Thesis Proposal: Scalable Systems and Algorithms for Genomic Variant Analysis

Frank Austin Nothaft

Abstract

Recent improvements in DNA sequencing technologies have dropped the cost of sequencing a human genome to under $1,000. This drastic economic change has enabled both the use of sequencing in clinical medicine and the genotyping of large cohorts. However, this shift brings new problems: the cost of analyzing genomic data is growing as quickly as the cost of collecting genomic data is falling.

Genomics is experiencing quintessential "big data" problems. To uncover the link between genomic variation and traits, we need to statistically analyze large cohorts. Because each genome represents $\mathcal{O}(1-100\text{GB})$ of data, moderately sized cohorts like the 1,000 Genomes project (which contains more than 70TB of data) present significant data storage and processing challenges. By collecting and processing this data, we are able to uncover statistical linkages between genomic variation and diseases. However, these statistical correlations do not always provide biological insight: in some cancers, such as Acute Myeloid Leukemia (AML), very few genes are modified by mutations. In diseases like AML, we need to make use of both statistical links and novel genomic annotation techniques to understand the underlying disease process.

This problem requires a two pronged approach that tackles both infrastructural and algorithmic problems. We have developed the ADAM system, which provides an efficient API for distributing genomic analyses across many nodes. On top of ADAM, we are building Avocado, a distributed variant caller that introduces efficient algorithms for identifing genomic variants. In addition to having lower runtime complexity than prior variant detection algorithms, the core algorithm in Avocado computes canonical representations of variation, which simplifies downstream statistical analyses. To enable very large scale genotype-phenotype association queries, we are building Gnocchi, a map-reduce based statistical query engine. Finally, we are developing a novel algorithm, Fig, which can be used to identify variants that modify the effect of regulatory protein binding sites in the genome. By understanding how genome variation can impact transcriptional regulation, we hope to better understand complex diseases like AML.

1 Thesis Statement

To run genotype-phenotype association analyses on large cohorts ($\mathcal{O}(> 100,000)$ individuals) requires the use of novel computational technology, and the development of linear time genome analysis algorithms. By coupling these analyses with advanced genomic annotation algorithms, we can better understand the role of variants in non-coding regions of the genome, and their impact on diseases like AML.

2 Background

Since the completion of the Human Genome Project in 2003, genome sequencing costs have dropped by more than $10,000\times$ [23]. The rapidly declining cost of sequencing a single human genome has enabled large sequencing projects like the 1,000 Genomes Project [32] and the Cancer Genome

Atlas (TCGA, [39]). As these large sequencing projects perform analysis that process terabytes to petabytes of genomic data, they have created a demand for genomic analysis tools that can efficiently process these scales of data [28, 34].

Over a similar time range, commercial needs led to the development of horizontally scalable analytics systems. The development and deployment of MapReduce at Google [6, 7] spawned the development of a variety of distributed analytics tools and the Hadoop ecosystem [1]. In turn, these systems led to other systems that provided a more fluent programming model [40] and higher performance [43]. The demand for these systems has been driven by the increase in the amount of data available to analysts, and has coincided with the development of statistical systems that are accessible to non-experts, such as Scikit-learn [25] and MLI [33].

With the rapid drop in the cost of sequencing a genome, and the accompanying growth in available data, there is a good opportunity to apply modern, horizontally scalable analytics systems to genomics. New projects such as the 100K for UK, which aims to sequence the genomes of 100,000 individuals in the United Kingdom [10], and the Department of Veterans Affairs' Million Veteran project [37] will generate three to four *orders of magnitude* more data than prior projects like the 1,000 Genomes Project [32]. Additionally, periodic releases of new reference datasets such as reference genomes necessitates the periodic re-analysis of these large datasets. These projects use the current "best practice" genomic variant calling pipelines [3], which takes approximately 120 hours to process a single, high-quality human genome using a single, beefy node [35]. To address these challenges, scientists have started to apply computer systems techniques such as map-reduce [13, 19, 29] and columnar storage [9] to custom scientific compute/storage systems. While these systems have improved analysis cost and performance, current implementations incur significant overheads imposed by the legacy formats and codebases that they use.

