World Journal of Engineering Research
Print ISSN: 2959-9865
Online ISSN: 2959-9873
DOI: https://doi.org/10.61784/wjer4124

TRIAD: DEADLINE- AND RELIABILITY-AWARE SCHEDULING FOR POST-EARTHQUAKE STRUCTURAL ASSESSMENT WORKFLOWS

Dai Teng
University of Illinois Urbana-Champaign, Urbana 61801, IL, USA.

Abstract: Post-earthquake structural assessment requires timely execution of computational workflows under rapidly changing workloads, strict deadlines, and uncertain computing resource availability. Conventional scheduling approaches often prioritize deadlines without adequately accounting for overload and execution failures, potentially delaying high-priority assessments. This paper presents TRIAD (Triage with Reliability-Inflated Admission and Dispatch), an online scheduling framework that integrates failure-aware workload estimation, probabilistic deadline assessment, priority-based admission control, and reliability-aware task dispatch. TRIAD dynamically estimates worker failure conditions, adjusts expected execution requirements, and prioritizes assessment workflows according to their operational urgency and probability of timely completion. We evaluate TRIAD using trace-driven simulations constructed from building inventories in San Mateo and Alameda, California, a USGS earthquake scenario, FEMA Hazus v6.1 damage-and-loss information, and execution measurements from 446 OpenSees structural analyses. Workload arrivals, deadline constraints, and computing resource failures are simulated to examine scheduling performance under post-earthquake operating conditions. Compared with eight baseline scheduling methods, TRIAD achieves a weighted deadline satisfaction ratio of 0.875 on the San Mateo workload, exceeding the strongest baseline at 0.860 and conventional earliest-deadline-first scheduling at 0.260. It also reduces the weighted commitment-violation rate from 0.112 to 0.093 while maintaining sub-millisecond scheduling overhead. The results demonstrate the value of combining reliability estimation and overload-aware admission control for improving the timeliness and dependability of computational structural assessment workflows.
Keywords: Urgent computing; Workflow scheduling; Deadline; Reliability; Overload management; Regional earthquake loss assessment; Hazus

1 INTRODUCTION

Rapid regional assessment is one of the first computational activities after a damaging earthquake. Within minutes of the main shock, the U.S. Geological Survey (USGS) releases ShakeMap estimates of ground shaking [1]; loss-estimation tools such as FEMA Hazus combine such intensity fields with building inventories and fragility functions to estimate damage and losses [2-3]. Research platforms extend this pipeline with building-specific response simulation: the NHERI SimCenter framework orchestrates hazard mapping, structural response simulation and damage-and-loss estimation as a modular workflow that runs on high-performance and cloud resources [4], and city-scale nonlinear response-history analysis has been shown to be practical for regional loss estimation [5]. These workflows are computationally heavy: every building is analyzed under many ground-motion realizations, and the analyses are repeated whenever the shaking estimate is revised or an aftershock strikes.

The computing service that executes these workflows after an event faces three coupled difficulties. First, *load arrives in bursts*. A main-shock wave of requests covering the whole affected area is followed by re-runs triggered by ShakeMap revisions and by aftershocks, whose rate decays only slowly with time [6]. Second, *results have deadlines*. A damage estimate for a heavily shaken neighborhood is most useful while search-and-rescue and inspection teams are being dispatched; urgent computing systems such as SPRUCE were built precisely because late results in emergencies are of little use [7]. Third, *the platform is less reliable than usual*. Local data centers and edge resources near the disaster area suffer power and network disruptions, aftershocks cause correlated outages, and preemptible capacity is reclaimed under demand surges. Field data from production systems already show that node failures are frequent and bursty, with inter-failure times following a Weibull distribution with decreasing hazard [8]; post-disaster conditions make failures more frequent still.

Classical scheduling answers to these difficulties address them one at a time. Earliest deadline first (EDF) is optimal in underload but collapses in overload because it keeps serving jobs that can no longer finish, the well-known domino effect [9-13]. Value-based overload management sheds low-value work when the system is overloaded [12], but it treats execution as reliable and therefore commits to deadlines that node failures later break. Fault-tolerant workflow schedulers protect tasks with replication or primary–backup copies [14-16]; when capacity is the binding constraint, as it is right after a main shock, this redundancy consumes exactly the capacity that urgent jobs need. What is missing is a scheduler that reasons jointly about *which* jobs can still be delivered on time given the current failure environment,

© By the Author(s) 2025, under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0).

which jobs should be committed when not all of them can, and *where* committed work should run on a heterogeneous, failure-prone pool.

This paper develops such a scheduler for post-disaster structural assessment workflows. The scope is deliberately computational: the scheduler optimizes the timeliness and reliability with which assessment workflows are executed. It does not decide whether any building is safe to occupy; that judgment remains with the engineering procedures and field inspections that consume the computed results. Hazus-based damage estimates enter the scheduler only as urgency information that ranks assessment requests.

The contributions are as follows.

• Problem formulation. We formulate online scheduling of assessment workflows as maximizing the weighted fraction of jobs completed by their deadlines on a heterogeneous platform with stochastic task durations and fail-stop node failures, where job weights and deadline tiers are derived from Hazus expected damage and repair loss (Section 3).

• Reliability-inflated risk model. We derive a failure-inflated work and critical-path estimate from the closed-form expected execution time under fail-restart, combine it with a discounted Gamma–Poisson hazard estimator, and obtain a Poisson–Gaussian deadline-meeting probability that is monotone in both slack and hazard (Section 4).

• TRIAD scheduler. We design a probability-weighted Moore–Hodgson triage that commits jobs on a fluid EDF schedule in $O(n\log n)$ time and reduces to the optimal Moore–Hodgson rule in the deterministic single-machine limit, together with a risk-aware dispatcher that places committed work on the nodes maximizing deadline-meeting probability.

• Real-data evaluation. We build an evaluation pipeline from real inventories, a USGS scenario ShakeMap, Hazus v6.1 tables, and measured OpenSees analysis costs, and compare TRIAD with eight baselines through main, ablation, sensitivity and robustness experiments with paired statistical tests (Sections 5–6) (Figure 1).

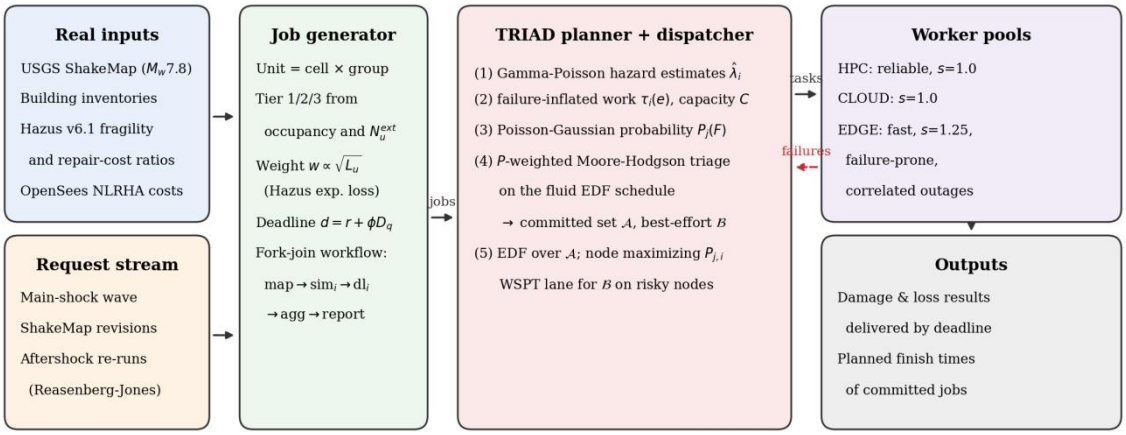


Figure 1 Overview of the Evaluation Pipeline and of TRIAD. Real Inputs Define Assessment Jobs; TRIAD Plans Commitments with a Reliability-Inflated Risk Model and Dispatches Tasks to Heterogeneous, Failure-Prone Worker Pools Whose Observed Failures Feed Back into the Hazard Estimates

