ispiacademy : Pages:15–27

Original Research

Optimizing Resource Allocation for Big Data Workloads in Cloud Computing Platforms

Ahmad Faiz Bin Rahman[1] and Zainuddin Bin Yusof[2]

[1]Sabah Institute of Computer Studies, Department of Information Technology, Jalan Tun Fuad Stephens No:88, Kota Kinabalu, Sabah, Malaysia.
[2]Research Assistant at Malaysia University of Science and Technology.

Abstract

This paper presents a rigorous examination of strategies for optimizing resource allocation in cloud computing platforms handling large-scale data-driven workloads. The problem of resource optimization becomes particularly crucial when heterogeneous clusters must accommodate intensive jobs with varying computational, storage, and networking demands. In this work, we analyze frameworks capable of dynamically distributing resources across a massive pool of nodes, focusing on performance metrics such as execution latency, throughput, and cost efficiency. We discuss approaches for predicting workload characteristics in real time, leveraging algorithmic and statistical models that guide scheduling policies to maximize utilization while avoiding excessive resource contention. Our discussion emphasizes practical issues that arise in live production clusters, including time-varying data arrival patterns and the effects of skewed job distributions on both performance and fault tolerance. We incorporate highly advanced mathematical modeling to characterize the cloud environment, applying theoretical insights to support adaptivity in resource provisioning. The performance of the proposed strategies is demonstrated through hypothetical yet carefully constructed results showing significant reductions in latency and improvements in overall computational throughput. While the methodologies exhibit robust behavior over a wide range of workloads, specific limitations arise from incomplete knowledge of future demand patterns and dependencies on accurate forecasting. The study concludes by outlining potential avenues for future refinements, ensuring broader applicability and resilience.

1. Introduction

Cloud computing platforms have grown to become the backbone of modern enterprise infrastructures, scientific research projects, and real-time data analytics [1]. As computational demand continues to surge, resource allocation mechanisms have had to evolve to handle increasingly heterogeneous workloads characterized by fluctuating intensities and complex interdependencies. In large data centers, considerable attention is devoted to effectively balancing processing power, memory, and storage to ensure minimal response times and high resource utilization levels [2]. Achieving optimal or near-optimal resource allocation frequently involves intricate decisions that must be made under constraints imposed by hardware, network limitations, and the dynamic nature of big data applications.

A key motivating factor behind this research lies in the tension between resource overprovisioning and underprovisioning. Overprovisioning guarantees stable performance but yields inefficiencies and inflated operational costs [3]. Underprovisioning may minimize short-term expenditures but leads to performance bottlenecks, service-level violations, and compromised user experience. In multi-tenant cloud environments, ensuring fair and efficient resource division is a nontrivial challenge [4]. Different workloads, often arriving concurrently, can exhibit widely varying patterns of CPU usage, memory consumption, disk input-output rates, and network throughput requirements [5]. The goal is to craft

allocation policies that satisfy service-level objectives while optimizing energy usage, cluster occupancy, and overall computational throughput.

The complexity is heightened when data-driven workloads exhibit sharp spikes in volume and velocity, as is often the case in domains like streaming analytics and real-time machine learning inference [6]. System performance can degrade rapidly in the absence of robust mechanisms to reallocate or scale resources in an agile manner. Failures in any layer of the system, from software to hardware, can propagate detrimental effects throughout the cluster [7]. Methods that dynamically adapt to these conditions require advanced techniques grounded in queueing theory, stochastic modeling, and real-time optimization strategies. Indeed, many state-of-the-art approaches couple predictive analytics with feedback-based control to minimize the latency penalty incurred by abrupt changes in workload characteristics.

To set the stage for the technical discussion that follows, we examine several fundamental building blocks [8]. First, we note the concept of resource heterogeneity, in which different machine types or virtual machine flavors are optimized for specific tasks (such as compute-intensive versus memory-intensive workloads). Second, we explore how advanced scheduling models exploit knowledge of priority classes or job deadlines to distribute tasks more intelligently [9]. Third, we address how job profiling and performance prediction can be leveraged to allocate resources in a cost-effective yet performance-aware manner. These considerations guide the design of algorithms that must strike an adequate balance between local efficiency and global cluster optimization [10].

In this work, we develop an in-depth framework that couples insights from classical optimization theory with specialized heuristics designed for large-scale computing clusters. Our presentation delves into detailed mathematical models while contextualizing them within practical system considerations. Over subsequent sections, we outline the problem formulation, propose theoretical solution strategies, and highlight the ramifications of uncertain or partial information regarding future workloads [11]. We further discuss the complexities introduced by dynamic resource allocation in the presence of big data workloads and demonstrate how our methodology retains a degree of robustness across fluctuating conditions. Finally, we present hypothetical performance scenarios, reflect upon identified limitations, and pinpoint areas where further research could yield substantial improvements in efficiency and scalability.

2. Problem Formulation and Modeling

The primary objective in allocating resources for large-scale data-intensive workloads is often to minimize expected completion times while respecting capacity and budget constraints [12]. In mathematical terms, consider a set of jobs indexed by $j = 1, 2, \ldots, J$. Each job j may require CPU cycles, memory, storage, or other ancillary resources, which we denote collectively as $\mathbf{r}_j$. For example, $\mathbf{r}_j$ could be a vector $\left(r_j^{\mathrm{cpu}}, r_j^{\mathrm{mem}}, r_j^{\mathrm{disk}}\right)$, reflecting the specific demands each job requires. The computing environment consists of a set of servers $i = 1, 2, \ldots, I$, each with a capacity vector $\mathbf{C}_i$. The capacities might similarly be expressed as $\left(C_i^{\mathrm{cpu}}, C_i^{\mathrm{mem}}, C_i^{\mathrm{disk}}\right)$. A resource allocation scheme determines how the vectors $\mathbf{r}_j$ are assigned among the $\mathbf{C}_i$ in real time or in a batch-processed manner.

