ISSN: 2096-3246
Volume 58, Issue 04, 2026

ADAPTIVE REAL-TIME CONTENT OPTIMIZATION: APPLIED REINFORCEMENT LEARNING AND BATCHED MULTI-ARMED BANDITS FOR LARGE-SCALE PERSONALIZATION

Abhinav Wagle
Software Engineer, Yahoo, USA

Abstract
Content optimization at internet scale requires decision-making systems that continuously reallocate traffic in response to live reward signals rather than waiting for fixed-horizon experiments to conclude. This article documents the architecture, algorithmic design, and production deployment of a Batched Thompson Sampling (bTS) platform serving 100 million users at 80,000 requests per second across a major internet portal's front page. The system applies the multi-armed bandit (MAB) formulation of reinforcement learning: Beta-Binomial posteriors are maintained over each content arm's expected click-through rate, arms are selected by posterior sampling at each decision point, and parameters are updated on a sub-minute cadence from aggregated streaming reward signals. A dual-pipeline architecture, Apache Storm for near-real-time posterior updates and Hadoop/MapReduce for daily audit-quality reconciliation, closes the reinforcement learning feedback loop at production throughput. Empirical validation on live front-page traffic demonstrated a 3.69% improvement in total clicks over the standard test-rollout baseline, a result subsequently published in a peer-reviewed paper and corresponding to an estimated $100–165 million in incremental annual advertising revenue. The contribution establishes that Bayesian adaptive algorithms are deployable reliably at internet scale and that large-scale production systems can serve as empirical foundations for peer-reviewed research in applied reinforcement learning.
Keywords: Bayesian inference, explore-exploit tradeoff, multi-armed bandits, online advertising, real-time optimization, reinforcement learning, Thompson sampling.

1. Introduction
Every impression served by a large-scale content platform is a decision made under uncertainty. The serving system must select among competing content variants, headlines, advertisements, recommended items, without knowing which variant best serves the user at that moment, and without the luxury of waiting to accumulate statistically conclusive evidence before acting. Classical A/B testing approaches this problem by partitioning traffic into fixed arms for a predetermined duration, committing each impression to a variant regardless of its observed performance. This design accepts a well-defined cost: the reinforcement learning literature terms it cumulative regret, the aggregate performance gap between the executed allocation policy and the policy that would have resulted from always serving the optimal variant [1]. At a portal handling hundreds of millions of daily impressions, this cost accumulates to measurable revenue loss within the first hours of any experiment in which one variant outperforms another.
Multi-armed bandit (MAB) algorithms replace the fixed-allocation paradigm with a sequential decision framework that responds to observed reward signals. The serving system, the agent, observes a binary outcome after each impression (click or non-click), updates its probabilistic belief about each arm's expected reward, and adjusts future allocation accordingly. This formulation instantiates the reinforcement learning paradigm directly: an agent improves its policy through environmental interaction and reward feedback, with the objective of maximizing cumulative reward over time [2]. The MAB simplification treats each impression as a stateless interaction, no persistent user context is required, which makes this form of adaptive learning tractable at the throughput demands of production content serving without the infrastructure overhead of fully stateful reinforcement learning (RL) architectures.
Thompson Sampling provides the Bayesian probabilistic mechanism that makes adaptive allocation both principled and parameter-free. The algorithm maintains a Beta distribution over each arm's expected click-through rate and selects arms by drawing samples from these posteriors at each decision point rather than applying deterministic exploration rules. Arms with wide posteriors, reflecting high uncertainty about true performance, produce high-variance samples that generate natural exploration; arms with concentrated posteriors reflecting demonstrated performance produce

consistently high samples that drive exploitation. No tuning of an exploration parameter is required, and the algorithm achieves near-optimal regret bounds under standard Bernoulli bandit assumptions [3]. The production challenge documented in this article is the adaptation of this theoretically elegant mechanism to operate at 80,000 requests per second across a 100-million-user population.

