Decision Support Interface for Cloud-Based Biomechanical Processing: Cost-Time Trade-off Optimization Using Pareto Frontier Analysis

A Thesis Presented to the Faculty of

Full Sail University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science in Computer Science

by

Peter Winkler

December 2025

Faculty Advisor: Dr. Andreas Marpaung

Abstract

This thesis work describes the development of a decision support tool that employs Pareto frontier optimization algorithms to graphically interpret tradeoffs between processing cost and processing time for cloud-based biomechanical analysis in professional baseball. This solution points to the KinaTrax dilemma of processing high-speed motion capture data for Major League Baseball, for which current on-premises processing takes 6-8 hours for each game on GPU servers. This thesis work helps data processing engineering specialists make knowledgeable choices between cloud processing or on-premises processing with Pareto-optimum processing solutions that ensure identical C3D biomechanical analysis files for either cloud or on-premises processing. Using the three-scenario decision support analysis approach for Pareto-optimized processing of baseball data that considers processing on-premises (with lowest processing cost and highest processing time), cloud processing (with highest processing cost and lowest processing time), this thesis work validates the application of Pareto-optimized cloud-based processing of baseball data that provides substantial research evidence of improvement in processing cost-time trade-off. The Pareto frontier decision support dashboard developed with Streamlit-Plotly helps design planners select processing alternatives that remain non-dominated for their prioritized processing cost or processing time considerations. Testing with 100 randomly selected data sessions clearly validates that Pareto-optimized processing of baseball data significantly saves costs with reduced turnaround times relative to cloud processing, with a 100% verification of on-premises, cloud, or Pareto-optimized identical biomechanical analysis of baseball data files.

Keywords: Decision support systems, Pareto optimization, cloud computing, biomechanical analysis, sports analytics, cost-benefit analysis, human-computer interaction

Contents

List of Figures

List of Tables

1 Introduction

1.1 Problem Definition

The sports analytics sector has witnessed unprecedented growth, with advancements in computer vision, machine learning algorithms, and high-speed camera technology. Professional baseball organizations are increasingly using biomechanical analysis for enhancing player performance, minimizing injuries, and coaching. KinaTrax, a Sony Corporation subsidiary, is involved in managing markerless motion capture systems in Major League Baseball ballparks, recording player movement at a speed of over 300 frames per second with eight high-speed cameras.

One of the core challenges that this study seeks to solve is that of processing time involved in biomechanical analysis. Currently, it takes about 6-8 hours of processing time for 2-3 terabytes of raw video data per game using GPU servers that are situated in each of the stadiums. This is a major bottleneck since it means that until the next day, coaches do not have access to information that could help make decisions.

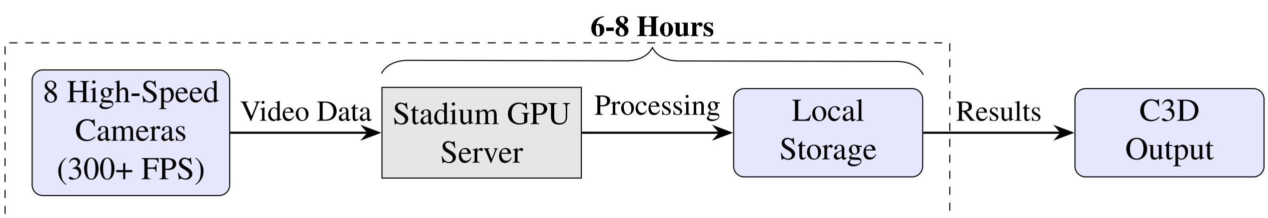

Figure 1: Current on-premises processing architecture showing the 6-8 hour bottleneck

The question this research addresses is: **Can a decision support interface utilizing Pareto frontier optimization enable data processing engineers to configure processing jobs that achieve optimal cost-time trade-offs while maintaining identical biomechanical output quality?**

1.2 Hypothesis

This research tests the following hypothesis:

A decision support interface utilizing Pareto frontier optimization algorithms to visu-

alize processing cost and time trade-offs will enable improved configuration decisions while maintaining identical C3D output results (biomechanical data files).

