urbana-Champaign

EXECUTIVE SUMMARY

Our overarching investigations use Blue Waters to address two important complex data-driven problems and propose new design and analysis enhancements. First, computational genomics: We have developed a software suite, IGen, for runtime optimizations of genomic workloads, including short-read alignment and human variant discovery, which are significantly faster on a single XE-6 compute node and scale out nearly linearly, leveraging available graphics processing units (GPUs) on XK-7 hybrid nodes (Fig. 1). Second, data-driven resilience at extreme scale: Working with Los Alamos National Laboratory, National Energy Research Scientific Computing Center, and CRAY, the project has developed a new design and assessment of resilience (reliability and security) of extreme-scale systems, using real data from Blue Waters failures and attacks. The tools use large-scale probabilistic graphs to drive machine learning at scale for runtime detection and mitigation of failures and attacks.

FIGURE 1: Comparing IGen's performance with unoptimized tools.



INTRODUCTION

Whole-genome sequencing and analysis is an increasingly important part of the standard of care in many hospitals and will continue to be in the years to come. The standard pipeline used for this purpose involves high computational and storage costs, which is a major hurdle for the routine implementation of genome-based individualized medicine. In this project, we have demonstrated the potential for runtime-level performance optimizations for a large number of computational-genomics workloads and built a prototype application called IGen for short-read alignment and human variant discovery. IGen is significantly (Fig. 1) faster on a single XE-6 compute node, scales out nearly linearly as a message passing interface (MPI) job, and leverages the GPUs available on the XK-7 hybrid nodes. IGen will allow large hospitals to make use of high-performance computing resources such as Blue Waters to perform their analysis in a cost-effective manner. However, even at the scale of a single large hospital, thousands of compute nodes will have to be used to keep up with sequence data being analyzed for every arriving patient. At this scale, fault tolerance becomes an important issue, which must be tackled, keeping in mind the application as well as the system configuration. In this project, we analyze the log information from various sources (e.g., syslogs) to understand the fault-isolation domains in the subsystems and their complex interactions leading to unusual failure modes.

METHODS & RESULTS

Efficient Genomic Pipelines

Common Mathematical Kernels: Static analysis of the algorithms used in computational-genomics tools revealed the existence of common algorithmic kernels. These kernels form the basis of the mathematical models used to analyze genomic data, and a profiling study on Blue Waters revealed that these contribute to the bulk of processing time. We built efficient implementations of these kernels for the alignment and variant-calling steps for the AMD central processing units (CPUs) and the NVIDIA GPUs in Blue Waters. These were compiled into a curated list of "best-performant" kernels as a part of the Illinois Genomics Execution Environment (IGen) library. The IGen library also provides primitives for handling genomic data by utilizing the parallel file access (MPI file I/O) instead of the POSIX-based I/O used in the traditional tools.

Data Flow-Based Runtime System: The static analysis mentioned above also led us to the observation that most computational-genomics applications can be expressed as directed acyclic graphs (DAGs) with kernels as vertices and data dependencies between kernels as edges. We built a runtime system called ExEn (the Execution Engine) to execute computational-genomics applications as DAGs across CPUs and GPUs on a single node and scale to multiple nodes using MPI one-sided communication primitives. At the core of the ExEn runtime is a scheduling algorithm that can perform task placement using three constraints:

- Locality: Minimizing data movement, keeping in mind ccNUMA (Cache Coherent Non-Uniform Memory Access) domains, accelerator-to-host communication, and communication over the Gemini network.
- Processor affinity: Given implementations of some kernel functions on both CPUs and GPUs, deciding on a partition of work between the processors, keeping in mind the structure of the DAG and the historic performance of a kernel on that device for a "median" dataset.
- Shared resource contention: Accounting for microarchitectural resources on processors to ensure that colocated tasks do not significantly interfere with each other's performance.