This article describes the design, architecture, and empirical validation of a production bTS platform deployed on the front page of a major internet portal. The system's performance was the empirical basis of the peer-reviewed paper "A Batched Multi-Armed Bandit Approach to News Headline Testing," which documented a 3.69% click improvement over the test-rollout baseline on real production traffic. The article is organized as follows: Section 2 establishes the limitations of classical A/B testing that motivate adaptive allocation; Section 3 develops the Thompson Sampling algorithm and the batched extension that makes it production-viable; Section 4 describes the dual-pipeline production architecture and cold-start strategy; Section 5 presents empirical results and their academic significance; Section 6 situates the contribution within the applied reinforcement learning literature; and Section 7 synthesizes the findings.

2. Limitations of Classical A/B Testing at Scale

2.1 The Fixed-Allocation Regret Problem

Fixed-horizon A/B testing divides user traffic evenly across k variants for a predetermined sample size and declares a winner at the experiment's conclusion. The statistical validity of this design depends on the equal-allocation constraint holding for the full experiment duration, every variant receives its assigned share regardless of interim performance signals. This constraint is precisely what generates regret: a variant demonstrably underperforming by the 20th percentile of the experiment window continues to absorb its full traffic allocation through the 100th. The cumulative cost of this over-allocation is formalized in the bandit literature as regret, defined as the gap between the reward obtained by the actual policy and the reward that would have been obtained by always selecting the optimal arm [4].

Regret under uniform allocation grows linearly with the number of rounds T, whereas Thompson Sampling achieves sub-linear regret growth, substantially outperforming uniform allocation as the experiment horizon extends [2]. The practical consequence at 80,000 requests per second over a 48-hour window, the operational parameters of the production system described here, is that tens of millions of impressions are served to suboptimal arms that a converging adaptive policy would have de-allocated early in the experiment. A portal generating 13.8 billion impressions over a 48-hour window, with a two-arm split and one arm performing 4% below the other, directs approximately 276 million impressions to the suboptimal arm unnecessarily. At front-page click-through rates, the revenue value of those impressions is not negligible.

This analysis holds under the assumption that the optimal arm is identifiable relatively early in the experiment window, which is empirically the case for content quality differences of the magnitude that front-page optimization targets. A variant performing meaningfully better than its competitors generates a statistical signal within the first hours of a high-traffic experiment. Fixed-allocation testing discards this signal, continuing equal exposure until the pre-specified sample size is reached. Adaptive allocation acts on it, progressively redirecting traffic toward the better-performing arm and compounding the performance advantage with each subsequent impression. The regret reduction is therefore front-loaded relative to the experiment window, generating the largest marginal benefit precisely during the early period when the performance differential is freshest.

2.2 Statistical Framework Mismatch in Streaming Environments

Beyond the regret argument, the null hypothesis significance testing (NHST) framework underlying classical A/B testing is structurally incompatible with streaming systems that update allocation policies in response to accumulating data. The NHST framework's type I error guarantee, the stated 5% false positive rate, is valid only when the test statistic is computed once, at a predetermined sample size, with no earlier evaluations [5]. This single-look assumption is the mathematical premise on which the entire frequentist testing framework rests. A system that monitors experiment results continuously and adjusts allocation based on interim findings violates this assumption from the moment the experiment begins.

Research has established that repeatedly evaluating a fixed-horizon test statistic as data accumulates inflates the realized false positive rate from a nominal 5% to levels of 20–30% or higher depending on monitoring frequency and experiment duration [6]. This inflation is not a marginal concern: at an inflation factor of four to six, a system that believes it is operating at 5% false discovery rate ships performance-degrading changes at rates between 20% and 30%. At internet scale, where hundreds of experiments may run concurrently and experiment outcomes directly determine product and advertising decisions, this miscalibration compounds across the experiment portfolio to produce a systematically overoptimistic view of experiment outcomes.