We define allocation variables $x_{i,j}$, which equal 1 if job j is assigned to server i, and 0 otherwise. However, multi-server scenarios often require splitting the job across multiple servers or resources, so in some contexts $x_{i,j}$ might be a real-valued fraction in the interval $[0, 1]$. The system's total CPU usage on server i is constrained by [13]

$$\sum_{j} x_{i,j}\, r_j^{\mathrm{cpu}} \le C_i^{\mathrm{cpu}},$$

and analogous constraints apply for memory, storage, and possibly network bandwidth. We can combine these into a vector inequality

$$\sum_{j} x_{i,j}\, \mathbf{r}_j \le \mathbf{C}_i, \quad \forall i.$$

When job splitting is allowed, each resource dimension can be assigned independently, giving a higher-dimensional but potentially more flexible optimization space. [14]

In dynamic settings, let λ represent the arrival rate of new jobs. If arrivals follow a Poisson process, one might treat the system using a queueing-theoretic framework [15]. The objective then becomes one of minimizing the mean sojourn time, subject to the aforementioned capacity constraints. In such analyses, the time needed to process job j depends on the fraction of computational resources it is assigned. If job j is allocated ϕ_j CPU cores, the service rate might be modeled as $\mu_j(\phi_j) = \alpha_j \phi_j$, where α_j reflects the parallelization factor for that job's computations. The interplay between λ and $\mu_j(\phi_j)$ directly influences the system's congestion. For certain classes of big data workloads, parallel speedup may saturate beyond a certain threshold, so more complex speedup models are required [16]. Instead of linear scaling, one might employ diminishing returns in computational capacity, such as

$$\mu_j(\phi_j) = \alpha_j \left(1 - e^{-\beta_j \phi_j}\right),$$

where β_j captures how effectively job j can utilize increasing computational resources.

We can thus define a constrained optimization problem of the form: [17]

$$\underset{\{x_{i,j}, \phi_j\}}{\text{minimize}} \quad \mathbb{E}[T],$$

subject to the capacity constraints for CPU, memory, and storage across the cluster. Here, $\mathbb{E}[T]$ is the expected completion time or sojourn time of a job, which in many queueing systems is proportional to the ratio of arrival rate to service rate, but complicated by multi-dimensional resource constraints. The presence of complicated objective functions and nonlinear constraints can make direct solutions computationally infeasible for large J. In response, researchers frequently employ heuristics or approximate solutions that rely on local decisions informed by system state [18]. The approach we present integrates queueing-theoretic insights with real-time load monitoring, enabling adaptive resource reallocation as workload conditions evolve.

Potential complexities arise from correlated job arrivals. For large-scale data analytics, jobs may arrive in bursts corresponding to periodic events, such as daily user activity patterns or streaming sensor data [19]. Simplistic assumptions, such as independent and identically distributed arrival times, may fail to capture peak-load effects. Accounting for these fluctuations requires time-variant resource allocation strategies, wherein the assignment matrix $\{x_{i,j}\}$ and CPU fraction variables $\{\phi_j\}$ must be updated continuously or at discrete intervals as new information arrives. While this approach introduces implementation overhead, it significantly boosts the platform's resilience against unexpected spikes or troughs. [20, 21]

The challenge intensifies when reliability constraints are included. Many big data workloads rely on replication or checkpointing strategies to guarantee fault tolerance. In these scenarios, each job might need multiple copies running on different servers to avoid single points of failure [22]. This replication further complicates resource allocation, necessitating additional constraints of the form

$$\sum_i x_{i,j} \geq R_j,$$

where R_j is the replication factor for job j. Handling such constraints within the same framework of dynamic, large-scale optimization underscores the need for specialized algorithmic structures that can manage dimensionality, nonlinearity, and stochasticity simultaneously. [23]

3. Analytical Solution Approaches and Queueing-Theoretic Insights

Solving the optimization problem analytically in closed form can be challenging, given the high dimensionality and intricate constraints. However, substantial insight can be gleaned by examining simplified

versions of the problem using queueing-theoretic methods. Consider a single-resource model where jobs have an arrival rate λ and a parameterized service rate $\mu(\phi)$, contingent on the fraction of CPU ϕ allocated to that job [24]. If we view each job as joining a global queue with service capacity equal to the sum of all CPU resources in the cluster, we might approximate the system by an M/M/1-type queue with variable service rates. The average response time can be estimated using Little's law or more refined results in multi-server queues. [25]

When parallelization is possible, one might refine this approach by invoking an M/M/c queue, where c represents the number of CPU cores. An approximate formula for average waiting times in the M/M/c scenario is given by

$$W \approx \frac{\rho^c}{c!(1-\rho)} \frac{c}{c - \lambda/\mu_{\text{eff}}},$$

where $\rho = \frac{\lambda}{c\mu_{\text{eff}}}$ and μ_{eff} is the effective service rate per core. Although oversimplified for real systems, such expressions provide guidelines for bounding the performance of particular allocation strategies [26]. Higher-level allocation mechanisms can then be designed to keep the system's load factor ρ within an acceptable range, thus assuring predictable response times.

Stochastic modeling also comes into play when analyzing variance in job sizes and interarrival times [27, 28]. The variability of job durations can significantly affect queue lengths and waiting times. Cloud computing platforms that run big data workloads often encounter heavy-tailed job size distributions. In these situations, classic exponential models may underestimate the likelihood of extremely large jobs that can dominate system performance [29]. More advanced distributions, such as Pareto or lognormal, may better fit empirical data, but they often lack closed-form expressions for performance measures. One alternative involves bounding techniques or large deviations principles that offer asymptotic estimates for delays or backlogs [30]. These methods, though abstract, provide valuable safety margins for system provisioning.

Another potent method for addressing the resource allocation challenge is the use of fluid or mean-field approximations. Large-scale systems with many servers and many small jobs can sometimes be approximated by continuous flows [31]. Such models replace discrete jobs with fluid that arrives at rate λ, is served at some rate $\mu(\phi)$, and is subject to constraints. In a fluid approximation, large numbers of discrete events are replaced by differential equations that describe the evolution of the system over time [32]. The solution to these differential equations provides insight into how the resource allocation and queue lengths might evolve. While fluid approximations can lose accuracy in low-load or bursty conditions, they offer a computationally tractable route for analyzing high-level performance trade-offs.