Additionally, although these experiments have provided us with a large set of data about human variation, several large questions remain. One large question pertains to the role of variation that occurs outside of the coding portions of the human genome. While these variants do not impact protein structure, variants in regulatory regions can impact the rate of transcription of nearby genes [15, 38]. Variation in regulatory regions is critical to understanding diseases that are likely to have genomic drivers, but where few variants occur in coding sections of the genome, such as AML [36]. While we can use the aforementioned statistical techniques to identify correlative links between these variants and the diseases we would like to study, these statistical techniques do not explain the underlying disease biology, and cannot be used to determine causation.

To do this, we build upon projects such as ENCODE [11] that have profiled the interaction of regulatory elements with genomic sequence, While there is an emerging literature that is studying the "grammatical architecture" of regulatory regions [15, 38], this literature largely depends on synthetic experiments [30] to drive inference. While the results from these synthetic experiments are valuable, and will improve our ability to model the impact of variants on regulation, we instead seek to ask if there is a way for us to understand this relationship from existing variation datasets? We believe we can answer these questions by building a variant annotation engine that incorporates knowledge of this regulatory grammer, and integrating these annotations into our correlative genomic models.

3 Work To Date

To address the immediate computational needs of sequencing analyses, we embarked on the design of the ADAM system [18, 24]. ADAM is a genomic processing system that was built using open source technologies that are designed for data intensive computing, such as Apache Parquet [2] and Spark [42, 43]. The main goal of building the ADAM system was to demonstrate that a well architected system that was built using commodity technologies could outperform optimized, genomics-specific systems, while letting bioinformaticians write genomic analyses using higher level primitives.

To make it possible to program genomic analyses against higher level abstractions, we created the decomposed stack shown in Figure 1, which was inspired by the stack models common in networking [44]. The key feature of this stack model is the use of a schema to describe the data we are processing. Conventional genomics pipelines rely on binary data formats that do not distinguish the logical structure of data from the physical representation of data. This is problematic, as legacy genomic formats such as SAM/BAM [16] use the blurring of lines between logical and physical representations to perform "optimizations" that make analyses brittle to write. In the popular Genome Analysis Toolkit (GATK, [19]), analyses are written against the "walker" API, which presents a sorted iterator over the genome. By blurring the bounds between the logical and physical views of the data, the GATK can accelerate "walker" traversals by pushing a sort order invariant down into the data. However, as we have shown in prior work [24], this approach can lead to subtle algorithm correctness bugs. In this prior work, we demonstrate how these same algorithms can be written using common relational primitives such as aggregates and joins, and we show that this allows us to scale genomic analyses linearly to run on thousands of cores.

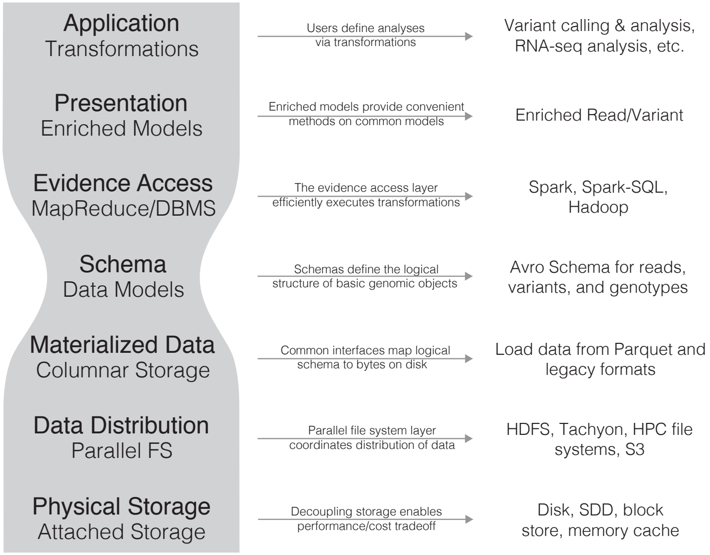
Application
Transformations
Users define analyses via transformations
Variant calling & analysis, RNA-seq analysis, etc.
Presentation
Enriched Models
Enriched models provide convenient methods on common models
Enriched Read/Variant
Evidence Access
MapReduce/DBMS
The evidence access layer efficiently executes transformations
Spark, Spark-SQL, Hadoop
Schema
Data Models
Schemas define the logical structure of basic genomic objects
Avro Schema for reads, variants, and genotypes
Materialized Data
Columnar Storage
Common interfaces map logical schema to bytes on disk
Load data from Parquet and legacy formats
Data Distribution
Parallel FS
Parallel file system layer coordinates distribution of data
HDFS, Tachyon, HPC file systems, S3
Physical Storage
Attached Storage
Decoupling storage enables performance/cost tradeoff
Disk, SDD, block store, memory cache