2 RELATED WORK

2.1 Regional Seismic Assessment and Urgent Computing

Hazus provides a standardized methodology for estimating earthquake damage and loss over building portfolios using fragility functions of model building types and design levels [2-3]. ShakeMap supplies the rapidly updated intensity fields that drive such estimates [1]. The NHERI SimCenter workflow integrates inventories, hazard mapping, response simulation and damage-and-loss models into a modular, computationally scalable framework [4], and Lu et al. demonstrated regional loss estimation with city-scale nonlinear time history analysis built on OpenSees [5, 17]. These works focus on the engineering models and on workflow composition; the scheduling of many concurrent assessment workflows under deadlines and failures has received far less attention. Urgent computing research addressed the provisioning side: SPRUCE introduced right-of-way tokens that give emergency jobs elevated priority on shared supercomputers [7], and Trebon and Beckman derived empirical probabilistic upper bounds on completion times of urgent applications [18]. TRIAD is complementary: it assumes a pool of resources has been obtained and decides how to use it when the demand exceeds what the pool can deliver on time.

2.2 Real-Time Scheduling under Overload

EDF is optimal on a single processor whenever a feasible schedule exists [9-10], and least-laxity scheduling extends this reasoning to multiprocessors [19]. In overload, deadline-driven priorities degrade sharply. Jensen, Locke and Tokuda introduced value functions and value-density heuristics for overloaded real-time systems [11]; Buttazzo, Spuri and Sensini compared value-driven and deadline-driven policies and found that deadline scheduling in underload combined with value-based rejection in overload performs best in practice [12]; Koren and Shasha proved competitive

bounds for on-line overload scheduling with D ^{over} [13]. In operations research, the Moore–Hodgson algorithm minimizes the number of late jobs on a single machine by processing jobs in deadline order and discarding the longest job whenever a deadline is violated [20]. TRIAD generalizes this rejection logic to weighted, failure-prone, parallel workflows by replacing deterministic processing times with failure-inflated work and deterministic feasibility with a deadline-meeting probability.

2.3 Deadline- and Reliability-Aware Workflow Scheduling

List-scheduling heuristics such as HEFT map task graphs onto heterogeneous processors to minimize makespan [21]; deadline-constrained variants for clouds trade cost against completion time [22]; workflow management systems such as Pegasus handle task retries and clustering in production [23]. Reliability-aware schedulers maximize the probability of failure-free execution [16], minimize the redundancy needed to reach a reliability goal [15], or combine primary–backup copies with elastic provisioning for real-time workflows [14]. Checkpointing theory provides closed-form expected execution times on failure-prone platforms [24]. These methods assume either sufficient capacity to protect every task or a fixed, feasible set of deadlines. Post-disaster assessment violates both assumptions: demand bursts exceed capacity, so the scheduler must decide which deadlines to pursue, and the failure environment changes during the response, so reliability must be estimated online. TRIAD integrates these two decisions in a single probabilistic model.

3 SYSTEM MODEL AND PROBLEM FORMULATION

3.1 Assessment Jobs and Workflows

The affected region is partitioned into spatial *units*, each a grid cell of roughly 0.55×0.53 km combined with an asset group: critical facilities (educational, government, hospital and medical-office occupancies), other non-single-family buildings, and single-family residences. An assessment job j requests the regional workflow for one unit. Following the structure of SimCenter regional workflows, the job is a fork–join directed acyclic graph $G_j = (V_j, E_j)$: a hazard-mapping task samples intensity measures at the building sites; k_j parallel response-simulation tasks each run nonlinear response-history analyses (NLRHA) for a chunk of buildings under R ground-motion realizations; each simulation task feeds a damage-and-loss task; an aggregation task and a reporting task close the job [4]. Chunks are sized so that the simulation work of one task is close to a target $\bar{c}$. Each task v has an *estimated* work e_v (seconds on a unit-speed node), known to the scheduler, and an *actual* work $a_v = e_v\varepsilon_v$ with $\ln\varepsilon_v \sim \mathcal{N}(-\varsigma^2/2, \varsigma^2)$ and $\varsigma^2 = \ln(1 + \mathrm{CV}^2)$, revealed only at completion.

Job j is released at r_j, has deadline d_j and weight w_j. Requests come from three sources: the main-shock wave, ShakeMap revisions that re-run the units of tiers 1 and 2, and aftershocks that re-run the units close to their epicenters.

3.2 Urgency Weights and Deadline Tiers from Hazus

For building b with peak ground acceleration PGA_b from the ShakeMap, the Hazus PGA-based fragility of its model building type and design level gives

$$P_b(\mathrm{DS} \geq k) = \Phi\left(\frac{\ln(\mathrm{PGA}_b/\theta_k)}{\beta_k}\right), \quad k = 1, ..., 4, \tag{1}$$

for the slight, moderate, extensive and complete damage states. With repair-cost ratios $\mathrm{LR}_{o,k}$ of occupancy o and replacement cost RC_b, the expected repair loss of unit u is $L_u = \sum_{b\in u}\sum_k P_b(\mathrm{DS} = k)\,\mathrm{LR}_{o(b),k}\,\mathrm{RC}_b$, and the expected number of heavily damaged buildings is $N_u^{\mathrm{ext}} = \sum_{b\in u} P_b(\mathrm{DS} \geq 3)$. Tier 1 contains critical-facility units and the 15% of other units with the largest N_u^{ext}; tier 2 contains the remaining non-single-family units; tier 3 contains the remaining single-family units. A job of tier q released at r_j receives deadline $d_j = r_j + \phi D_q$ with $(D_1, D_2, D_3) = (45, 90, 180)$ min and deadline factor ϕ. The weight is

$$w_j \propto \iota_u\sqrt{L_u/\bar{L}}, \tag{2}$$

where $\iota_u = 3$ for critical-facility units and $\iota_u = 1$ otherwise, normalized to unit mean; the square root keeps a few high-value commercial properties from dominating the objective. These quantities rank computational requests; they are not used to judge the safety of individual buildings.

3.3 Platform and Failure Model

The platform consists of M worker nodes in pools that differ in speed and reliability: an on-premise high-performance partition (HPC), on-demand cloud instances (CLOUD), and fast but exposed regional edge or preemptible capacity (EDGE). Node i has speed s_i and runs one task at a time. Up-times follow a Weibull distribution with shape $k_W < 1$,

consistent with the decreasing hazard observed in production failure logs [8], and repairs are log-normal. Strong aftershocks near the region can additionally take down the whole EDGE pool at once. A failure destroys the running task; the scheduler detects it after a heartbeat delay δ, after which the task is re-executed from scratch. Launching on a node that has failed but has not yet been detected fails immediately and marks the node as down. Execution of task copies is non-preemptive.