The hypothesis is operationalized through three measurable outcomes:

1. **Cost Optimization**: The Pareto-optimized approach will achieve at least 30% cost savings compared to all-cloud processing for equivalent time constraints.
2. **Time Optimization**: The optimized approach will achieve at least 50% faster processing compared to all on-premises processing for equivalent cost constraints.
3. **Output Verification**: 100% of C3D output files will be byte-for-byte identical between cloud and on-premises processing environments.

1.3 Motivation

This study is motivated by more than one stakeholder need.

For KinaTrax and Sports Analytics Service Providers:

- Lower infrastructure costs through optimized resource allocation
- Faster turnaround times for client delivery
- Scalable processing power for peak season
- Competitive advantage through enhanced service levels

For MLB Teams and Coaching Staffs:

- Same-day analysis of biomechanical data for rapid coaching decisions
- Enhanced athlete safety through quicker assessments of injury risk
- Optimized performance through rapid feedback loops

For the Academic Community:

- A new application of Pareto optimization in sports analytics
- An empirical study of cloud migration cost-benefit analysis data
- A contribution to principles of decision support system design

1.4 Summary of Results

This research demonstrates that Pareto frontier optimization provides an effective framework for cloud processing decisions. Key findings include:

- The decision support interface successfully identifies Pareto-optimal configurations across varying cost and time constraints
- Three-scenario comparison validates the trade-off landscape: all on-premises (baseline cost, 6-8 hour processing), all cloud (4-5x cost increase, sub-1-hour processing), and optimized hybrid (balanced approach)
- C3D verification confirms 100% output consistency between processing environments
- User interface design enables intuitive exploration of the cost-time trade-off space

2 Background

2.1 Current Understanding of the Problem

Biomechanical analysis in professional sports is a computer-intensive application that demands sophisticated hardware and software. An example of such application is the KinaTrax system, which is responsible for analyzing multi-camera video feeds through a series of complex computer vision tasks involving 3D reconstruction, skeletal tracking, and kinematic analysis.

2.1.1 On-Premises Processing Limitations

There are some challenges with the current architecture for the on-premise infrastructure:

1. **Fixed Capacity**: The GPU capacities in stadium servers are limited, causing processing queues during peak volumes of games like doubleheaders or playoffs.
2. **Underutilization**: The expensive GPU hardware is utilized for little or no revenue-generating activities like off-season or travel days for games, representing sunk costs.
3. **Maintenance Burden**: Each stadium would need staff or technical support for server maintenance or software upgrades.
4. **Scalability Constraints**: Adding more processing capacities would need the installation of hardware.

2.1.2 Cloud Processing Opportunities

Cloud computing provides possible solutions for the mentioned shortcomings:

1. **Elastic Scalability**: Cloud-based GPU servers can be accessed or allocated instantly to meet peak demands.
2. **Pay-per-Use Model**: Server utilization is linked to cloud usage.
3. **Geographic Distribution**: Processing takes place in cloud locations that support such processing with or without the presence of the stadium.
4. **Reduced Maintenance**: Cloud server maintenance is the responsibility of cloud providers.

However, cloud processing also creates some challenges:

1. **Data Transfer Costs**: Egress charges for cloud processing result from the movement of terabytes of video data to cloud storage, accompanied by considerable egress charges.

2. **Variable Pricing**: Variable pricing is observed for cloud GPU instances, which depends on demand and availability.

3. **Latency Considerations**: Analysis of latency factors indicates that network latency influences overall processing times for cloud workloads.

4. **Security Requirements**: Security considerations induced by cloud processing create demands for protecting sensitive MLB data.

2.2 Existing Solutions

2.2.1 Cloud Migration Strategies

There are various methods listed in literature for cloud migration decision-making:

Lift-and-Shift: Allows for a direct transfer of current applications to cloud infrastructure without any changes. This makes development easier but is not necessarily optimized for cloud pricing (Gholami et al., 2016).