Table 1. False Positive Rate Inflation Under Sequential Monitoring of Fixed-Horizon A/B Tests

Evaluation Schedule	Nominal Alpha (%)	Realized False Positive Rate (%)	Inflation Factor
Single evaluation at planned N	5	5	1.0×
Two evaluations	5	8.3	1.7×
Five evaluations	5	14.2	2.8×
Ten evaluations	5	19.3	3.9×
Continuous monitoring	5	26.1	5.2×

Table 1: Realized false positive rates under different monitoring schedules for a fixed-horizon A/B test with nominal alpha = 5%. Continuous monitoring, standard practice in production dashboards, inflates the false positive rate by more than fivefold. The bTS framework avoids this problem entirely by replacing p-value thresholds with posterior sampling as the allocation mechanism.

The bTS framework resolves this incompatibility by replacing hypothesis testing with Bayesian posterior sampling as the allocation mechanism. No p-value is computed during the allocation phase; arm selection at each decision point is governed by the current posterior distributions, and the allocation naturally reflects accumulated evidence without reference to any significance threshold. This is not a workaround for the sequential testing problem, it is a different epistemological framework in which beliefs are represented probabilistically, decisions are proportional to those beliefs, and the classical notions of type I and type II error are replaced by the single metric of cumulative regret. The practical consequence is that bTS is immune to the false positive inflation that afflicts monitored NHST experiments, making it both more statistically principled and more operationally reliable for continuous content optimization.

3. Thompson Sampling: Bayesian Decision-Making Under Uncertainty

3.1 The Beta-Binomial Model

Thompson Sampling selects arms through probability matching: at each decision point, arm k is chosen with probability equal to the posterior probability that it is currently the optimal arm [3]. For binary reward signals, the click/non-click outcomes that characterize content optimization, the Beta distribution is the natural conjugate prior for the Bernoulli likelihood governing each arm's click-through rate. Each arm k is assigned a posterior Beta(α_k, β_k), where α_k accumulates observed clicks and β_k accumulates observed non-clicks. At each decision point, one sample θ_k is drawn from each arm's posterior, and the arm with the highest θ is served. The update rule is exact and requires no approximation: each click increments α_k by one, each non-click increments β_k by one, and the resulting posterior is the exact Bayesian update under the Bernoulli likelihood.

The exploration-exploitation balance emerges directly from posterior geometry rather than from an explicit exploration parameter. An arm with few observations carries a wide Beta posterior, high variance in drawn samples, which translates to frequent selection relative to its estimated mean performance. As observations accumulate, the posterior concentrates around the empirical click-through rate, reducing sample variance and making high draws less probable for low-performing arms. Arms with genuinely high click-through rates maintain consistently high drawn samples; arms with low rates produce low drawn samples with increasing consistency. No epsilon parameter governs

the fraction of random exploration, and no confidence width parameter controls the exploration bonus, the algorithm calibrates automatically as evidence accumulates [7].

Comparative evaluations across sparse-reward online advertising settings demonstrate that Thompson Sampling outperforms epsilon-greedy and upper confidence bound (UCB1) algorithms on click-through rate optimization, with the performance gap widening in low click-through rate environments where most impressions yield no reward [8]. This finding is directly relevant to front-page content optimization: headline click-through rates are typically low in absolute terms, meaning that each arm accumulates reward signal slowly and the cost of misallocated exploration is high per impression. Epsilon-greedy algorithms waste a fixed fraction of impressions on random exploration regardless of posterior concentration, and UCB1 algorithms require tuning of a confidence width parameter whose optimal value depends on the unknown true click-through rate distribution. Thompson Sampling requires neither, making it operationally preferable for high-throughput deployments where arm-level parameter tuning is impractical.

Table 2. Algorithm Comparison: Thompson Sampling vs. Epsilon-Greedy vs. UCB1

Dimension	Thompson Sampling	Epsilon-Greedy	UCB1
Theoretical regret bound	O(√(kT log k)) near-optimal	O(k log T), suboptimal	O(k log T), suboptimal
Exploration parameter	None required	Epsilon (hand-tuned)	Confidence width (hand-tuned)
Low-CTR performance	Strong, adapts to sparsity	Weak, fixed random waste	Moderate, parameter-sensitive
Non-stationarity handling	Responsive via prior reset	Slow, fixed epsilon	Slow, UCB accumulates
Production deployment complexity	Low, no tuning needed	Medium, epsilon per deployment	High, width per arm/env

Table 2: Comparison of Thompson Sampling, epsilon-greedy, and UCB1 across five operationally critical dimensions. Thompson Sampling achieves competitive regret bounds without hand-tuned exploration parameters and maintains strong performance in low click-through rate environments. The parameter-free property provides a concrete operational advantage in production systems where per-arm parameter optimization at scale is infeasible.