Extreme-Scale Resilience

To understand the triggers of the recovery mechanisms and its failure/success, we created an augmented LogDiver tool (ALDT). The ALDT creates recovery-sequence clusters by performing temporospatial modeling of error logs, which helps to track failure/recovery-related events, which can affect the currently triggered recovery. Recovery-sequence clusters help to understand the failure propagation path and quantify the impact on the system/applications (Fig. 2). This abstraction was used to understand the reasons for the failure of the recovery mechanisms in the Blue Waters interconnect networks. The following insights on interconnect-related failures and recovery were developed:

- Our analyses identified the following causes for the failure of recovery mechanisms: (a) failures during recovery, (b) lack of hardware support to maintain consistent routing tables across the interconnect network, and (c) bad coordination (handshake and timeout issues) between services and different recoveries during the recovery period. Further, this understanding will allow system designers to create fault-injection test beds to evaluate and check next-generation systems.
- Failures during recovery are assumed to be low-probability events and are ignored in theory and practice. However, we show that this is no longer the case. Specifically, we show that 24% of the link-

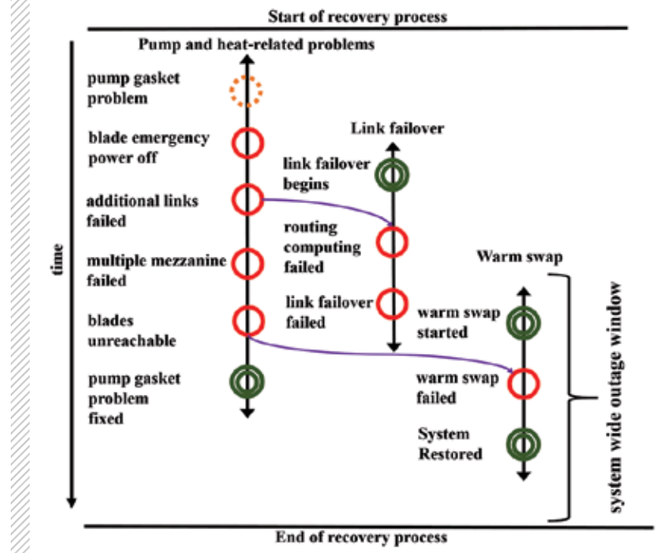


FIGURE 2: Failure propagation path showing sequence of events leading to system-wide outage.

recovery operations and 8% of blade-swapping procedures failed, which led to the failure of 20% of the active applications during these recovery epochs. A successful recovery does not guarantee to protect the application and system but only 0.02% of the active applications failed during the successful recovery period.

- Using real attack data and associated alerts as drivers, we have developed and evaluated *AttackTagger*, an adaptive learning-based IDS. The approach is based on probabilistic graphical models—specifically factor graphs—which integrates security alerts from multiple sources for accurate and preemptive detection. The method was validated using real data from attacks at NCSA.

WHY BLUE WATERS

Blue Waters is one of the few systems that can scale computations to tens or hundreds of thousands of cores on CPUs and GPUs. It also enables the study of failures in production petascale systems with its unique mix of XE6 and XK7 nodes. This capacity allows us to understand the performance–fault-tolerance continuum in HPC systems by enabling the investigation of application-level designs for mixed CPU and GPU node systems, and fault isolation in system components to mitigate failures at the application level. This allows us to design high-performance and resilient genomics pipelines that can make use of HPC systems.

NEXT GENERATION WORK

We ascertained the performance pathologies of several common kernels used in popular computational-genomics tools and built a scheduling algorithm that is able to dynamically decide task placement in a heterogeneous cluster. Also, we analyzed log data produced by Blue Waters to discover and quantify new failure modes that could not be efficiently handled by system-level recovery mechanisms. Our future work will bring together these observations to build a holistic runtime system that will jointly reason about performance and resiliency for coordinated placement and checkpointing decisions.