3.4 Objective and Metrics

Let C_j be the completion time of job j. The primary objective is the weighted deadline satisfaction ratio

$$\mathrm{WDSR} = \frac{\sum_j \; w_j \, \mathbb{1}[C_j \le d_j]}{\sum_j \; w_j}, \tag{3}$$

which the scheduler maximizes online without knowledge of future arrivals, failures or actual task durations. We also report the tier-1 WDSR, the unweighted deadline satisfaction ratio (DSR), the weighted mean tardiness $\sum_j \; w_j \max\big(0, C_j - d_j\big)/\sum_j \; w_j$, the lost work destroyed by failures, and the *commitment-violation rate* (CVR): the weighted fraction of jobs that the scheduler commits to at its first planning epoch after release but that still miss their deadline. Schedulers without admission control implicitly commit to every job. CVR measures how trustworthy the scheduler's promises are, which matters to an operations center that plans around announced result times.

The offline version of the problem is NP-hard even on one machine without failures: with a common deadline D, selecting jobs of maximum total weight whose processing times fit in D is the 0/1 knapsack problem. TRIAD therefore pursues a principled heuristic built on an explicit risk model.

4 THE TRIAD SCHEDULER

TRIAD alternates between a *planning* step, executed every $\Delta_p = 5$ min, on job arrivals (at most every 30 s) and on detected node state changes, and a *dispatch* step executed whenever a node becomes idle. Planning updates hazard estimates, computes failure-inflated work and deadline-meeting probabilities, and partitions active jobs into a committed set $\mathcal{A}$ and a best-effort set $\mathcal{B}$. Dispatch assigns ready tasks to idle nodes.

4.1 Online Hazard Estimation

The nominal mean time between failures (MTBF) of a pool is known before the event but becomes unreliable once the disaster degrades infrastructure. TRIAD maintains, for each node i, an exponentially discounted failure count and exposure,

$$\begin{aligned} n_i(t) \quad &= \sum_{f \in \mathcal{F}_i} e^{-\kappa\left(t - t_f\right)}, \\ U_i(t) \quad &= \int_0^t \mathbb{1}\,[i\mathrm{up\ at}u]\; e^{-\kappa(t-u)}\, du, \end{aligned} \tag{4}$$

with $\kappa = \ln 2/H$ for a half-life H, and uses the posterior mean of a Gamma prior with shape α_0 and rate $\beta_{0,i} = \alpha_0 \,\mathrm{MTBF}_i^{\mathrm{nom}}$, partially pooled with the pool averages $\bar{n}_{P(i)}, \bar{U}_{P(i)}$:

$$\hat{\lambda}_i = \frac{\alpha_0 + n_i + \bar{n}_{P(i)}}{\beta_{0,i} + U_i + \bar{U}_{P(i)}}. \tag{5}$$

Pooling lets a correlated pool outage raise the estimates of all nodes in that pool at once; discounting lets the estimates follow the non-stationary post-event environment. Repair times are tracked per pool with exponential smoothing, giving $\widehat{\mathrm{MTTR}}_{P(i)}$.

4.2 Failure-Inflated Work

Lemma 1. *Let a task of work e run on node i whose failures form a Poisson process of rate λ, and let each failure cost a downtime δ before re-execution from scratch. The expected completion time is*

$$\tau_i(e) = \left(\frac{1}{\lambda} + \delta\right)\left(e^{\lambda e / s_i} - 1\right). \tag{6}$$

Proof. Let $\omega = e/s_i$ and $X \sim \mathrm{Exp}(\lambda)$ be the time to the next failure. Conditioning on whether $X < \omega$ gives the renewal equation $\mathbb{E}[T] = \mathbb{E}[\min(X, \omega)] + \Pr[X < \omega](\delta + \mathbb{E}[T])$. With $\mathbb{E}[\min(X, \omega)] = \left(1 - e^{-\lambda\omega}\right)/\lambda$ and $\Pr[X < \omega] = 1 - e^{-\lambda\omega}$, solving for $\mathbb{E}[T]$ yields (6). The same expression appears in the checkpointing analysis of [24] without checkpoints. □

As $\lambda \to 0$, $\tau_i(e) \to e/s_i$, and the inflation grows exponentially with the exposure $\lambda e/s_i$, which penalizes long tasks on unreliable nodes. For planning, TRIAD uses the capacity-weighted mean hazard $\bar{\lambda} = \sum_{i \in \mathcal{U}} s_i \, \hat{\lambda}_i / \sum_{i \in \mathcal{U}} s_i$ and mean speed $\bar{s}$ over the set $\mathcal{U}$ of nodes believed up. The failure-inflated remaining work of job j is $\widetilde{W}_j = \sum_{v \in \mathcal{R}_j} \bar{s} \; \bar{\tau}(e_v)$,

where $\mathcal{R}_j$ are its unfinished tasks (running tasks counted by their remaining estimate) and $\bar{\tau}$ is (6) evaluated with $(\bar{\lambda}, \bar{s})$. The remaining critical path L_j is computed on the residual DAG and inflated proportionally, $\tilde{L}_j = L_j \widetilde{W}_j / W_j$. The effective capacity accounts for downtime,

$$C = \sum_{i \in \mathcal{U}} \frac{s_i}{1 + \hat{\lambda}_i \widehat{\mathrm{MTTR}}_{P(i)}}. \tag{7}$$

4.3 Deadline-Meeting Probability

Given a planned finish time F, failures add delay while runtime errors add noise. Let B_j be the number of remaining parallel branches and $\nu_j = \bar{\lambda} \widetilde{W}_j / \bar{s}$ the expected number of failures hitting the job. Failures are spread over branches, and a branch hit k times is delayed by about $k\Delta_j$ with $\Delta_j = \delta + e_j^{\max}/(2\bar{s})$ (detection plus, on average, half of the largest remaining task). The number of delay units of the slowest branch, $K_j = \max_b N_b$ with $N_b \sim \mathrm{Poisson}\left(\nu_j / B_j\right)$, has distribution $G_j(k) = \Pr\left[K_j \le k\right] = F_{\mathrm{Pois}}\left(k; \nu_j / B_j\right)^{B_j}$. Modeling the remaining runtime uncertainty as Gaussian with $\sigma_j(F)^2 = \left(c_v L_j / \bar{s}\right)^2 + \left(\kappa_p (F - t)\right)^2 + \sigma_0^2$, the deadline-meeting probability is

$$P_j(F) = \sum_{k=0}^{3} \left(G_j(k) - G_j(k-1)\right) \Phi\left(\frac{d_j - F - k\Delta_j}{\sigma_j(F)}\right), \tag{8}$$

with $G_j(-1) = 0$; outcomes with more than three delay units are treated as misses.

Proposition 1. *For a fixed dispersion σ_j, $P_j(F)$ is non-increasing in the planned finish time F and in the hazard $\bar{\lambda}$.*

Proof. Write $P_j(F) = \mathbb{E}\left[\psi\left(K_j\right)\right]$ with $\psi(k) = \Phi\left(\left(d_j - F - k\Delta_j\right)/\sigma_j\right) \mathbb{1}[k \le 3]$. For fixed σ_j, ψ is non-increasing in k and, for every k, non-increasing in F, which proves the first claim. Increasing $\bar{\lambda}$ increases ν_j; each $N_b \sim \mathrm{Poisson}\left(\nu_j / B_j\right)$ is stochastically increasing in its mean, hence so is their maximum K_j, and the expectation of the non-increasing function ψ cannot increase. □

Proposition 1 makes the triage consistent: a higher estimated hazard or a later planned finish never makes a job more attractive to commit. In the planner, σ_j grows with the planning horizon to reflect the larger uncertainty of distant predictions; this widening is evaluated once per epoch and does not affect the ordering argument above.