Figure 1: A Stack Model for Genomic Analyses

As a continuation of this work, we are working on a manuscript that demonstrates how ADAM can be used to improve the performance of conventional variant calling pipelines, without sacrificing accuracy. To do this, we have evaluated a hybrid variant calling pipeline that uses ADAM, along with the GATK's HaplotypeCaller [8]. We have compared this to the GATK's "Best Practices" pipeline [3], and have been able to achieve a 3× latency improvement while improving cost by 2×. The two pipelines generate statistically equivalent variant calls. As part of the manuscript, we will demonstrate how ADAM makes it less expensive to process large genomic datasets by recalling the 50TB Simons Genome Diversity Dataset [31].

We have been developing several tools that run on top of the ADAM processing framework. One of these tools is the Avocado variant caller, which we aim to use to replace the GATK. Avocado is a fully distributed variant caller, which calls variants using a local reassembly strategy. We are in the process of preparing a manuscript on the novel local reassembly algorithm used by Avocado. While realignment and reassembly based algorithms are used in all common variant callers (such as the GATK's HaplotypeCaller, Platypus [27], and Scalpel [22]), these algorithms are slow due to their high runtime complexity. These algorithms use a graph-theoretic algorithm to assemble potential haplotypes from the reads around a putative variant locus. However, these algorithms achieve poor throughput because of the existance of a $\mathcal{O}(n^2)$ stage where all haplotype-read pairs are scored. In Avocado, we exploit the structure of the *de Bruijn* graph that is assembled from the reads at the locus to achieve $\mathcal{O}(n)$ performance. Additionally, our algorithm provides provably canonical representations of insertion and deletion variants, which simplifies the joint analysis of multiple samples.

4 Proposed Work

I plan to tackle two remaining problems:

- How can we parallelize the statistical analysis of genotype-phenotype associations on large cohorts?
- Can we use genomic annotations to better understand the impact of low frequency variation on transcriptional regulation?

Although tools such as PLINK [26, 5] can already be used to run genotype-phenotype analyses, these tools have largely been designed to work with legacy genetic data. The data generated by modest sized sequencing experiments (such as the 1,000 Genomes project, which has 1.8TB of genotypic data) has grown to the point where the data is too large to fit in memory on a single machine. To work around this, these tools rely on ad hoc parallelization methods [5], such as parallel script dispatch. These ad hoc methods are both inefficient and prone to machine failures, and cannot make use of efficient distributed file systems, like HDFS. The I/O patterns of genotype-phenotype analyses that are run on genotypes generated by next generation sequencing workflows are particularly inefficient, as these queries only touch select fields from the genotype data, but must read the entirety of each VCF genotype record. By building Gnocchi, a genotype-phenotype correlation engine, on top of ADAM, we will be able to make use of our columnar file format to eliminate the I/O bottleneck present in current genotype-phenotype tools and to enable the efficient parallelization of genotype-phenotype analyses to hundreds of machines.

Additionally, with this work, we see a great opportunity to make a contribution to the state-of-the-art in scalable systems for machine learning. While recent research at the intersection of computer systems and machine learning has focused on algorithms for training a single statistical model on systems that are "tall and skinny" [21, 41], genomics exhibits neither of these characteristics. Unlike Internet scale companies, where feature vectors are small and there are many users, feature vectors for genomic datasets are comprised of all of the genotypes seen for an individual sample. For a whole genome sequencing run, this sparse vector can represent more than 1 million genotypes. Although fast algorithms are known for fitting linear mixed models [17] for learning genotype-phenotype correlations, there has not been significant research into fast clustering methods for "wide and flat" data. In preliminary work, we have found that sampling-based approaches can be used to efficiently apply parallel methods for clustering on tall and skinny datasets [4, 21] to this wide and flat data, but at the loss of some accuracy. We will improve our clustering accuracy by deriving efficient parallel methods that are optimized for wide and flat data.