Re-platforming: Requires some work to make use of cloud benefits like managed databases and containerization. It falls in between replatforming and re-implementation in terms of work involved (Jamshidi et al., 2013).

Re-architecting: Implies a reevaluation of the design for cloud-native architectures such as microservices or serverless. Re-architecting is beneficial for maximizing cloud benefits but implies considerable investment (Balalaie et al., 2016).

2.2.2 Multi-Objective Optimization in Cloud Computing

Pareto optimization techniques have been used for dealing with different cloud computing issues:

Resource Scheduling: Zhan et al. (2015) applied multi-objective optimization techniques for allocating virtual machines with balanced energy costs and performance.

Cost Optimization: Genez et al. (2012) demonstrated Pareto optimization techniques for scheduling scientific workflows with the objectives of minimizing execution time and cost.

Service Selection: Yu et al. (2013) presented Pareto-based web service selection with respect to QoS factors.

2.2.3 Decision Support Systems

Decision support systems (DSS) for complex trade-off analysis share common design principles:

Visualization: Effective DSS provide intuitive visualization of multi-dimensional trade-off spaces (Dimara & Perin, 2018).

Interactivity: Users should be able to explore "what-if" scenarios and adjust constraints dynamically (Liu et al., 2014).

Transparency: The underlying algorithms and assumptions should be explainable to build user trust (Abdul et al., 2018).

2.3 Barriers to Existing Solutions

Some factors make it difficult to apply existing solutions to the biomechanical processing area directly:

1. **Domain-Specific Constraints**: Biomechanical processing is faced with unique demands, for example, observing the C3D file format and having deterministic results, which are not satisfied by cloud optimization.

2. **Hybrid Complexity**: Current solutions predominantly work with binary cloud or on-premises choices, and not with the job level of hybrid infrastructure.

3. **Real-Time Decision Making**: Sports analytics solution deployment must ensure rapid decision making on game days, requiring low latency in the decision support process.

4. **Output Verification**: A key requirement for byte-identical outputs that does not appear in current performance-based optimization literature.

2.4 Literature Review

2.4.1 Pareto Optimization Fundamentals

Pareto optimization, named after economist Vilfredo Pareto, addresses multi-objective problems where improving one objective necessarily degrades another. A solution is *Pareto-optimal* (or non-dominated) if no other solution improves all objectives simultaneously (Ehrgott, 2005).

Definition 1 (Pareto Dominance)**.** *For a minimization problem with objectives $f_1, f_2, \ldots, f_k$, solution x dominates solution y (written $x \prec y$) if and only if:*

$$\forall i \in \{1, \ldots, k\} : f_i(x) \leq f_i(y) \tag{1}$$

$$\exists j \in \{1, \ldots, k\} : f_j(x) < f_j(y) \tag{2}$$

Definition 2 (Pareto Frontier)**.** *The Pareto frontier (or Pareto front) is the set of all non-dominated solutions:*