4.4 Probability-Weighted Moore–Hodgson Triage

TRIAD commits jobs on a fluid EDF schedule. Sorting active jobs by deadline, job j's planned finish is

$$F_j = t + \max\left(\frac{\tilde{L}_j}{\bar{s}}, \frac{1}{C} \sum_{m \in \mathcal{A},\, d_m \le d_j} \widetilde{W}_m\right). \tag{9}$$

Algorithm 1 processes jobs in deadline order and keeps the committed jobs in a min-heap keyed by their *success-weighted value density*

$$\rho_j = \frac{w_j P_j\left(t + \tilde{L}_j / \bar{s}\right)}{\widetilde{W}_j}, \tag{10}$$

where the probability is evaluated as if j ran alone. A job whose solo probability is below the threshold θ cannot be saved and goes to $\mathcal{B}$ directly. Otherwise j is inserted; while $P_j\left(F_j\right) < \theta$, the committed job of lowest density is removed and moved to $\mathcal{B}$, which reduces the fluid load ahead of j. Jobs removed earlier are never re-inserted within the same epoch, but every epoch re-plans from scratch, so a job can be committed again when failures abate or work completes.

Algorithm 1: TRIAD planning epoch at time t

1. Update $\hat{\lambda}_i$ by (5); compute $\bar{\lambda}$, $\bar{s}$ and C by (7).
2. For every active job j, compute W_j, $\widetilde{W}_j$, L_j, $\tilde{L}_j$, B_j and $e_j^{\max}$.
3. Set $\mathcal{A} = \mathcal{B} = \emptyset$, an empty min-heap $\mathcal{H}$, and $D = 0$.
4. For each job j in non-decreasing order of d_j: compute $P_j^{\mathrm{solo}} = P_j\left(t + \tilde{L}_j / \bar{s}\right)$ and $\rho_j = w_j P_j^{\mathrm{solo}} / \widetilde{W}_j$; if $d_j \le t$ or $P_j^{\mathrm{solo}} < \theta$, put j in $\mathcal{B}$ and continue with the next job.
5. Push j into $\mathcal{H}$ with key ρ_j, add it to $\mathcal{A}$, and set $D \leftarrow D + \widetilde{W}_j$.
6. While $P_j\left(t + \max\left(\tilde{L}_j / \bar{s}, D/C\right)\right) < \theta$: pop the minimum-density job m from $\mathcal{H}$, move it from $\mathcal{A}$ to $\mathcal{B}$, set $D \leftarrow D - \widetilde{W}_m$, and stop if $m = j$.
7. Recompute F_j by (9) and $P_j\left(F_j\right)$ for all $j \in \mathcal{A}$; return $\mathcal{A}$ and $\mathcal{B}$.

Proposition 2. *Consider a single unit-speed node ($C = \bar{s} = 1$), no failures ($\bar{\lambda} = 0$), exact runtimes ($\sigma_j \to 0$), equal weights, all jobs available at time t, and $\theta \in (0,1)$. Then Algorithm 1 commits a maximum-cardinality set of on-time jobs.*

Proof. With $\bar{\lambda} = 0$ and $\sigma_j \to 0$, $P_j(F) = \mathbb{1}[F \le d_j]$ and $\widetilde{W}_j = W_j \ge L_j$, so $F_j = t + \sum_{m \in \mathcal{A}, d_m \le d_j} W_m$ is the EDF completion time on one machine. A job that is feasible alone has $P_j^{\text{solo}} = 1$, so with equal weights $\rho_j = 1/W_j$ and popping the minimum-density job removes the longest job among those committed so far. The algorithm is then exactly the Moore–Hodgson procedure, which is optimal for minimizing the number of late jobs on a single machine [20]. □

Proposition 3. *A planning epoch with n active jobs, M nodes and $|V|$ unfinished tasks runs in $O(n\log n + |V| + M)$ time.*

Proof. Hazard updates cost $O(M)$ and per-job quantities $O(|V|)$ in total. Sorting costs $O(n\log n)$. Each job is pushed into and popped from the heap at most once per epoch, giving $O(n\log n)$ for the loop, and each probability evaluation in (8) uses a constant number of terms. □

4.5 Risk-Aware Dispatch

Whenever nodes are idle, TRIAD orders the ready tasks of committed jobs by EDF, which is consistent with the fluid schedule assumed in (9). For a ready task v of committed job j and idle node i, it estimates the job's deadline-meeting probability if v runs on i,

$$P_{j,i} = p_i \, \Phi\left(\frac{d_j - F_{j,i}}{\sigma_j}\right) + (1 - p_i)\, \Phi\left(\frac{d_j - F_{j,i} - \Delta_{j,i}}{\sigma_j}\right), \tag{11}$$

where $p_i = e^{-\hat{\lambda}_i e_v / s_i}$ is the probability that the copy survives, $F_{j,i} = F_j + e_v/s_i - e_v/\bar{s}$ adjusts the planned finish for the node speed, and $\Delta_{j,i} = \delta + e_v/(2s_i) + e_v/\bar{s}$ is the delay of a failure followed by re-execution. The task goes to the node with the largest $P_{j,i}$, with ties broken by the smallest $\tau_i(e_v)$. When the job has ample slack all nodes give $P_{j,i} \approx 1$ and the fastest expected completion wins; when slack is short, the reliability of the node dominates. After committed work is placed, best-effort jobs are served in weighted-shortest-work order $w_j/\widetilde{W}_j$, which reduces their weighted tardiness, and their tasks are placed on the idle node with the largest $\tau_i(e_v)$, so that reliable nodes remain free for committed work that arrives next.

5 EXPERIMENTAL SETUP

5.1 Real Data

Inventories. We use two parcel-level building inventories distributed with the NHERI SimCenter R2D examples [25]: San Mateo, California, with 21,729 buildings, and Alameda, California, with 14,166 buildings. Each record provides location, number of stories, year built, occupancy class, Hazus structure type, design level, plan area and replacement cost. Missing design levels are assigned by construction era (pre-1941 Pre-Code, 1941–1974 Moderate-Code, 1975 and later High-Code).

Hazard. Ground shaking comes from the USGS ShakeMap of the M_w 7.8 northern San Andreas scenario (event identifier npsanandreassansapsasm7p8_se; North Coast, Peninsula and Santa Cruz Mountains segments), a 421×281 grid of PGA, PGV, MMI and spectral accelerations [1, 25]. Intensities are bilinearly interpolated to every building.

Fragility and loss. Hazus v6.1 PGA-based building fragilities, roof-drift fragilities and repair-cost ratios are taken from the SimCenter Damage and Loss Model Library [3, 26]. Table 1 summarizes the resulting regions and workloads, and Figure 2(a) shows the San Mateo inventory, the PGA field and the assessment units.