While genotype-phenotype analyses can generate correlative findings, these findings do not necessarily shed light on the underlying biology that drives these correlations. This is particularly troublesome for the non-coding regions of the genome, where traditional variant effect annotation methods that look for impacts on protein composition cannot be used [20]. We have done some preliminary investigations to see how variation impacts the regulatory regions upstream of the transcription start site (TSS) of a gene. We implemeneted this as part of the Fig tool, which is an ADAM-based engine for annotating genomic variants. In this preliminary experiment, we searched for variants from phase 3 of the 1,000 Genomes project that modified transcription factor binding affinity or spacing. Since Transcription Factor Binding Site (TFBS) spacing requires knowledge about whether two TFBS are on the same strand of DNA, we made use of the phasing from the 1,000 Genomes project to annotate phased haplotypes. To annotate TFBSs, we used data from the ENCODE project that was compiled by Kheradpour and Manolis [12] and the GRCh37 gene annotations.

Our preliminary experiments showed several interesting results. First, we found that the regulatory regions upstream of the TSS have a lower rate of variation than expected. 43% of the annotated regions did not contain variants, which is lower than expected as the rate of common variation in the human genome is 0.1%[1]. Under a Binomial distribution with $p = 0.001$ and $n = 1500$, we expect only 22% of regions to not have variants. This implies that these regions are evolutionarily conserved. However, we did see several regions where more than three TFBSs were lost, which warrants further investigation. Additionally, we found that these variants were unlikely to modify the binding affinity of TFBSs. Binding compatibility was lost at fewer than 1% of sites, and binding affinity was decreased at approximately 10% of sites.

Ultimately, we would like to unify the annotations generated by Fig with the associations generated by Gnocchi. By applying these methods to a large dataset, such as the GEUVADIS subset of the 1,000 Genomes project [14], we will be able to verify whether the synthetic models for regulatory grammar [15, 30, 38] are applicable in vivo. Additionally, this work will serve as a valuable way to calibrate the functional annotations generated by Fig. Although we have gained TFBS data from the ENCODE project [11, 12], it is unlikely to believe that all predicted TFBS have an equal impact on transcriptional regulation across all genes. By feeding this data back into the Fig annotation engine, we can better predict the effect of variants that occur in the regulatory regions. With this knowledge, we can better understand the role of the non-coding variants in diseases by being able to predict the impact of these variants on gene expression.

References

[1] Apache. Hadoop. `http://hadoop.apache.org`.

[2] Apache. Parquet. `http://parquet.apache.org`.

[3] Auwera, G. A., Carneiro, M. O., Hartl, C., Poplin, R., del Angel, G., Levy-Moonshine, A., Jordan, T., Shakir, K., Roazen, D., Thibault, J., et al. From FastQ data to high-confidence variant calls: The Genome Analysis Toolkit best practices pipeline. *Current Protocols in Bioinformatics* (2013), 11–10.

[4] Bahmani, B., Moseley, B., Vattani, A., Kumar, R., and Vassilvitskii, S. Scalable k-means++. *Proceedings of the VLDB Endowment 5*, 7 (2012), 622–633.

[5] Chang, C. C., Chow, C. C., Tellier, L. C., Vattikuti, S., Purcell, S. M., and Lee, J. J. Second-generation PLINK: rising to the challenge of larger and richer datasets. *GigaScience 4* (2015).

[1] This rate underestimates the rate of variation in a single individual. This estimate is derived from the 1,000 Genomes project, where one "common" variant (present in > 5% of the population) were seen per 1,000bp.

[6] Dean, J., and Ghemawat, S. MapReduce: simplified data processing on large clusters. In *Proceedings of the 6th Symposium on Operating System Design and Implementation (OSDI '04)* (2004), ACM.

[7] Dean, J., and Ghemawat, S. MapReduce: simplified data processing on large clusters. *Communications of the ACM 51*, 1 (2008), 107–113.