An illustrative fluid-model analysis might treat the fraction ϕ_j as a continuous variable that can be adjusted smoothly [33]. One might write:

$$\frac{dQ(t)}{dt} = \lambda - \mu\big(\phi(t)\big)Q(t),$$

where $Q(t)$ is a continuous measure of the queue length and $\mu(\phi)$ is an aggregate service capacity [34]. This equation can be extended to incorporate multi-dimensional resources and job classes, leading to systems of ordinary differential equations. Though these models require simplifying assumptions about linearity or differentiability, they can reveal stable operating points and guide dynamic resource allocation policies aimed at stabilizing or driving $Q(t)$ toward a desired target. By coupling these approaches with robust feedback controllers, cloud platforms can maintain near-optimal performance across a broad range of workload intensities. [35]

The interplay between theoretical models and practical heuristics is often pivotal. The mathematics provides bounds and conceptual guidelines, while heuristic algorithms account for complexities like job priority, resource fragmentation, and scheduling overheads. In subsequent sections, we delve into specific optimization strategies and outline how to adapt them to real-world data center conditions, discussing computational tractability and potential limitations. [36]

4. Advanced Optimization Techniques for Resource Allocation

In practical scenarios, resource allocation within a large cluster must be computed in a fraction of a second or at most within a few seconds, making purely analytical approaches impractical. As a result, many production systems turn to numerical optimization routines, machine learning–based policies, and hybrid heuristic strategies to achieve acceptable performance [37, 38]. Among the more sophisticated computational techniques are:

1) Gradient-based methods with Lagrange multipliers. Given a continuous relaxation of the job assignment problem, one can introduce Lagrange multipliers for each resource constraint [39]. The objective is to minimize the sum of expected job completion times, or some cost function that balances performance and resource usage. This leads to conditions of the form [40]

$$\nabla_{\{x_{i,j},\phi_j\}}(\mathcal{L}) = 0,$$

where

$$\mathcal{L}(\{x_{i,j},\phi_j\},\{\lambda_k\}) = \sum_j f_j(x_{i,j},\phi_j) + \sum_k \lambda_k\Big(g_k(\{x_{i,j},\phi_j\}) - b_k\Big).$$

Here, f_j might represent a performance cost for job j, and $g_k(\cdot)$ represents each resource constraint, with b_k its capacity limit. Solving this system yields a stationary point that, under convexity assumptions, is globally optimal. However, in discrete or non-convex settings, further approximations or specialized solvers are needed to converge within limited time. [41]

2) Distributed optimization and dual decomposition. In large-scale cloud environments, it may be infeasible to gather the entire state of the system into a single centralized solver [42]. Dual decomposition techniques allow partitioning the allocation problem into smaller subproblems, each associated with a subset of resources or job classes. Each subproblem is optimized locally, and then information is exchanged between subproblems via updated dual variables that reflect resource prices or usage constraints. Iterating this procedure can converge to near-optimal allocations, provided the system is not too dynamic. [43]

3) Mixed-integer linear programming (MILP). If job splitting is not permitted or is limited, the allocation variables $x_{i,j}$ become integer-valued. This yields a high-dimensional MILP that may be solved by branch-and-bound, branch-and-cut, or heuristic approaches like genetic algorithms [44]. While generic MILP solvers can handle moderately sized instances, big data workloads often produce problem scales that are beyond off-the-shelf solver capabilities. Tailored heuristic or metaheuristic methods, such as simulated annealing or tabu search, can sometimes find good feasible solutions within practical time limits. The cost is a potential suboptimality gap that may be challenging to quantify. [45, 46]

4) Machine learning–guided methods. Recent research has explored training policies that map observed system states—such as current queue lengths, resource utilizations, and job arrival characteristics—to resource allocation decisions [47]. Depending on the approach, supervised or reinforcement learning strategies can be used. Reinforcement learning, in particular, provides a framework in which the system learns from repeated interactions to allocate resources in a way that maximizes a long-term reward signal, such as throughput or cost-effectiveness. Although this can adapt well to dynamic environments, reliable operation requires substantial exploration and well-designed reward functions [48]. Convergence is not always guaranteed, especially with high-dimensional state spaces. Additionally, the learned policy may fail if the workload distribution shifts beyond what was observed during training. [49]

5) Hybrid heuristics bridging the above techniques. Production systems often combine a short-term load balancer with a longer-horizon global optimizer. The short-term layer reacts quickly to immediate scheduling demands, ensuring that arriving jobs are placed somewhere [50]. Then, periodically, a global optimizer re-evaluates and potentially migrates or reschedules jobs to improve performance metrics. This tiered approach balances responsiveness with a more strategic perspective [51]. The global optimizer

might employ distributed or ML-based methods, while the short-term balancer uses simpler heuristics driven by local resource availability.

In all cases, effective resource allocation involves navigating trade-offs between solution quality, computation overhead, and resilience to workload perturbations. Strictly speaking, from a theoretical perspective, certain formulations of the resource allocation problem are NP-hard, implying that exact solutions will not scale to extremely large instances [52]. Practical solutions must thus adopt some level of approximation or heuristics. The advanced mathematical models discussed can guide the design of these heuristics, ensuring they are underpinned by robust theoretical principles. [53]

5. Performance Evaluation and Limitations

Analyzing the performance of resource allocation algorithms in big data environments typically requires a combination of theoretical metrics, synthetic workloads, and possibly real or benchmark traces from production systems. In this section, we present hypothetical experiments designed to demonstrate the effectiveness of the proposed dynamic allocation strategies and to highlight scenarios where the models underperform. While the numerical results are illustrative, they reflect patterns frequently observed in large-scale deployments. [54, 55]

Consider a simulated environment with a cluster of 500 servers, each providing a capacity vector $(64\ \text{CPU cores}, 256\ \text{GB memory}, 2\ \text{TB storage})$. Jobs arrive according to a non-homogeneous Poisson process with an average rate of 100 jobs per minute, though with bursts peaking at 400 jobs per minute. Job sizes follow a mixed distribution: 70 percent are short tasks requiring a few CPU-minutes and under 2 GB memory, while 30 percent are more intensive tasks, some requiring tens of CPU-hours and up to 50 GB of memory. Storage and network usage vary widely but remain within cluster limits for most tasks if allocated properly [56]. The job parallelization factor is assumed to have diminishing returns beyond 16 cores, reflecting typical big data analytics tasks that exhibit partial parallelism.

We analyze two allocation strategies [57]. In Strategy A, each incoming job is greedily allocated to the least-loaded server, without dynamic reassignments unless a server approaches a critical load threshold. In Strategy B, a more advanced method continuously solves a fluid-model approximation to determine the fraction of CPU to allocate to each job. This method updates allocations every 10 seconds, reassigning resources where necessary [58]. Hypothetical results reveal that Strategy A exhibits good performance under moderate loads but struggles during bursty periods, leading to queue backups. Average job completion time increases by a factor of three during high spikes, and memory contention emerges as a frequent bottleneck [59]. Strategy B, on the other hand, manages to reduce average completion times by up to 40 percent under bursty arrivals, thanks to dynamic redistribution of CPU and memory, though at the cost of additional overhead from continuous monitoring and reallocation.