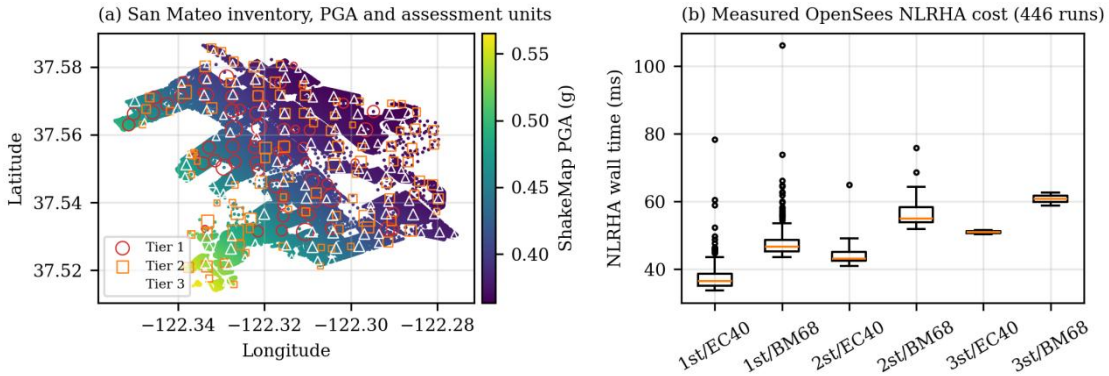


Figure 2 (a) San Mateo Buildings Colored by ShakeMap PGA, with Assessment Units by Tier (Marker Size Proportional to the Hazus-Derived Weight). (b) Measured Wall Time of the OpenSees NLRHA by Number of Stories and Ground-Motion Record

Table 1 Study Regions and Generated Assessment Workloads (Seed 0)

Region	Buildings	Mean PGA (g) (range)	Mean MMI	Exp. loss (M$)	Units (tiers)	Jobs	Tasks	Work (node-h)
San Mateo, CA	21,729	0.411 (0.36–0.57)	8.76	2,647.7	259 (63/94/102)	583	2,923	48.3
Alameda, CA	14,166	0.288 (0.24–0.33)	7.95	101.9	160 (45/63/52)	536	2,752	53.1

Note: PGA/MMI from the USGS ShakeMap scenario; expected repair loss from Hazus v6.1 PGA-based fragilities; units listed as total (tier 1/2/3).

5.2 Measured Workflow Costs

Task durations are derived from measurements rather than assumed. For the response-simulation stage we built, for 446 sampled buildings–record pairs stratified over all structure types of both inventories, a nonlinear multi-degree-of-freedom shear-building model in OpenSeesPy [17], in the spirit of city-scale NLRHA [5]: one lumped mass per story, story springs with bilinear (Steel01) or pinched hysteretic (wood) behavior, a fundamental period from the approximate-period relation $T_a = C_t h_n^x$ of ASCE 7-16 [27], a design-level-dependent yield base-shear coefficient, and 5% Rayleigh damping. Each model was analyzed with the Newmark average-acceleration method under one of two historical records—the 1940 El Centro north–south component (PGA 0.32 g, 31.2 s) and the 1968 Borrego Mountain record at El Centro (40 s)—scaled to the site PGA perturbed by a log-normal factor ($\sigma_{\ln} = 0.6$). The measured cost per analysis ranged from 34 to 106 ms (Figure 2(b)); a linear model in the number of time steps, steps×stories and lnPGA explains 55% of the variance with a residual coefficient of variation of 0.13, and yields 43, 50 and 58 ms per analysis for one-, two- and three-story buildings. Hazard mapping with four-neighbor sampling, damage-and-loss sampling and aggregation were measured on the full inventories at 20, 13 and 5 μs per building for 100 realizations, and launching a task (a fresh interpreter importing the workflow stack) took 1.02 s on average. With $R = 100$ realizations per building, matching the realization count of the SimCenter ShakeMap example [25], and a chunk target $\bar{c} = 30$ min, a San Mateo main-shock wave requires roughly 26 node-hours of NLRHA.

5.3 Load, Platform and Failures

Main-shock jobs are released uniformly within 5–35 min after the event. ShakeMap revisions at 45 min and 3 h re-run all tier-1 and tier-2 units. Aftershocks with $M \geq 5$ are generated for 24 h from the Reasenberg–Jones model $\lambda(t, M) = 10^{a+b(M_m - M)}(t + c)^{-p}$ with the generic California parameters $a = -1.67$, $b = 0.91$, $p = 1.08$, $c = 0.05$ d [6]; each aftershock occurs near the study region with probability 0.25, and units within $10^{0.5M-2.2}$ km of its epicenter are re-assessed 5–20 min later. The base platform has nine nodes: three HPC nodes (speed 1.0, MTBF 72 h, median repair 60 min), three CLOUD nodes (1.0, 12 h, 10 min) and three EDGE nodes (1.25, 2 h, 10 min). Up-times are Weibull with $k_W = 0.7$, repairs log-normal with $\sigma_{\ln} = 0.8$, and a nearby aftershock of $M \geq 5.5$ takes down the EDGE pool with probability 0.5 for a log-normal duration with median 30 min. Failure detection takes $\delta = 60$ s, and runtime estimates have CV = 0.2. Figure 3 shows one realization: the offered work rate exceeds the available capacity several-fold during the main-shock wave and again after ShakeMap revisions and nearby aftershocks, while EDGE nodes fail repeatedly. Figure 1 summarizes the complete evaluation pipeline.

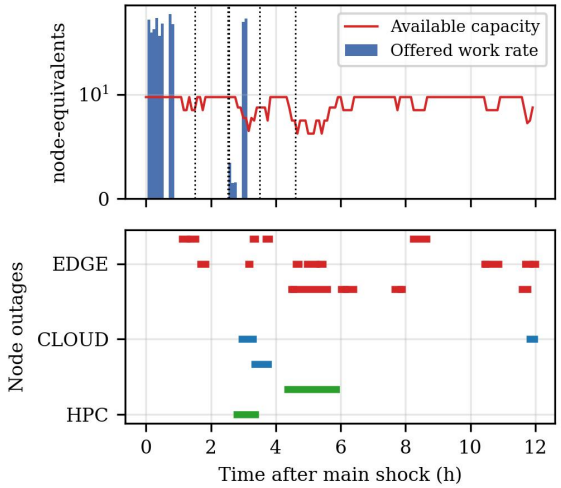


Figure 3 San Mateo Workload, Seed 0. Top: Offered Work Rate (5-Min Bins) versus Available Capacity; Dotted Lines Mark Nearby Aftershocks. Bottom: Node Outages by Pool

5.4 Baselines and Protocol

We compare eight baselines that cover queue-based, deadline-based, value-based and reliability-based scheduling. FIFO serves jobs in release order. SP serves tiers in static priority order and releases in FIFO order within a tier. EDF [9] and LLF [19] order jobs by deadline and by laxity $d_j - L_j/\bar{s}$. HVDF serves the highest value density w_j/W_j first [11]. RED-V is EDF with value-based overload shedding in the spirit of [12]: it runs the same Moore–Hodgson triage with nominal, failure-free work, raw capacity and a deterministic feasibility test. EDF-RA combines EDF with reliability-driven mapping to the idle node of lowest nominal hazard [16]. RC-EDF combines EDF with active replication sized to a reliability goal: each task receives the smallest number of copies $k \le 3$ for which $1-(1-p)^k \ge 0.999$ under nominal hazards, copies go to the most reliable idle nodes, and the first finished copy cancels the others [14-15]. Unless noted, baselines place tasks on the fastest idle node, as in earliest-finish-time list scheduling [21]. All policies re-execute lost tasks after detection, as production workflow systems retry failed jobs [23].

TRIAD uses $\theta = 0.5$, half-life $H = 6$ h, prior weight $\alpha_0 = 1$, $c_v = 0.2$, $\kappa_p = 0.15$ and $\sigma_0 = 60$ s. All policies see identical workloads and failure traces for a given seed. Main and ablation experiments use 20 seeds; sensitivity and robustness experiments use 10 seeds. We report means with 95% Student-t confidence intervals and test differences with the paired Wilcoxon signed-rank test. The simulator, the profiling scripts and all raw results are released with the paper.