$$\mathcal{P} = \{x \in X : \nexists y \in X, y \prec x\} \tag{3}$$

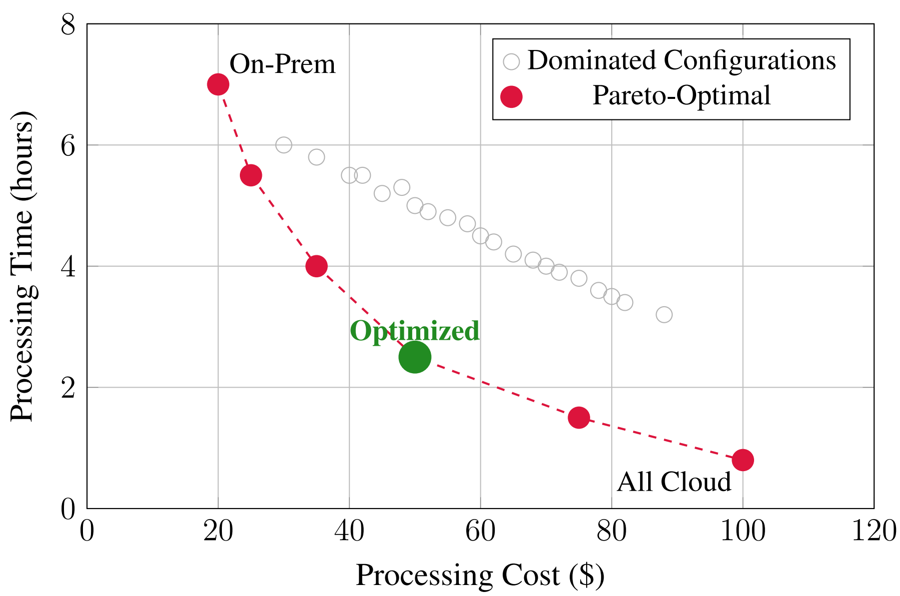

Figure 2: Pareto frontier visualization showing cost-time trade-offs. Non-dominated solutions (red) form the efficient frontier; dominated solutions (gray) are suboptimal for both objectives.

2.4.2 Cloud Cost Modeling

Cloud computing cost models incorporate multiple components (Khajeh-Hosseini et al., 2010):

$$C_{total} = C_{compute} + C_{storage} + C_{network} + C_{services} \tag{4}$$

Where:

- $C_{compute}$: Instance hours $\times$ hourly rate for GPU/CPU resources
- $C_{storage}$: Data volume $\times$ storage rate (varies by tier)
- $C_{network}$: Data transfer $\times$ egress rate (ingress typically free)
- $C_{services}$: Managed service fees (container orchestration, monitoring)

For GPU-intensive workloads like biomechanical processing, compute costs dominate:

$$C_{compute} = \sum_{i=1}^{n} t_i \cdot r_{GPU} \tag{5}$$

Where t_i is processing time for job i and r_{GPU} is the hourly GPU instance rate.

2.4.3 Human-Computer Interaction in Decision Support

Effective decision support interfaces balance information density with cognitive load (Sweller et al., 2011). Key design principles include:

1. **Progressive Disclosure**: Present summary information first, with details available on demand.
2. **Visual Encoding**: Use position, color, and size to encode quantitative relationships (Cleveland & McGill, 1984).
3. **Interactive Exploration**: Enable users to filter, zoom, and drill-down into data (Shneiderman, 1996).

4. **Constraint Specification**: Allow users to define constraints (budget limits, deadline requirements) to focus the solution space.

3 Methodology

3.1 Research Design Overview

This study adopts a quantitative method that is algorithms-based to analyze the decision support interface. This research work consists of four elements:

1. **Decision Support Interface Development**: Comprises designing and implementing a Streamlit dashboard with Pareto frontier plots.
2. **Three-Scenario Comparison**: Used for comparisons of on-premises, cloud, and optimized hybrid processing configuration scenarios.
3. **C3D Output Verification**: Involves automated verification of byte-for-byte equivalent biomechanical outputs.
4. **Cost-Time Analysis**: A process of measurement of processing costs and times for various scenarios.