To probe the impact of heavy-tailed job sizes, we artificially introduce rare but extremely large jobs that demand prolonged CPU usage. In both strategies, these heavy jobs dominate resource usage when they occur, causing local performance degradation [60]. However, Strategy B is more resilient because it adapts resource fractions over time, isolating the large jobs so they do not monopolize entire servers. Strategy A's coarse approach frequently allows such jobs to create imbalance in the cluster. [61]

Despite the relative success of Strategy B, certain limitations are evident. If the job arrival distribution is mispredicted or changes significantly over time, the fluid-model parameters used for real-time optimization can lead to suboptimal or even destabilizing allocations. For example, if the system expects a near-constant arrival rate of small jobs but experiences a sudden surge of large memory-intensive jobs, the reallocation mechanism may lag in adjusting memory fractions, resulting in resource thrashing or repeated migrations [62]. Additionally, the overhead of continuous optimization itself can become non-negligible in extremely large clusters, overshadowing the gains made through improved scheduling. This overhead includes not only computational costs but also network and coordination overheads involved in collecting system metrics and making reallocation decisions. [63]

Another inherent limitation lies in the assumption that job performance scales smoothly with allocated resources. Real-world big data tasks might have complex performance profiles influenced by data

locality, caching mechanisms, or container overhead. For instance, doubling the CPU cores available to a job might not halve its runtime if data input-output or inter-process communication dominates execution time [64]. Additionally, tasks using distributed frameworks may rely on global data shuffles or aggregations that limit the benefits of local parallelism. Such intricacies introduce discrepancies between the theoretical models and actual outcomes, occasionally leading to suboptimal resource usage patterns. [65]

A final concern is reliability. Many cloud applications demand fault tolerance, typically addressed through replication or checkpointing. The resource overhead of maintaining multiple replicas or frequent checkpoints is not always straightforward to incorporate into short-horizon allocation decisions [66]. Some advanced queueing models account for replicated tasks by adjusting effective service rates, but this remains a simplifying approximation. In real systems, orchestrating replicas across a cluster can trigger network bottlenecks and synchronization overhead [67]. The interplay between reliability and performance thus remains an open challenge.

6. Proposed Architecture and Implementation Considerations

To translate these advanced theoretical and computational methods into a running system, an overarching control plane is needed. The control plane can be viewed as a layered stack [68]. At the lowest layer, resource monitors continuously measure per-server usage (CPU cycles, memory occupancy, storage usage, and network throughput). This data is fed to a central or distributed coordinator that executes the optimization algorithm [69, 70]. The coordinator, in turn, issues reconfiguration commands to cluster management daemons, which adjust allocations by starting, stopping, or migrating containerized tasks.

Conceptual view of the proposed multi-layer control architecture for resource allocation in a cloud platform. Monitoring agents feed data to the coordinator, which executes optimization and orchestrates local decisions across the cluster.

The control loop would operate in discrete intervals, with a typical frequency of a few seconds to a few minutes, depending on workload volatility and system scale. During each interval, a distributed optimization procedure collects resource usage statistics, job sizes, and performance metrics [71]. It then solves an approximate version of the allocation problem to update the resource shares. If the solution indicates that some tasks should migrate to less-loaded servers, the system initiates job migration while ensuring continuity of execution through checkpointing or container-based mechanisms. Over time, the system converges to a quasi-steady distribution of resources, subject to fluctuations in arrival rates and job compositions. [72, 73]

Implementation challenges arise from the need to maintain consistency in a distributed environment. Multiple servers may attempt to balance loads independently, leading to ping-pong effects if the system design does not enforce coordination or damping [74]. One approach to mitigate such oscillations is to assign a leader coordinator that collects global state and enforces a consistent set of reallocation actions at each iteration. Alternatively, a fully distributed approach may be adopted, but care must be taken to ensure that concurrency does not lead to contradictory decisions. Techniques such as virtual prices and message passing can maintain approximate global coordination without requiring a single point of control. [75]

At scale, the overhead of exchanging resource usage data and reconfiguration messages can become significant. Techniques to reduce overhead include restricting the frequency of global optimization, limiting the scope of migrations to a subset of servers, or using hierarchical clustering of servers [76]. In hierarchical clustering, each cluster node aggregates local usage data, passes aggregated statistics up to a higher-level aggregator, and so forth, culminating in a cluster-wide view. This hierarchical approach can be adapted to partition large data centers into smaller logical pools, each managed by a local control plane. Inter-pool load balancing becomes an additional layer on top of these local decisions. [77, 78]

Another practical concern is the integration with data locality and specialized processing frameworks. In big data environments, frameworks like distributed SQL engines or stream processors often have their own resource management and scheduling subsystems that operate at an application level [79]. Reconciling these application-level schedulers with the cluster-level resource optimizer can be nontrivial, as the application-level logic might override or conflict with the cluster-level decisions in pursuit of different objectives or assumptions. A well-designed architecture should expose interfaces that allow for negotiation between these layers, possibly by establishing a standard protocol for exchanging resource availability and job requirements. In the best-case scenario, the application-level scheduler is augmented to communicate predictive usage data, while the cluster-level optimizer reserves resources accordingly. [80]

From a security standpoint, multi-tenant clouds require isolation guarantees, which can complicate fine-grained resource sharing. Container-based virtualization can offer partial isolation, but allowing fluid sharing of memory or CPU cycles among untrusted tenants may introduce side-channel vulnerabilities [81]. Hence, any dynamic resource reallocation scheme must respect security policies that constrain how resources can be shared or migrated between tasks owned by different organizations. The overall design must carefully balance the pursuit of high utilization with the imperative of tenant isolation.

In summary, a robust implementation requires consideration of system-level details, from the frequency of decision-making to hierarchical data collection and scheduling constraints imposed by tenant security or application-level frameworks [82]. While the theoretical and algorithmic foundation provides a powerful toolkit, bridging the gap between model-based insights and operational realities remains a formidable aspect of resource allocation in big data cloud platforms.