[8] DePristo, M. A., Banks, E., Poplin, R., Garimella, K. V., Maguire, J. R., Hartl, C., Philippakis, A. A., del Angel, G., Rivas, M. A., Hanna, M., et al. A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nature Genetics 43*, 5 (2011), 491–498.

[9] Fritz, M. H.-Y., Leinonen, R., Cochrane, G., and Birney, E. Efficient storage of high throughput DNA sequencing data using reference-based compression. *Genome Research 21*, 5 (2011), 734–740.

[10] Genomics England. 100,000 genomes project. `https://www.genomicsengland.co.uk/`.

[11] Gerstein, M. B., Kundaje, A., Hariharan, M., Landt, S. G., Yan, K.-K., Cheng, C., Mu, X. J., Khurana, E., Rozowsky, J., Alexander, R., et al. Architecture of the human regulatory network derived from ENCODE data. *Nature 489*, 7414 (2012), 91–100.

[12] Kheradpour, P., and Kellis, M. Systematic discovery and characterization of regulatory motifs in ENCODE TF binding experiments. *Nucleic acids research 42*, 5 (2014), 2976–2987.

[13] Langmead, B., Schatz, M. C., Lin, J., Pop, M., and Salzberg, S. L. Searching for SNPs with cloud computing. *Genome Biology 10*, 11 (2009), R134.

[14] Lappalainen, T., Sammeth, M., Friedländer, M. R., ACt Hoen, P., Monlong, J., Rivas, M. A., Gonzàlez-Porta, M., Kurbatova, N., Griebel, T., Ferreira, P. G., et al. Transcriptome and genome sequencing uncovers functional variation in humans. *Nature 501*, 7468 (2013), 506–511.

[15] Levo, M., and Segal, E. In pursuit of design principles of regulatory sequences. *Nature Reviews Genetics 15*, 7 (2014), 453–468.

[16] Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., et al. The sequence alignment/map format and SAMtools. *Bioinformatics 25*, 16 (2009), 2078–2079.

[17] Lippert, C., Listgarten, J., Liu, Y., Kadie, C. M., Davidson, R. I., and Heckerman, D. FaST linear mixed models for genome-wide association studies. *Nature Methods 8*, 10 (2011), 833–835.

[18] Massie, M., Nothaft, F., Hartl, C., Kozanitis, C., Schumacher, A., Joseph, A. D., and Patterson, D. A. ADAM: Genomics formats and processing patterns for cloud scale computing. Tech. rep., UCB/EECS-2013-207, EECS Department, University of California, Berkeley, 2013.

[19] McKenna, A., Hanna, M., Banks, E., Sivachenko, A., Cibulskis, K., Kernytsky, A., Garimella, K., Altshuler, D., Gabriel, S., Daly, M., et al. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research 20*, 9 (2010), 1297–1303.

[20] McLaren, W., Pritchard, B., Rios, D., Chen, Y., Flicek, P., and Cunningham, F. Deriving the consequences of genomic variants with the Ensembl API and SNP Effect Predictor. *Bioinformatics 26*, 16 (2010), 2069–2070.

[21] Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., Freeman, J., Tsai, D., Amde, M., Owen, S., et al. MLlib: Machine learning in Apache Spark. *arχiv preprint arχiv:1505.06807* (2015).

[22] Narzisi, G., O'Rawe, J. A., Iossifov, I., Fang, H., Lee, Y.-h., Wang, Z., Wu, Y., Lyon, G. J., Wigler, M., and Schatz, M. C. Accurate de novo and transmitted INDEL detection in exome-capture data using microassembly. *Nature methods 11*, 10 (2014), 1033–1036.

[23] NHGRI. DNA sequencing costs. `http://www.genome.gov/sequencingcosts/`.

[24] Nothaft, F. A., Massie, M., Danford, T., Zhang, Z., Laserson, U., Yeksigian, C., Kottalam, J., Ahuja, A., Hammerbacher, J., Linderman, M., Franklin, M., Joseph, A. D., and Patterson, D. A. Rethinking data-intensive science using scalable analytics systems. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (SIGMOD '15)* (2015), ACM.

[25] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research 12* (2011), 2825–2830.

[26] Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M. A., Bender, D., Maller, J., Sklar, P., De Bakker, P. I., Daly, M. J., et al. PLINK: a tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics 81*, 3 (2007), 559–575.