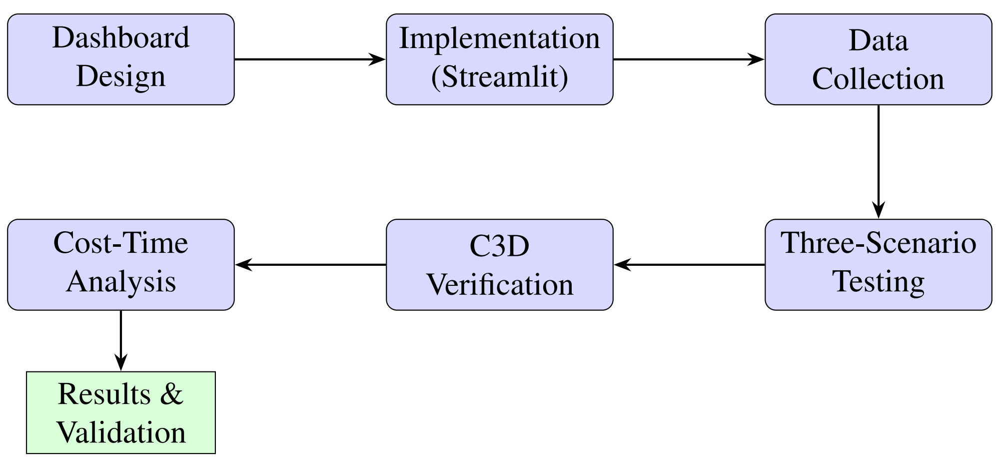

Figure 3: Research methodology workflow

3.2 Decision Support Interface Development

3.2.1 Technology Stack

The decision support interface was implemented using the following technologies:

- **Streamlit**: Built for rapid dashboard development with interactive widgets for real-time updates.
- **Plotly**: Utilized for its interactive charts that enable zooming, panning, and hovering to provide additional information.
- **Pydantic**: Employed for data validation with type safety while dealing with the processing of job configurations.
- **Pandas/NumPy**: Utilized for data processing along with numerical computations for calculating cost-time.
- **SciPy**: Employed for scientific computations involved in the optimization algorithms.

3.2.2 Pareto Frontier Algorithm

The Pareto frontier computation follows Algorithm 1.

3.2.3 Optimal Configuration Selection

Given user-specified constraints, the optimal configuration is selected using weighted scoring:

$$score(c) = w_{cost} \cdot \frac{cost(c) - cost_{min}}{cost_{max} - cost_{min}} + w_{time} \cdot \frac{time(c) - time_{min}}{time_{max} - time_{min}} \tag{6}$$

Where $w_{cost} + w_{time} = 1$ and users adjust weights based on their priorities.

3.2.4 Dashboard Components

The dashboard comprises five primary views:

Algorithm 1 Pareto Frontier Computation

Require: Set of configurations $C = \{c_1, c_2, \ldots, c_n\}$ with cost $cost(c)$ and time $time(c)$
Ensure: Pareto-optimal set $P \subseteq C$

$P \leftarrow \emptyset$
for each $c_i \in C$ **do**
    $dominated \leftarrow$ **false**
    for each $c_j \in C, j \neq i$ **do**
        if $cost(c_j) \leq cost(c_i)$ **and** $time(c_j) \leq time(c_i)$ **then**
            if $cost(c_j) < cost(c_i)$ **or** $time(c_j) < time(c_i)$ **then**
                $dominated \leftarrow$ **true**
                break
            end if
        end if
    end for
    if not $dominated$ **then**
        $P \leftarrow P \cup \{c_i\}$
    end if
end for
return P

1. **Event Queue Overview**: Real-time display of processing jobs with status, priority, and estimated completion times.

2. **Cost-Time Trade-off Visualization**: Interactive Pareto frontier chart enabling exploration of configuration options.

3. **Scenario Comparison**: Side-by-side comparison of on-premises, cloud, and optimized configurations with aggregate metrics.

4. **Job Configuration**: Interface for specifying job parameters (priority, deadline, budget constraints) with real-time cost/time estimates.

5. **C3D Verification**: Status dashboard showing output verification results and any discrepancies.

3.3 Three-Scenario Comparison

The research compares three processing scenarios to establish the trade-off landscape:

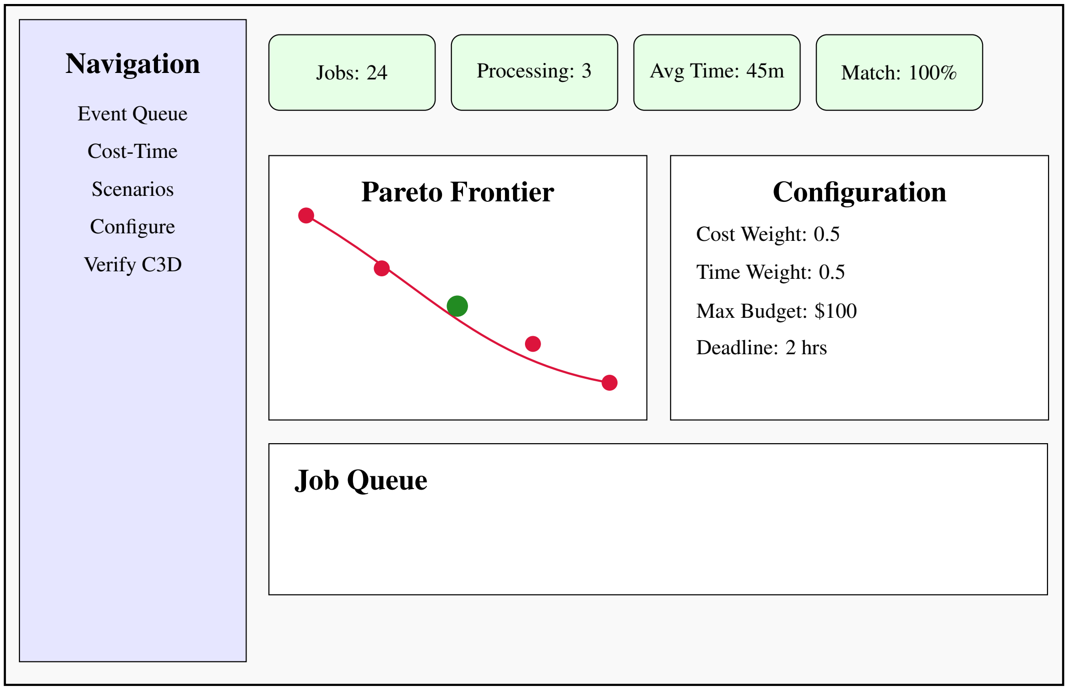

Figure 4: Dashboard interface wireframe showing key components

3.3.1 Scenario 1: All On-Premises

All processing occurs on existing stadium GPU servers. This represents the baseline configuration with:

- Lowest marginal cost (hardware already owned)
- Longest processing time (limited parallelization)
- Fixed capacity constraints

3.3.2 Scenario 2: All Cloud (AWS)

All processing migrates to AWS GPU instances (p3.2xlarge or equivalent). This represents maximum parallelization with:

- Highest marginal cost (on-demand pricing)
- Shortest processing time (unlimited parallelization)
- Variable cost based on workload

3.3.3 Scenario 3: Pareto-Optimized Hybrid

Processing allocation determined by the Pareto optimization algorithm based on:

- User-specified cost/time priorities
- Job urgency and deadline constraints
- Current queue depth and resource availability

Table 1: Three-Scenario Comparison Framework

Metric	**On-Premises**	**All Cloud**	**Optimized**
Processing Time	6-8 hours	30-60 minutes	1-3 hours
Cost per Game	$5-10	$50-80	$20-40
Scalability	Fixed	Unlimited	Dynamic
C3D Verification	Baseline	Required	Required

3.4 C3D Output Verification

Ensuring identical biomechanical output is critical for system validity. The verification process includes:

3.4.1 Hash Comparison

SHA-256 cryptographic hash of C3D file contents:

$$verify(c_{cloud}, c_{onprem}) = \begin{cases} \text{PASS} & \text{if } SHA256(c_{cloud}) = SHA256(c_{onprem}) \\ \text{FAIL} & \text{otherwise} \end{cases} \quad (7)$$

3.4.2 Structural Validation

Beyond byte-level comparison, structural validation confirms:

- Identical marker count (tracked body points)

- Identical frame count (temporal samples)
- Identical parameter blocks (metadata)

3.4.3 Trajectory Comparison

For cases where floating-point precision variations occur, trajectory RMSE provides tolerance-based comparison:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_{cloud,i} - p_{onprem,i})^2} \tag{8}$$

With acceptance threshold $RMSE < 0.1mm$.