7. Conclusion

This paper has presented a comprehensive analysis of resource allocation challenges and solutions for big data workloads in modern cloud computing environments [83]. By unifying elements from queueing theory, continuous optimization, and heuristic scheduling algorithms, a framework emerges that can adaptively reallocate CPU, memory, and storage resources to meet performance objectives. The theoretical constructs offer guidance on bounding latency and throughput under varying workload conditions, while advanced optimization techniques illuminate potential methods for coping with the intractably large solution spaces inherent in multi-dimensional, large-scale systems.

An overarching theme has been the tension between theoretical elegance and practical feasibility [84]. While closed-form expressions and fluid or mean-field models provide valuable insights, real-world workloads exhibit bursts, skew, and heavy-tailed distributions that demand robust and adaptive policies. Effective approaches must therefore meld rigorous mathematical modeling with heuristic algorithms, potentially informed by machine learning [85]. The hypothetical evaluation examined both static and dynamic allocation strategies, showing that continuous reoptimization can deliver significant performance improvements but requires additional overhead and depends on accurate workload characterizations. Mismatches between modeled conditions and actual system behavior can degrade performance, highlighting the need for ongoing monitoring and adaptation.

Despite the promising results, important limitations remain [86]. Models typically assume some degree of continuity or convexity in job behavior that does not always hold in practice. Overheads associated with gathering real-time telemetry, coordinating distributed decisions, and migrating running tasks can outweigh theoretical gains under certain conditions [87]. Furthermore, incorporating reliability requirements, data locality considerations, and inter-job dependencies can exacerbate the complexities of the allocation problem. This underscores the ongoing research challenge of designing resource allocation systems that are both analytically well-founded and operationally resilient.

Potential future directions include further refining queueing approximations to incorporate multi-tenancy and correlated arrivals, integrating advanced machine learning methods to forecast workload variations and predict job speedups, and exploring more scalable distributed algorithms that can converge to near-optimal allocations without centralized bottlenecks [88]. Enhanced fault tolerance mechanisms

and stronger security isolation must also be considered integral parts of the resource allocation process. As cloud infrastructures continue to expand, the importance of these topics grows, necessitating a synergy of theoretical research, algorithmic development, and practical engineering to sustain the performance and economic viability of big data workloads in the modern computing landscape[89].

References

[1] A. Gupta, A. V. Deokar, L. S. Iyer, R. Sharda, and D. Schrader, "Big data & analytics for societal impact: Recent research and trends," *Information Systems Frontiers*, vol. 20, pp. 185–194, 3 2018.

[2] M.-P. Hosseini, D. Pompili, K. Elisevich, and H. Soltanian-Zadeh, "Random ensemble learning for eeg classification.," *Artificial intelligence in medicine*, vol. 84, pp. 146–158, 1 2018.

[3] T. Bagies, W. Le, J. Sheaffer, and A. Jannesari, "Reducing branch divergence to speed up parallel execution of unit testing on gpus," *The Journal of Supercomputing*, vol. 79, pp. 18340–18374, 5 2023.

[4] G. S. Woo, D. Truong, and W. Choi, "Visual detection of small unmanned aircraft system: Modeling the limits of human pilots," *Journal of Intelligent & Robotic Systems*, vol. 99, pp. 933–947, 2 2020.

[5] M. Abouelyazid and C. Xiang, "Architectures for ai integration in next-generation cloud infrastructure, development, security, and management," *International Journal of Information and Cybersecurity*, vol. 3, no. 1, pp. 1–19, 2019.

[6] A. Ahmad, S. Garhwal, S. K. Ray, G. Kumar, S. J. Malebary, and O. M. Ba-Rukab, "The number of confirmed cases of covid-19 by using machine learning: Methods and challenges.," *Archives of computational methods in engineering : state of the art reviews*, vol. 28, pp. 1–9, 8 2020.

[7] J. H. Moore, "Ten important roles for academic leaders in data science," *BioData mining*, vol. 13, pp. 18–, 10 2020.

[8] J. Yuan, H. Abdul-Rashid, and B. Li, "A survey of recent 3d scene analysis and processing methods," *Multimedia Tools and Applications*, vol. 80, pp. 19491–19511, 2 2021.

[9] L. Qian, J. Yu, G. Zhu, L. Wang, H. Chen, H. Pang, F. Mei, W. Lu, and Z. Mei, "Overview of cloud computing," *IOP Conference Series: Materials Science and Engineering*, vol. 677, pp. 042098–, 12 2019.

[10] M. Muniswamaiah, T. Agerwala, and C. C. Tappert, "Context-aware query performance optimization for big data analytics in healthcare," in *2019 IEEE High Performance Extreme Computing Conference (HPEC-2019)*, pp. 1–7, 2019.

[11] B. Khesin, G. Misiołek, and A. Shnirelman, "Geometric hydrodynamics in open problems," *Archive for Rational Mechanics and Analysis*, vol. 247, 2 2023.

[12] R. Dubey, D. Bryde, G. Graham, C. Foropon, S. Kumari, and O. K. Gupta, "The role of alliance management, big data analytics and information visibility on new-product development capability," *Annals of operations research*, vol. 333, pp. 1–25, 11 2021.

[13] A. Jain, S. Rallapalli, and D. Kumar, "Cloud-based neuro-fuzzy hydro-climatic model for water quality assessment under uncertainty and sensitivity.," *Environmental science and pollution research international*, vol. 29, pp. 65259–65275, 4 2022.

[14] P. A. Aleksandrov, A. B. Svechnikov, V. V. Gorev, E. V. Ryan, W. H. Ryan, and H. K. Stange-Love, "Modeling of mechanical properties of highly inhomogeneous systems under external action," *Technical Physics*, vol. 64, pp. 998–1005, 8 2019.

[15] T. Abar, A. B. Letaifa, and S. E. Asmi, "Heterogeneous multiuser qoe enhancement over dash in sdn networks," *Wireless Personal Communications*, vol. 114, pp. 2975–3001, 6 2020.

[16] S. Bebortta, S. K. Das, M. Kandpal, R. K. Barik, and H. Dubey, "Geospatial serverless computing: Architectures, tools and future directions," *ISPRS International Journal of Geo-Information*, vol. 9, pp. 311–, 5 2020.

[17] F. Buonamici, R. Furferi, L. Governi, S. Lazzeri, K. S. McGreevy, M. Servi, E. Talanti, F. Uccheddu, and Y. Volpe, "A practical methodology for computer-aided design of custom 3d printable casts for wrist fractures," *The Visual Computer*, vol. 36, pp. 375–390, 1 2019.