6 RESULTS

6.1 Main Comparison

Table 2 and Figure 4 report the main results. On San Mateo, TRIAD reaches a WDSR of 0.875, the highest among all policies, followed by RED-V (0.860) and HVDF (0.786). The deadline-driven policies without admission control collapse under the main-shock burst: EDF obtains 0.260 and LLF 0.117, because jobs that can no longer finish keep receiving nodes and push other jobs past their deadlines. Static tier priority (SP, 0.463) protects some urgent work but serves each tier in arrival order and cannot shed hopeless jobs. Reliability mechanisms alone do not help in this regime: EDF-RA (0.260) behaves like EDF, and RC-EDF (0.226) is worse than EDF because replicas consume the capacity that the overloaded system lacks. TRIAD also achieves the best tier-1 WDSR (0.835 versus 0.813 for RED-V), the best unweighted DSR and the lowest weighted tardiness, so the gain reaches the most consequential assessments. The paired Wilcoxon tests confirm that TRIAD's WDSR advantage over every baseline, including RED-V, is significant at $p < 0.001$ across the 20 seeds.

Table 2 Main Results on the San Mateo Workload (9 Nodes, 20 Seeds; Mean±95% CI)

Policy	WDSR ↑	Tier-1 WDSR ↑	DSR ↑	W. tardiness (h) ↓	CVR ↓	Lost work (node·h)	p (WDSR)
FIFO	0.247±0.034	0.086±0.013	0.307±0.042	1.521±0.503	0.753±0.034	2.07	<0.001
SP	0.463±0.079	0.638±0.086	0.408±0.081	0.618±0.347	0.537±0.079	1.93	<0.001
EDF	0.260±0.058	0.390±0.057	0.220±0.061	0.834±0.387	0.740±0.058	2.15	<0.001
LLF	0.117±0.043	0.184±0.060	0.094±0.042	1.127±0.460	0.883±0.043	2.10	<0.001
HVDF	0.786±0.029	0.707±0.027	0.832±0.027	0.392±0.161	0.214±0.029	1.93	<0.001
RED-V	0.860±0.027	0.813±0.026	0.893±0.024	0.390±0.164	0.112±0.016	2.14	<0.001
EDF-RA	0.260±0.059	0.389±0.060	0.220±0.062	0.829±0.387	0.740±0.059	2.04	<0.001
RC-EDF	0.226±0.048	0.359±0.057	0.184±0.047	0.869±0.379	0.774±0.048	2.01	<0.001
TRIAD	**0.875±0.027**	**0.835±0.028**	**0.903±0.024**	**0.375±0.163**	**0.093±0.015**	2.05	–

Note: p: Wilcoxon signed-rank test of WDSR against TRIAD.

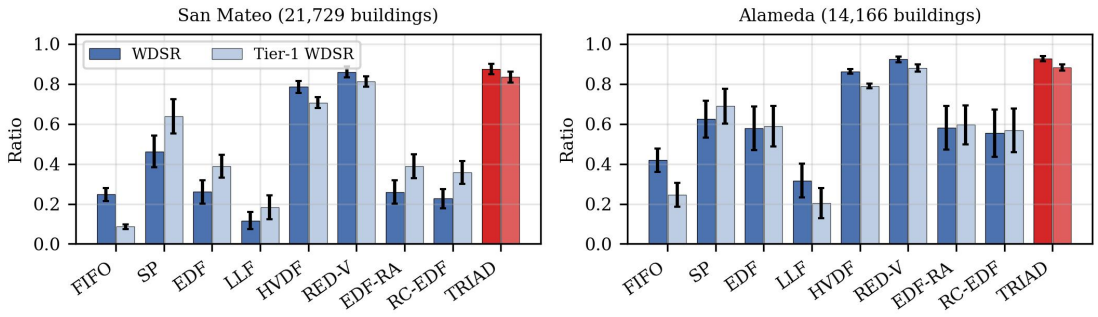


Figure 4 WDSR and Tier-1 WDSR of All Policies on the Two Regions (20 Seeds, 95% CIs). TRIAD in Red

Table 3 repeats the comparison on Alameda with the same platform. The Alameda inventory is smaller, so its main-shock wave requires about 18 node-hours of NLRHA instead of 26, and the platform is less overloaded during the critical first hour. All overload-aware policies perform well: TRIAD obtains the highest mean WDSR (0.926) together

with the best tier-1 WDSR, DSR and CVR, with RED-V close behind (0.923, $p = 0.154$), against 0.578 for EDF and 0.863 for HVDF. The margin between the two triage-based policies narrows when fewer commitment decisions lie at the margin of feasibility, while the ordering of the policy families is preserved across regions.

Table 3 Main Results on the Alameda Workload (Same Platform, 20 Seeds)

Policy	WDSR ↑	Tier-1 WDSR ↑	DSR ↑	W. tardiness (h) ↓	CVR ↓	Lost work (node·h)	p (WDSR)
FIFO	0.419±0.059	0.246±0.059	0.516±0.063	0.620±0.230	0.581±0.059	1.80	<0.001
SP	0.624±0.093	0.689±0.087	0.590±0.101	0.257±0.183	0.376±0.093	1.91	<0.001
EDF	0.578±0.109	0.589±0.101	0.569±0.115	0.266±0.189	0.422±0.109	1.89	<0.001
LLF	0.317±0.085	0.203±0.075	0.387±0.098	0.443±0.233	0.683±0.085	1.98	<0.001
HVDF	0.863±0.013	0.790±0.012	0.910±0.009	0.192±0.062	0.137±0.013	1.85	<0.001
RED-V	0.923±0.015	0.880±0.018	0.950±0.010	**0.172±0.060**	0.060±0.009	1.91	0.154
EDF-RA	0.582±0.109	0.595±0.099	0.571±0.116	0.258±0.180	0.418±0.109	1.83	<0.001
RC-EDF	0.554±0.118	0.568±0.109	0.547±0.123	0.285±0.192	0.446±0.118	1.87	<0.001
TRIAD	**0.926±0.013**	**0.883±0.016**	**0.952±0.009**	0.173±0.058	**0.057±0.008**	1.87	–

6.2 Reliability of Commitments

A practical requirement of an operations center is that announced result times hold. Figure 5 plots WDSR against the commitment-violation rate. Policies without admission control commit to every job and therefore violate a large share of their implicit commitments (CVR of 0.740 for EDF). RED-V reduces violations to 0.112 by shedding jobs, yet its deterministic, failure-free feasibility test still promises completion times that failures later break. TRIAD lowers CVR to 0.093 while also completing more weighted work on time: the failure-inflated work and the deadline-meeting probability make the commitments both more accurate and more valuable.

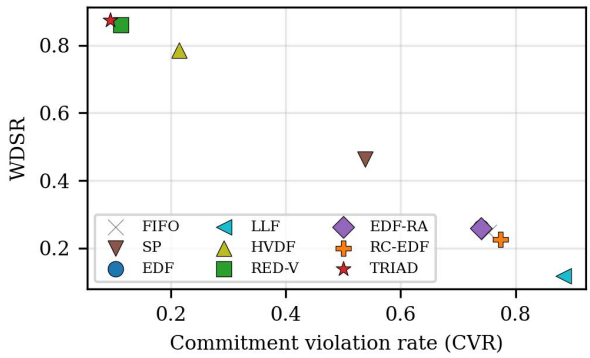


Figure 5 Trade-Off between WDSR and Commitment-Violation Rate (San Mateo, 20 Seeds). Upper-Left Is Better

6.3 Ablation Study

Table 4 removes one component at a time under the base failure environment and under a severe environment in which all MTBFs are divided by four while the scheduler's priors remain nominal. Triage is the foundation: without it WDSR drops from 0.875 to 0.260, because the dispatcher falls back to plain EDF over all jobs. Failure inflation is the most important reliability component: removing it reduces WDSR to 0.866 (base) and 0.828 (severe) and raises CVR, since the triage then overestimates the useful capacity. Online hazard estimation contributes further gains, most visibly when the nominal priors are wrong (0.831 versus 0.838 under severe failures), and the success-weighted density of (10) improves the choice of which jobs to shed. Risk-aware placement leaves WDSR unchanged in these workloads (0.875): during bursts a single node typically becomes idle at a time, so the placement choice is rarely open, and the commitment decisions of the triage determine the outcome. RED-V, which lacks the whole reliability model, is consistently below the full TRIAD (0.860 and 0.821).