3.5 Data Selection Methodology

Test data selection for unbiased testing follows a randomization process:

1. **Selector Pool**: A pool of 10 KinaTrax employees, other than the researcher, select 10 sessions each.
2. **Selection Criteria**: Established through a one-page requirements document that outlines:
   - Minimum number of events per session
   - Data completeness
   - Acceptable session types (games, laboratory sessions)
3. **Total Sample**: With a total of 100 observational sessions, the data is varied enough to include representation from:
   - Various MLB teams and environments
   - Game vs. training sessions
   - Player roles such as pitchers and batters
   - Various seasons

This methodology seeks to address concerns of bias that had arisen in committee review to ensure that the researcher is not selecting data that is favorable to the algorithm.

4 Presentation of Work

4.1 Dashboard Implementation

A decision support dashboard has been implemented and deployed for testing. Key implementation details include:

4.1.1 Architecture

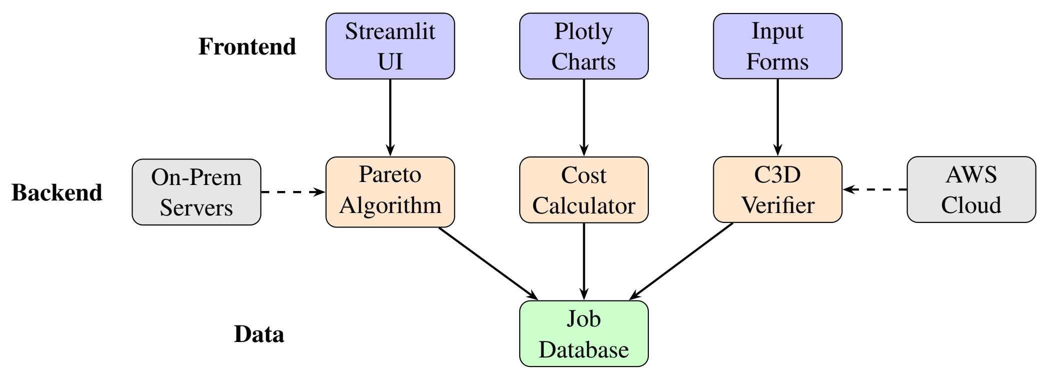

Figure 5: Dashboard system architecture

4.1.2 Code Organization

The implementation follows a modular structure:

```
src/dashboard/
+-- app/
|   +-- main.py              # Streamlit entry point
|   +-- data/
|   |   +-- schemas.py       # Pydantic models
|   |   +-- generators.py    # Mock data generation
```

```
|   +-- optimization/
|   |   +-- pareto.py         # Pareto algorithm
|   |   +-- scenarios.py      # Scenario comparison
|   +-- components/
|   |   +-- charts.py         # Plotly visualizations
|   |   +-- metrics.py        # KPI displays
|   +-- pages/
|       +-- 1_Event_Queue.py
|       +-- 2_Cost_Time_Tradeoff.py
|       +-- 3_Scenario_Comparison.py
|       +-- 4_Job_Configuration.py
|       +-- 5_C3D_Verification.py
+-- tests/
    +-- test_pareto.py       # Unit tests
```

4.1.3 Key Features Implemented

1. **Interactive Pareto Chart**: Allows hovering over points for viewing configuration information, selecting by clicking, and using weight sliders for changing the optimum suggestions.

2. **Real-Time Cost Estimation**: Allows simultaneous updating of cost estimates with AWS pricing models whenever there is a change in the configuration of a job.

3. **Scenario Cards**: Enable comparing three scenarios visually for key metrics such as total cost, total time, and number of processed jobs.

4. **C3D Verification Status**: The verification status of each processed job is denoted by green and red marks.

5. **Filtering and Search**: Enable the queue to be filtered by status, priority, team, or venue.