[18] K. Sheng-Kai, "Integrating travel history via big data analytics under universal healthcare framework for disease control and prevention in the covid-19 pandemic.," *Journal of clinical epidemiology*, vol. 130, pp. 147–148, 9 2020.

[19] I. Grooms, "A comparison of nonlinear extensions to the ensemble kalman filter: Gaussian anamorphosis and two-step ensemble filters.," *Computational geosciences*, vol. 26, pp. 633–650, 3 2022.

[20] I. Ahmad, "Discover internet of things editorial, inaugural issue," *Discover Internet of Things*, vol. 1, pp. 1–4, 2 2021.

[21] R. Avula, "Architectural frameworks for big data analytics in patient-centric healthcare systems: Opportunities, challenges, and limitations," *Emerging Trends in Machine Intelligence and Big Data*, vol. 10, no. 3, pp. 13–27, 2018.

[22] P. D. Girolamo, S. D. Sabatino, C. L. Archer, C. Buontempo, S. Bordoni, G. Budillon, A. Buzzi, D. Cimini, G. Curci, J. Cuxart, S. Davolio, R. Ferretti, G. Gerosa, F. S. Marzano, M. M. Miglietta, T. Paccagnella, M. Petitta, F. Pilla, E. Richard, R. Rotunno, S. Serafin, C. Serio, A. Troccoli, and D. Zardi, "Introducing the bulletin of atmospheric science and technology," *Bulletin of Atmospheric Science and Technology*, vol. 1, pp. 1–11, 3 2020.

[23] O. Graydon, "An open question," *Nature Photonics*, vol. 17, pp. 288–289, 3 2023.

[24] T. Yang, A. Bandyopadhyay, Z. O'Neill, J. Wen, and B. Dong, "From occupants to occupants: A review of the occupant information understanding for building hvac occupant-centric control.," *Building simulation*, vol. 15, pp. 913–932, 12 2021.

[25] M. Rakic, M. Jaboyedoff, S. Bachmann, C. Berger, M. Diezi, P. do Canto, C. B. Forrest, U. Frey, O. Fuchs, A. Gervaix, A. S. Gluecksberg, M. A. Grotzer, U. Heininger, C. R. Kahlert, D. Kaiser, M. V. Kopp, R. Lauener, T. J. Neuhaus, P. Paioni, K. M. Posfay-Barbe, G. P. Ramelli, U. Simeoni, G. D. Simonetti, C. Sokollik, B. D. Spycher, and C. E. Kuehni, "Clinical data for paediatric research: the swiss approach," *BMC proceedings*, vol. 15, pp. 1–15, 9 2021.

[26] B. Weber-Lewerenz and I. Vasiliu-Feltes, "Empowering digital innovation by diverse leadership in ict – a roadmap to a better value system in computer algorithms," *Humanistic Management Journal*, vol. 7, pp. 117–134, 4 2022.

[27] G. K. Zewdie, X. Liu, D. Wu, D. J. Lary, and E. Levetin, "Applying machine learning to forecast daily ambrosia pollen using environmental and nexrad parameters.," *Environmental monitoring and assessment*, vol. 191, pp. 1–11, 6 2019.

[28] A. K. Saxena, "Evaluating the regulatory and policy recommendations for promoting information diversity in the digital age," *International Journal of Responsible Artificial Intelligence*, vol. 11, no. 8, pp. 33–42, 2021.

[29] M. S. Wajid, H. Terashima-Marin, P. N. P. Rad, and M. A. Wajid, "Violence detection approach based on cloud data and neutrosophic cognitive maps," *Journal of Cloud Computing*, vol. 11, 11 2022.

[30] D. T. Broome, C. B. Hilton, and N. Mehta, "Policy implications of artificial intelligence and machine learning in diabetes management.," *Current diabetes reports*, vol. 20, pp. 5–, 2 2020.

[31] A.-U.-H. Yasar, H. Malik, and E. M. Shakshuki, "Guest editorial: towards enhancing ambient systems, networks and technologies," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 917–918, 1 2019.

[32] A. K. Song, "The digital entrepreneurial ecosystem—a critique and reconfiguration," *Small Business Economics*, vol. 53, pp. 569–590, 7 2019.

[33] M. M. Tajiki, B. Akbari, M. Shojafar, S. H. Ghasemi, L. Barazandeh, N. Mokari, L. Chiaraviglio, and M. Zink, "Cect: computationally efficient congestion-avoidance and traffic engineering in software-defined cloud data centers," *Cluster Computing*, vol. 21, pp. 1881–1897, 6 2018.

[34] M. Parto, C. Saldana, and T. R. Kurfess, "A novel three-layer iot architecture for shared, private, scalable, and real-time machine learning from ubiquitous cyber-physical systems," *Procedia Manufacturing*, vol. 48, pp. 959–967, 2020.

[35] M. Shahin, F. F. Chen, H. Bouzary, and K. Krishnaiyer, "Integration of lean practices and industry 4.0 technologies: smart manufacturing for next-generation enterprises," *The International Journal of Advanced Manufacturing Technology*, vol. 107, pp. 2927–2936, 3 2020.

[36] Y. Chen and G. D. Luca, "Technologies for developing a smart city in computational thinking," *International Journal of Simulation and Process Modelling*, vol. 13, no. 2, pp. 91–101, 2018.

[37] Z.-B. Fan and K. Zhang, "Visual order of chinese ink paintings.," *Visual computing for industry, biomedicine, and art*, vol. 3, pp. 1–9, 10 2020.

[38] M. Muniswamaiah, T. Agerwala, and C. Tappert, "Data virtualization for analytics and business intelligence in big data," in *CS & IT Conference Proceedings*, vol. 9, CS & IT Conference Proceedings, 2019.

[39] H. Bouzary, F. F. Chen, and M. Shahin, "Using machine learning for service candidate sets retrieval in service composition of cloud-based manufacturing," *The International Journal of Advanced Manufacturing Technology*, vol. 115, pp. 941–948, 1 2021.

[40] F. Yuan, J. G. Anderson, T. H. Wyatt, R. P. Lopez, M. Crane, A. Montgomery, and X. Zhao, "Assessing the acceptability of a humanoid robot for alzheimer's disease and related dementia care using an online survey," *International Journal of Social Robotics*, vol. 14, pp. 1223–1237, 1 2022.