Table 4 Ablation of TRIAD (San Mateo, 20 Seeds, Mean±95% CI)

	Base failures			Severe failures (*0.25* ×MTBF)		
Variant	WDSR	CVR	W.T.(h)	WDSR	CVR	W.T.(h)
TRIAD (full)	0.875±0.027	0.093	0.375	0.838±0.030	0.112	0.582
w/o triage	0.260±0.058	0.740	0.831	0.168±0.044	0.832	1.456
w/o failure inflation	0.866±0.027	0.107	0.383	0.828±0.030	0.134	0.599
w/o online hazard estimation	0.871±0.027	0.100	0.379	0.831±0.030	0.126	0.597
w/o P-weighted density	0.870±0.026	0.098	0.380	0.836±0.030	0.112	0.585
w/o risk-aware placement	0.875±0.027	0.094	0.375	0.838±0.030	0.112	0.585
RED-V (no reliability model)	0.860±0.027	0.112	0.390	0.821±0.030	0.139	0.608

6.4 Sensitivity Analysis

Figure 6 varies the main scenario parameters. Platform size (Figure 6(a)): all policies improve as nodes are added; TRIAD remains on top across the range, and its margin over EDF is largest on small platforms, where overload is most severe (0.783 versus 0.102 with six nodes). Failure intensity (Figure 6(b)): as MTBFs shrink from $4\times$ to $0.25\times$ the base values, TRIAD degrades gracefully from 0.881 to 0.834, and Figure 6(e) shows that its commitment-violation rate stays below that of RED-V across the range. Deadline tightness (Figure 6(c)): with deadlines scaled by 0.75, TRIAD obtains 0.825 versus 0.809 for RED-V and 0.152 for EDF; with doubled deadlines the overload largely disappears and the gap narrows (0.970 for TRIAD and RED-V, 0.890 for EDF). Burst intensity (Figure 6(d)): scaling the aftershock productivity up to $3\times$ adds re-assessment bursts; TRIAD keeps the highest WDSR (0.853 at $3\times$). Admission threshold (Figure 6(f)): WDSR varies only between 0.861 and 0.870 for $\theta \in [0.3, 0.7]$, while a larger θ commits more conservatively and lowers CVR, so θ acts as a dial for the reliability of commitments that does not hurt the primary objective.

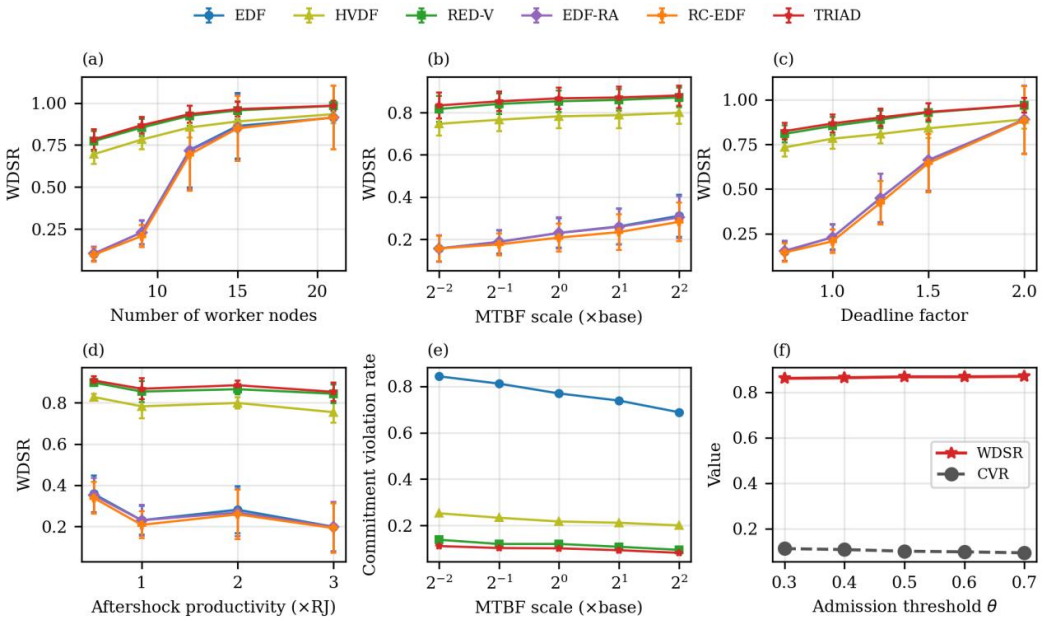


Figure 6 Sensitivity on San Mateo (10 Seeds, 95% CIs): (a) Number of Nodes, (b) Failure Intensity, (c) Deadline Factor, (d) Aftershock Productivity, (e) Commitment-Violation Rate versus Failure Intensity, (f) TRIAD Admission Threshold θ

6.5 Robustness

Table 5 perturbs assumptions that TRIAD's model relies on. TRIAD obtains the highest WDSR in all seven scenarios. With exact runtime estimates and with runtime errors of CV 0.4 and 0.6, TRIAD's model keeps its fixed $c_v = 0.2$ and still leads. Exponential failures ($k_W = 1$), which match the Poisson assumption behind (6)–(8), and burstier failures ($k_W = 0.5$), which depart from it, leave the ranking unchanged because the discounted estimator tracks the observed failure rate in both cases. Removing correlated outages or making them certain (every nearby $M \geq 5$ aftershock takes the EDGE pool down) changes absolute values but not the ranking: pooled hazard estimation quickly reflects pool-wide outages in the effective capacity of (7).

Table 5 Robustness: WDSR under Perturbed Scenarios (San Mateo, 10 Seeds; Best in Bold)

Scenario	EDF	HVDF	RED-V	EDF-RA	RC-EDF	TRIAD
Exact runtime estimates (CV=0)	0.247	0.783	0.863	0.238	0.236	**0.871**
Runtime error CV=0.4	0.234	0.780	0.847	0.223	0.214	**0.861**
Runtime error CV=0.6	0.234	0.780	0.841	0.233	0.218	**0.854**
Exponential failures (k=1.0)	0.248	0.782	0.856	0.250	0.233	**0.867**
Bursty failures (k=0.5)	0.187	0.770	0.849	0.182	0.166	**0.862**
No correlated outages	0.273	0.798	0.869	0.280	0.271	**0.881**
Strong correlated outages	0.187	0.753	0.823	0.184	0.170	**0.836**

6.6 Scheduling Overhead

TRIAD's decisions take 0.167 ms on average per planning or dispatch call (the largest per-run average is 0.367 ms) in a single-threaded Python implementation, and a full 48-hour simulation of the San Mateo workload completes in 0.76 s.

Since assessment tasks last from seconds to tens of minutes, the overhead is negligible and consistent with Proposition 3.

7 DISCUSSION

Operational use. TRIAD fits naturally into workflow systems that already separate planning from execution, such as Pegasus [23] and the SimCenter regional workflow [4]. Its inputs are available in an emergency: job deadlines follow from the decision timelines of the operations center, weights from the Hazus estimates that agencies routinely compute [3], nominal reliabilities from provider statistics, and observed failures from the workflow system's monitoring. Its outputs are a committed set with planned finish times and a best-effort queue; the CVR results show that the planned times are trustworthy enough to be communicated.