[27] Rimmer, A., Phan, H., Mathieson, I., Iqbal, Z., Twigg, S. R., Wilkie, A. O., McVean, G., Lunter, G., WGS500 Consortium, et al. Integrating mapping-, assembly- and haplotype-based approaches for calling variants in clinical sequencing applications. *Nature Genetics 46*, 8 (2014), 912–918.

[28] Schadt, E. E., Linderman, M. D., Sorenson, J., Lee, L., and Nolan, G. P. Computational solutions to large-scale data management and analysis. *Nature Reviews Genetics 11*, 9 (2010), 647–657.

[29] Schatz, M. C. CloudBurst: highly sensitive read mapping with MapReduce. *Bioinformatics 25*, 11 (2009), 1363–1369.

[30] Sharon, E., Kalma, Y., Sharp, A., Raveh-Sadka, T., Levo, M., Zeevi, D., Keren, L., Yakhini, Z., Weinberger, A., and Segal, E. Inferring gene regulatory logic from high-throughput measurements of thousands of systematically designed promoters. *Nature biotechnology 30*, 6 (2012), 521–530.

[31] Simons Foundation. Simons genome diversity project dataset. `https://www.simonsfoundation.org/life-sciences/simons-genome-diversity-project-dataset/`.

[32] Siva, N. 1000 genomes project. *Nature Biotechnology 26*, 3 (2008), 256–256.

[33] Sparks, E. R., Talwalkar, A., Smith, V., Kottalam, J., Pan, X., Gonzalez, J., Franklin, M. J., Jordan, M. I., and Kraska, T. MLI: An API for distributed machine learning. In *13th IEEE International Conference on Data Mining (ICDM '13)* (2013), IEEE, pp. 1187–1192.

[34] Stein, L. D., et al. The case for cloud computing in genome informatics. *Genome Biology 11*, 5 (2010), 207.

[35] Talwalkar, A., Liptrap, J., Newcomb, J., Hartl, C., Terhorst, J., Curtis, K., Bresler, M., Song, Y. S., Jordan, M. I., and Patterson, D. SMASH: A benchmarking toolkit for human genome variant calling. *Bioinformatics* (2014), btu345.

[36] The Cancer Genome Atlas Research Network and others. Genomic and epigenomic landscapes of adult de novo Acute Myeloid Leukemia. *The New England Journal of Medicine 368*, 22 (2013), 2059.

[37] U.S. Department of Veterans Affairs. Million veteran program (mvp). `http://www.research.va.gov/mvp`.

[38] Weingarten-Gabbay, S., and Segal, E. The grammar of transcriptional regulation. *Human genetics 133*, 6 (2014), 701–711.

[39] Weinstein, J. N., Collisson, E. A., Mills, G. B., Shaw, K. R. M., Ozenberger, B. A., Ellrott, K., Shmulevich, I., Sander, C., Stuart, J. M., Cancer Genome Atlas Research Network, et al. The Cancer Genome Atlas pan-cancer analysis project. *Nature Genetics 45*, 10 (2013), 1113–1120.

[40] Yu, Y., Isard, M., Fetterly, D., Budiu, M., Erlingsson, Ú., Gunda, P. K., and Currey, J. DryadLINQ: A system for general-purpose distributed data-parallel computing using a high-level language. In *OSDI* (2008), vol. 8, pp. 1–14.

[41] Zadeh, R. B., Meng, X., Yavuz, B., Staple, A., Pu, L., Venkataraman, S., Sparks, E., Ulanov, A., and Zaharia, M. linalg: Matrix computations in Apache Spark. *arχiv preprint arχiv:1509.02256* (2015).

[42] Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., Franklin, M., Shenker, S., and Stoica, I. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation (NSDI '12)* (2012), USENIX Association, p. 2.

[43] Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., and Stoica, I. Spark: cluster computing with working sets. In *Proceedings of the 2nd USENIX conference on Hot topics in Cloud Computing (HotCloud '10)* (2010), p. 10.

[44] Zimmermann, H. OSI reference model–the ISO model of architecture for open systems interconnection. *IEEE Transactions on Communications 28*, 4 (1980), 425–432.