[41] K. Alanezi and S. Mishra, "Incorporating individual and group privacy preferences in the internet of things.," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, pp. 1–16, 3 2021.

[42] G. C. Banks, H. M. Woznyj, R. Wesslen, and R. Ross, "A review of best practice recommendations for text analysis in r (and a user-friendly app)," *Journal of Business and Psychology*, vol. 33, pp. 445–459, 1 2018.

[43] J. A. Delgado, N. M. Short, D. P. Roberts, and B. Vandenberg, "Big data analysis for sustainable agriculture on a geospatial cloud framework," *Frontiers in Sustainable Food Systems*, vol. 3, 7 2019.

[44] C. Gonzalez, "Vernetzte automobile die cloud fährt mit," *ATZelektronik*, vol. 13, pp. 58–61, 6 2018.

[45] A. Zafeiri, P. Filis, S. Shaw, J. P. Iredale, M. J. Swortwood, R. T. Mitchell, D. C. Hay, P. J. O'Shaughnessy, and P. Fowler, "Maternal smoking during pregnancy and changes in prostaglandin enzymes in the human fetus.," *Reproductive Sciences*, vol. 26, pp. A62–A390, 12 2019.

[46] R. Avula, "Optimizing data quality in electronic medical records: Addressing fragmentation, inconsistencies, and data integrity issues in healthcare," *Journal of Big-Data Analytics and Cloud Computing*, vol. 4, no. 5, pp. 1–25, 2019.

[47] H. M. Pandey, N. Bessis, N. Kumar, and A. Chaudhary, "S. i: hybridization of neural computing with nature-inspired algorithms," *Neural Computing and Applications*, vol. 33, pp. 10617–10619, 3 2021.

[48] P. Rossi, C. Castagnetti, A. Capra, A. J. Brooks, and F. Mancini, "Detecting change in coral reef 3d structure using underwater photogrammetry: critical issues and performance metrics," *Applied Geomatics*, vol. 12, pp. 3–17, 5 2019.

[49] T. Castrignanò, S. Gioiosa, T. Flati, M. Cestari, E. Picardi, M. Chiara, M. Fratelli, S. Amente, M. Cirilli, M. A. Tangaro, G. Chillemi, G. Pesole, and F. Zambelli, "Elixir-it hpc@cineca: high performance computing resources for the bioinformatics community," *BMC bioinformatics*, vol. 21, pp. 1–17, 8 2020.

[50] C.-T. Yang, T.-Y. Chen, E. Kristiani, and S. F. Wu, "The implementation of data storage and analytics platform for big data lake of electricity usage with spark," *The Journal of Supercomputing*, vol. 77, pp. 5934–5959, 11 2020.

[51] D. Gupta, S. Rani, and S. H. Ahmed, "Icn-edge caching scheme for handling multimedia big data traffic in smart cities," *Multimedia Tools and Applications*, vol. 82, pp. 39697–39717, 8 2022.

[52] F. Ciampi, A. Giannozzi, G. Marzi, and E. I. Altman, "Rethinking sme default prediction: a systematic literature review and future perspectives.," *Scientometrics*, vol. 126, pp. 1–48, 1 2021.

[53] H. Cui, C. Wang, and H. Liu, "Monitoring and analysis of distributed new energy resources based on the internet of things," *IOP Conference Series: Earth and Environmental Science*, vol. 714, pp. 042025–, 3 2021.

[54] H. M. Ali, A. B. Bomgni, S. A. C. Bukhari, T. Hameed, and J. Liu, "Power-aware fog supported iot network for healthcare infrastructure using swarm intelligence-based algorithms," *Mobile Networks and Applications*, vol. 28, pp. 824–838, 3 2023.

[55] M. K. Kansara, "Overcoming technical challenges in large-scale it migrations: A literature-based analysis and practical solutions," *JNRID*, vol. 1, no. 3, 2023.

[56] K. Seetharam, N. Kagiyama, and P. P. Sengupta, "Application of mobile health, telemedicine and artificial intelligence to echocardiography.," *Echo research and practice*, vol. 6, pp. R41–R52, 6 2019.

[57] K. O'Shea and B. B. Misra, "Software tools, databases and resources in metabolomics: updates from 2018 to 2019.," *Metabolomics : Official journal of the Metabolomic Society*, vol. 16, pp. 36–36, 3 2020.

[58] K. Börner, F. N. Silva, and S. Milojević, "Visualizing big science projects," *Nature Reviews Physics*, vol. 3, pp. 753–761, 9 2021.

[59] D. M. M. Vianny, S. A. Vaddadi, C. Karthikeyan, M. Shahid, R. Dhanapal, and M. Ravichand, "Drug-based recommendation system based on deep learning approach for data optimization," *Soft Computing*, 7 2023.

[60] M. Chakraborty, "Fog computing vs. cloud computing," *SSRN Electronic Journal*, 2019.

[61] L. Guanter, M. Brell, J. C.-W. Chan, C. Giardino, J. Gomez-Dans, C. Mielke, F. Morsdorf, K. Segl, and N. Yokoya, "Synergies of spaceborne imaging spectroscopy with other remote sensing approaches," *Surveys in Geophysics*, vol. 40, pp. 657–687, 7 2018.

[62] A. D. Balomenos, V. Stefanou, and E. S. Manolakos, "Analytics and visualization tools to characterize single-cell stochasticity using bacterial single-cell movie cytometry data.," *BMC bioinformatics*, vol. 22, pp. 531–, 10 2021.

[63] Ákos Bogdán, A. D. Goulding, P. Natarajan, O. E. Kovács, G. R. Tremblay, U. Chadayammuri, M. Volonteri, R. P. Kraft, W. R. Forman, C. Jones, E. Churazov, and I. Zhuravleva, "Evidence for heavy-seed origin of early supermassive black holes from a z10 x-ray quasar," *Nature Astronomy*, vol. 8, pp. 126–133, 11 2023.

[64] B. Demir, S. Ergunay, G. Nurlu, V. Popovic, B. Ott, P. Wellig, J.-P. Thiran, and Y. Leblebici, "Real-time high-resolution omnidirectional imaging platform for drone detection and tracking," *Journal of Real-Time Image Processing*, vol. 17, pp. 1625–1635, 10 2019.