Interpretation of the gains. The experiments separate two effects. Admission control is essential under post-disaster overload: without it, deadline-driven policies serve jobs that are already lost. Reliability-awareness determines *how much* to commit: inflated work and online hazard estimates align the triage with the capacity that a failure-prone platform actually delivers, which raises the weighted on-time ratio and reduces broken commitments, especially when nominal reliabilities are optimistic. Static redundancy, in contrast, is counterproductive when capacity is the bottleneck.

Extensions. The model accommodates further information without structural change. Job-specific deadline and value curves can replace tier deadlines; checkpointed tasks can use the corresponding expected-time expressions [24]; and elastic provisioning can be coupled to the triage by requesting nodes whenever valuable jobs are about to be moved to the best-effort set. Applying the same scheduler to hurricane wind and storm-surge assessment workflows, whose damage models are also available in Hazus form, is a direct next step.

8 CONCLUSION

This paper addressed the scheduling of post-disaster structural assessment workflows on a computing service that is overloaded by bursts of requests and weakened by node failures. TRIAD couples a discounted Gamma–Poisson hazard estimator, failure-inflated work derived from the expected execution time under fail-restart, a Poisson–Gaussian deadline-meeting probability, a probability-weighted Moore–Hodgson triage and a risk-aware dispatcher. Evaluated on real building inventories from San Mateo and Alameda, a USGS scenario ShakeMap, Hazus v6.1 fragility and loss tables, and measured OpenSees analysis costs, TRIAD delivered the highest weighted deadline satisfaction on the San Mateo workload (0.875 versus 0.860 for the strongest baseline and 0.260 for EDF), the highest tier-1 satisfaction, and the most reliable commitments, with sub-millisecond decisions. The results show that, after an earthquake, the computing service should decide jointly which assessments it can still deliver on time and where to run them, using an explicit and continuously updated model of the failure environment.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] Wald D J, Quitoriano V, Heaton T H, et al. TriNet 'ShakeMaps': Rapid generation of peak ground motion and intensity maps for earthquakes in southern California. Earthquake Spectra, 1999, 15(3): 537-555. DOI: 10.1193/1.1586057.

[2] Kircher C A, Whitman R V, Holmes W T. HAZUS earthquake loss estimation methods. Natural Hazards Review, 2006, 7(2): 45-59. DOI: 10.1061/(ASCE)1527-6988(2006)7:2(45).

[3] Federal Emergency Management Agency. Hazus earthquake model technical manual, Hazus 6.1. Washington, DC: FEMA, 2024.

[4] Deierlein G G, Zsarnóczay A, Singh K, et al. A cloud-enabled application framework for simulating regional-scale impacts of natural hazards on the built environment. Frontiers in Built Environment, 2020, 6: 558706. DOI: 10.3389/fbuil.2020.558706.

[5] Lu X, McKenna F, Cheng Q, et al. An open-source framework for regional earthquake loss estimation using the city-scale nonlinear time history analysis. Earthquake Spectra, 2020, 36(2): 806-831.

[6] Reasenberg P A, Jones L M. Earthquake hazard after a mainshock in California. Science, 1989, 243(4895): 1173-1176. DOI: 10.1126/science.243.4895.1173.

[7] Beckman P, Nadella S, Trebon N, et al. SPRUCE: A system for supporting urgent high-performance computing. In: Grid-Based Problem Solving Environments. IFIP, vol. 239. Boston, MA: Springer, 2007: 295-311.

[8] Schroeder B, Gibson G A. A large-scale study of failures in high-performance computing systems. IEEE Transactions on Dependable and Secure Computing, 2010, 7(4): 337-350.

[9] Liu C L, Layland J W. Scheduling algorithms for multiprogramming in a hard-real-time environment. Journal of the ACM, 1973, 20(1): 46-61. DOI: 10.1145/321738.321743.

[10] Dertouzos M L. Control robotics: The procedural control of physical processes. Proceedings of the IFIP Congress, 1974: 807-813.

[11] Jensen E D, Locke C D, Tokuda H. A time-driven scheduling model for real-time operating systems. Proceedings of the 6th IEEE Real-Time Systems Symposium (RTSS), 1985: 112-122.
[12] Buttazzo G, Spuri M, Sensini F. Value vs. Deadline scheduling in overload conditions. Proceedings of the 16th IEEE Real-Time Systems Symposium (RTSS), 1995: 90-99.
[13] Koren G, Shasha D. An optimal on-line scheduling algorithm for overloaded uniprocessor real-time systems. SIAM Journal on Computing, 1995, 24(2): 318-339. DOI: 10.1137/S0097539792236882.
[14] Zhu X, Wang J, Guo H, et al. Fault-tolerant scheduling for real-time scientific workflows with elastic resource provisioning in virtualized clouds. IEEE Transactions on Parallel and Distributed Systems, 2016, 27(12): 3501-3517. DOI: 10.1109/TPDS.2016.2543731.
[15] Xie G, Zeng G, Li R, et al. Minimizing redundancy to satisfy reliability requirement for a parallel application on heterogeneous service-oriented systems. IEEE Transactions on Services Computing, 2020, 13(5). DOI: 10.1109/TSC.2017.2665552.
[16] Dongarra J J, Jeannot E, Saule E, et al. Bi-objective scheduling algorithms for optimizing makespan and reliability on heterogeneous systems. Proceedings of the 19th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), 2007: 280-288. DOI: 10.1145/1248377.1248423.
[17] McKenna F. OpenSees: A framework for earthquake engineering simulation. Computing in Science & Engineering, 2011, 13(4): 58-66. DOI: 10.1109/MCSE.2011.66.
[18] Trebon N, Beckman P H. Empirical-based probabilistic upper bounds for urgent computing applications. Proceedings of the IEEE International Conference on Cluster Computing (CLUSTER), 2008: 342-347.
[19] Dertouzos M L, Mok A K. Multiprocessor on-line scheduling of hard-real-time tasks. IEEE Transactions on Software Engineering, 1989, 15(12): 1497-1506.
[20] Moore J M. An n job, one machine sequencing algorithm for minimizing the number of late jobs. Management Science, 1968, 15(1): 102-109. DOI: 10.1287/mnsc.15.1.102.
[21] Topcuoglu H, Hariri S, Wu M-Y. Performance-effective and low-complexity task scheduling for heterogeneous computing. IEEE Transactions on Parallel and Distributed Systems, 2002, 13(3): 260-274.
[22] Rodriguez M A, Buyya R. Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. IEEE Transactions on Cloud Computing, 2014, 2(2): 222-235. DOI: 10.1109/TCC.2014.2314655.
[23] Deelman E, Vahi K, Juve G, et al. Pegasus, a workflow management system for science automation. Future Generation Computer Systems, 2015, 46: 17-35. DOI: 10.1016/j.future.2014.10.008.
[24] Bougeret M, Casanova H, Rabie M, et al. Checkpointing strategies for parallel jobs. Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC), 2011: 1-11.
[25] NHERI SimCenter. R2D user manual: Regional earthquake assessment examples (including ShakeMap scenarios and Alameda inventory). R2D v5.5 documentation, 2025. https://nheri-simcenter.github.io/R2D-Documentation/common/user_manual/examples/desktop/examples.html.
[26] NHERI SimCenter. Damage and Loss Model Library (DLML), version 2.1.0. Software/data release, 2025. https://github.com/NHERI-SimCenter/DamageAndLossModelLibrary/releases.
[27] American Society of Civil Engineers. Minimum design loads and associated criteria for buildings and other structures (ASCE/SEI 7-16). Reston, VA: ASCE, 2017.