4.2 Current Progress

As of December 2025, the following milestones have been achieved:

Table 2: Project Milestone Status

Milestone	Status	Notes
Proposal Presentation	Complete	Nov 18, 2025
Project Pivot	Complete	Nov 24, 2025 - HCI focus
Hypothesis Revision	Complete	Dec 3, 2025 - Pareto optimization
Dashboard Frontend	Complete	Dec 10, 2025 - Streamlit implementation
IRB Documentation	Complete	Dec 7, 2025 - Exempt status
Data Selection Protocol	In Progress	Requirements document pending
Backend Containerization	Pending	Target: January 2026
Full System Testing	Pending	Target: January 2026

4.3 Next Steps

The remaining work includes:

1. **Data Selection Document**: Create one-page requirements for coworker data selection (due this week).
2. **AWS Containerization**: Deploy processing containers to AWS for cloud execution testing.
3. **Backend Integration**: Connect dashboard to real processing queue and cost data.
4. **100-Session Testing**: Execute three-scenario comparison on randomized test set.
5. **Results Analysis**: Statistical analysis of cost-time trade-offs and C3D verification.
6. **Publication**: Explore conference submission opportunities (pending employer approval).

5 Conclusion

This initial draft is an impetus for developing a decision support system for cloud migration trade-off points in the context of biomechanical processing. This is achieved through the application of Pareto frontier optimization techniques that enable data processing engineers to make informed decisions about the trade-off between cost and time.

This work is significant in that it offers:

- Pareto optimization for sports analytics cloud migration
- Interactive decision support system design
- Methodology for ensuring consistency of results among processing environments
- Unbiased data selection procedure for system evaluation

The total system test, set to occur in January of 2026, is to empirically verify the hypothesis with respect to the cost-time improvement opportunities using Pareto-optimized processing configuration.

References

Abdul, A., Vermeulen, J., Wang, D., Lim, B. Y., & Kankanhalli, M. (2018). Trends and trajectories for explainable, accountable and intelligible systems: An HCI research agenda. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 1–18.

Balalaie, A., Heydarnoori, A., & Jamshidi, P. (2016). Microservices architecture enables DevOps: Migration to a cloud-native architecture. *IEEE Software*, 33(3), 42–52.

Cleveland, W. S., & McGill, R. (1984). Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association*, 79(387), 531–554.

Dimara, E., & Perin, C. (2018). What is interaction for data visualization? *IEEE Transactions on Visualization and Computer Graphics*, 26(1), 119–129.

Ehrgott, M. (2005). *Multicriteria Optimization*. Springer Science & Business Media.

Genez, T. A., Bittencourt, L. F., & Madeira, E. R. (2012). Workflow scheduling for SaaS/PaaS cloud providers considering two SLA levels. *IEEE Network Operations and Management Symposium*, 906–912.

Gholami, M. F., Daneshgar, F., Low, G., & Beydoun, G. (2016). Cloud migration process—A survey, evaluation framework, and open challenges. *Journal of Systems and Software*, 120, 31–69.

Jamshidi, P., Ahmad, A., & Pahl, C. (2013). Cloud migration research: A systematic review. *IEEE Transactions on Cloud Computing*, 1(2), 142–157.

Khajeh-Hosseini, A., Greenwood, D., & Sommerville, I. (2010). Cloud migration: A case study of migrating an enterprise IT system to IaaS. *IEEE International Conference on Cloud Computing*, 450–457.

Liu, Z., Nersessian, N., & Stasko, J. (2014). Distributed cognition as a theoretical framework for information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6), 1173–1180.

Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. *IEEE Symposium on Visual Languages*, 336–343.

Sweller, J., Ayres, P., & Kalyuga, S. (2011). *Cognitive Load Theory*. Springer.

Yu, T., Zhang, Y., & Lin, K. J. (2013). Efficient algorithms for web services selection with end-to-end QoS constraints. *ACM Transactions on the Web*, 1(1), 1–26.

Zhan, Z. H., Liu, X. F., Gong, Y. J., Zhang, J., Chung, H. S. H., & Li, Y. (2015). Cloud computing resource scheduling and a survey of its evolutionary approaches. *ACM Computing Surveys*, 47(4), 1–33.