[65] M. Bies, M. Cvetič, R. Donagi, L. Lin, M. Liu, and F. Ruehle, "Machine learning and algebraic approaches towards complete matter spectra in 4d f-theory," *Journal of High Energy Physics*, vol. 2021, pp. 1–71, 1 2021.

[66] R. Myro, "A policy for a new industrial revolution," *Journal of Industrial and Business Economics*, vol. 46, pp. 403–414, 7 2019.

[67] M. Izadi, M. Nejati, and A. Heydarnoori, "Semantically-enhanced topic recommendation systems for software projects," *Empirical Software Engineering*, vol. 28, 2 2023.

[68] M. N. Cabili, K. Carey, S. O. Dyke, A. J. Brookes, M. Fiume, F. Jeanson, G. Kerry, A. Lash, H. J. Sofia, D. Spalding, A.-M. Tassé, S. Varma, and R. N. Pandya, "Simplifying research access to genomics and health data with library cards," *Scientific data*, vol. 5, pp. 180039–180039, 3 2018.

[69] C. R. Panigrahi, J. L. Sarkar, M. Tiwary, B. Pati, and P. Mohapatra, "Datalet: An approach to manage big volume of data in cyber foraged environment," *Journal of Parallel and Distributed Computing*, vol. 131, pp. 14–28, 2019.

[70] M. Muniswamaiah, T. Agerwala, and C. C. Tappert, "Approximate query processing for big data in heterogeneous databases," in *2020 IEEE international conference on big data (big data)*, pp. 5765–5767, IEEE, 2020.

[71] M. Larbani and P.-L. Yu, "Empowering data mining sciences by habitual domains theory, part i: The concept of wonderful solution," *Annals of Data Science*, vol. 7, pp. 373–397, 5 2020.

[72] P. Burke, "Demonstration and application of diffusive and ballistic wave propagation for drone-to-ground and drone-to-drone wireless communications.," *Scientific reports*, vol. 10, pp. 14782–14782, 9 2020.

[73] R. Avula, "Strategies for minimizing delays and enhancing workflow efficiency by managing data dependencies in healthcare pipelines," *Eigenpub Review of Science and Technology*, vol. 4, no. 1, pp. 38–57, 2020.

[74] B. Pierluigi, "Potential impacts of climate change on welwitschia mirabilis populations in the namib desert, southern africa," *Journal of Arid Land*, vol. 10, pp. 663–672, 8 2018.

[75] M. Carabregu-Vokshi, G. Ogruk-Maz, S. Yildirim, B. Dedaj, and A. Zeqiri, "21st century digital skills of higher education students during covid-19—is it possible to enhance digital skills of higher education students through e-learning?," *Education and Information Technologies*, vol. 29, pp. 103–137, 11 2023.

[76] C. Plainaki, T. A. Cassidy, V. I. Shematovich, A. Milillo, P. Wurz, A. Vorburger, L. Roth, A. Galli, M. Rubin, A. Blöcker, P. Brandt, F. Crary, I. Dandouras, X. Jia, D. Grassi, P. Hartogh, A. Lucchetti, M. A. McGrath, V. Mangano, A. Mura, S. Orsini, C. Paranicas, A. Radioti, K. D. Retherford, J. Saur, and B. Teolis, "Towards a global unified model of europa's tenuous atmosphere," *Space Science Reviews*, vol. 214, pp. 1–71, 1 2018.

[77] B. Shabani, J. Ali-Lavroff, D. Holloway, S. Penev, D. Dessi, and G. Thomas, "Machine learning and cloud computing for remote monitoring of wave piercing catamarans: a case study using matlab on amazon web services," *Smart Ship Technology Conference 2020*, pp. 83–94, 10 2020.

[78] M. Abouelyazid, "Forecasting resource usage in cloud environments using temporal convolutional networks," *Applied Research in Artificial Intelligence and Cloud Computing*, vol. 5, no. 1, pp. 179–194, 2022.

[79] A. Eguchi, H. Okada, and Y. Muto, "Contextualizing ai education for k-12 students to enhance their learning of ai literacy through culturally responsive approaches.," *Kunstliche intelligenz*, vol. 35, pp. 153–161, 8 2021.

[80] A. Brinkmann, K. Mohror, W. Yu, P. Carns, T. Cortes, S. Klasky, A. Miranda, F.-J. Pfreundt, R. Ross, and M.-A. Vef, "Ad hoc file systems for high-performance computing," *Journal of Computer Science and Technology*, vol. 35, pp. 4–26, 1 2020.

[81] J. Übelhör, "Industrieunternehmen und die transformation von geschäftsmodellen im kontext der digitalisierung – eine empirische studie über die auswirkungen anhand des business model canvas," *HMD Praxis der Wirtschaftsinformatik*, vol. 56, pp. 453–467, 5 2018.

[82] E. Burns, "Neutron star mergers and how to study them," *Living Reviews in Relativity*, vol. 23, pp. 1–177, 11 2020.

[83] J. S. Rha and H.-H. Lee, "Research trends in digital transformation in the service sector: a review based on network text analysis," *Service Business*, vol. 16, pp. 77–98, 2 2022.

[84] J. H. Park, M. Younas, H. R. Arabnia, and N. Chilamkurti, "Emerging ict applications and services—big data, iot, and cloud computing," *International Journal of Communication Systems*, vol. 34, 11 2020.

[85] T. Karppi and Y. Granata, "Non-artificial non-intelligence: Amazon's alexa and the frictions of ai," *AI & SOCIETY*, vol. 34, pp. 867–876, 6 2019.

[86] Z. E. Yebdri, S. M. Benslimane, F. Lahfa, M. Barhamgi, and D. Benslimane, "Context-aware recommender system using trust network," *Computing*, vol. 103, pp. 1919–1937, 1 2021.

[87] N. Kaur, S. Bhattacharya, and A. J. Butte, "Big data in nephrology.," *Nature reviews. Nephrology*, vol. 17, pp. 676–687, 6 2021.

[88] A. Cuzzocrea, F. Martinelli, F. Mercaldo, and G. M. Grasso, "Experimenting and assessing machine learning tools for detecting and analyzing malicious behaviors in complex environments," *Journal of Reliable Intelligent Environments*, vol. 4, pp. 225–245, 10 2018.

[89] B. B. Gupta, D. P. Agrawal, S. Yamaguchi, and M. Sheng, "Advances in applying soft computing techniques for big data and cloud computing," *Soft Computing*, vol. 22, pp. 7679–7683, 10 2018.