PUBLICATIONS AND DATA SETS

Banerjee, S. S., et al., Efficient and Scalable Workflows for Genomic Analysis. Proceedings of the *ACM International Workshop on Data-Intensive Distributed Computing (DIDC '16)* (ACM, New York, New York, June 1, 2016), pp. 27–36.

Jha, S., et al., Understanding Gemini Interconnect Failovers on Blue Waters. *Proceedings of the Cray User's Group (CUG)*, London, England, May 8–12, 2016.

Banerjee, S. S., et al., Sequences to Systems. Coordinated Science Laboratory Technical Report UILU-ENG-16-2201.

Di Martino, C., et al., LogDiver: A Tool for Measuring Resilience of Extreme-Scale Systems and Applications. Proceedings of the *5th Workshop on Fault Tolerance for HPC at eXtreme Scale (FTXS '15)* (ACM, New York, New York, June 15–19, 2015), pp. 11–18.

Cao, P., E. C. Badger, Z. T. Kalbarczyk, and R. K. Iyer, A Framework for Generation, Replay, and Analysis of Real-World Attack Variants. Proceedings of the *Symposium and Bootcamp on the Science of Security (HotSoS 2016)* (ACM, New York, New York, April 19–21, 2016), pp. 28–37.

LARGE-SCALE LEARNING FOR VIDEO UNDERSTANDING

Allocation: GLCPC/377 Knh
PI: Jia Deng[1]
Co-PI: Jason Corso[1]

[1]University of Michigan

EXECUTIVE SUMMARY

Video understanding, endowing computers with the ability to interpret videos as humans do, is one of the fundamental challenges of computer vision and artificial intelligence. Video data is ubiquitous, but our ability to perform automated analysis on such data is still primitive. In this project, we use Blue Waters to advance the research in video understanding, including recognizing human actions and activities, extracting high-level semantics from instructional videos, and accelerating deep neural network computation.

INTRODUCTION

Video understanding, endowing computers with the ability to interpret videos as humans do, is one of the fundamental challenges of computer vision and artificial intelligence. Video data is ubiquitous and is projected to account for 79% of all consumer Internet traffic in 2018 [1]. Yet our ability to perform automated analysis on such data is still primitive. We use Blue Waters to advance the research in video understanding.

One important problem is to understand human actions and activities. That is, given a visual input such as a video frame, generate a list of human action categories and their locations, for example, predicting that there is a person riding a horse at the lower left region of the given video frame. Automated recognition of human actions is key to the success of many important applications, such as human-computer interaction, robotics, and smart healthcare systems.

Another problem is to understand the high-level semantics of instructional videos with a focus on cooking activities. For a given cooking video, such as making a peanut butter and jelly sandwich, we seek to learn a cross-modal model of the temporal structure and constraints of the cooking process from both visual and audio content. The learned model will be able to generate a visual-textual summary of the process and will serve as a natural index for search and query across many such processes.

A third problem is in the area of robotic localization and mapping, which have recently become important in the context of autonomous driving. With the availability of more computing power, techniques like deep reinforcement learning [2] have become viable in many practical problems, including autonomous driving. An interesting question is how to improve deep reinforcement learning algorithms using games and simulations. We are interested in evaluating this technique on real-world problems such as learning the concept of objects and learning simple physics rules.

A fourth problem is how to make video understanding algorithms efficient. One aspect is how to accelerate the computation of deep neural networks (DNN) [3, 4], which are widely used for many subtasks for video understanding.

METHODS & RESULTS

Our main approach is machine learning. For the problem of understanding human actions and activities, we investigated deep neural networks (DNN) [3, 4]. The recent development of DNNs has led to large improvements for object recognition [5]. But compared to objects, human actions are far more complex. We have found that naively applying DNN-based object recognition algorithms for human actions does not perform well. We thus investigated novel DNN architectures for recognizing human actions. We have developed a novel multi-stream architecture that can integrate cues from humans, objects, and scene context. Our approach has achieved **state-of-the-art performance** on a large-scale action recognition benchmark [6].