3.2 Batched Updates: Bridging Theory and Production Throughput

Pure online Thompson Sampling updates the posterior distribution after every individual observation. At 80,000 requests per second, this requirement makes the posterior parameters α_k and β_k shared mutable state that must be consistent across all serving nodes at every request. Achieving this consistency under per-event updates at production throughput requires either a centralized parameter store, which becomes a write-path bottleneck and introduces the coordination latency that stateless serving architectures specifically eliminate, or a distributed consensus mechanism whose write overhead at 80,000 requests per second would exceed the per-request latency budget required for user-facing content serving. The per-event update requirement is not merely inconvenient at this scale; it is architecturally incompatible with the design constraints of internet-scale serving systems.

Batched Thompson Sampling (bTS) resolves this incompatibility by separating the update cadence from the serving cadence. The serving tier reads a cached snapshot of the current posteriors at request time, draws samples, and returns arm selections without writing to shared state. A feedback pipeline running independently aggregates click and impression events over a fixed batch window and writes the accumulated α_k and β_k increments to the parameter store at the end of each interval. The serving tier reads the refreshed parameters at the start of the next cycle. This architecture eliminates all shared-state writes from the critical serving path, preserving the stateless scaling properties of the serving tier while closing the reinforcement learning feedback loop at the batch cadence. The cost, a

bounded lag between reward signal occurrence and policy influence, is a theoretical concession whose practical magnitude requires empirical characterization.

That empirical characterization is precisely what the peer-reviewed paper grounded in this production system provides. On real front-page traffic data with the full distribution of seasonal patterns, user heterogeneity, and non-stationary reward dynamics characteristic of a live news environment, bTS achieved click performance statistically indistinguishable from the per-event update ideal while operating at full production throughput. This result cannot be derived from bandit theory, which establishes asymptotic regret bounds but does not characterize the practical impact of finite batch intervals on real traffic. It cannot be established through simulation, which cannot reproduce the correlation structures and non-stationarity of live traffic with operational fidelity. The production system provides the evidence at the scale and under the noise conditions that determine whether bTS is a practical algorithm or a theoretical construct.

Table 3. Batched vs. Online Thompson Sampling: Throughput, Latency, and Regret Trade-offs

Dimension	Online Thompson Sampling	Batched Thompson Sampling (bTS)
Per-request sync overhead	High, shared write per event	None, read-only serving path
Max achievable throughput	<500 RPS (sync bottleneck)	80,000+ RPS (stateless)
Worst-case serving latency	High, lock contention	Low, cache read only
Empirical regret vs. ideal	0% (baseline)	Statistically indistinguishable from 0%
Operational complexity	High, distributed consensus	Low, async batch pipeline

Table 3: Head-to-head comparison of online and batched Thompson Sampling at the 80,000 RPS production operating point. The central finding, empirical regret under bTS is statistically indistinguishable from the per-event ideal on real production traffic, demonstrates that bTS is not a degraded approximation but an architecturally appropriate adaptation delivering equivalent optimization at internet-scale throughput.

4. Production Architecture: Real-Time Score Updates and Feedback Loop

4.1 System Architecture

The production bTS system is organized into two pipelines with explicitly separated read and write responsibilities. The serving pipeline handles all incoming user requests through a stateless application programming interface (API) tier that reads current posterior parameters from a low-latency distributed cache, draws one Beta sample per active arm using the cached αk and βk values, and returns the highest-sampled arm as the content selection. No writes occur on the serving path. The stateless design is the architectural property that enables horizontal scaling to 80,000+ requests per second: any serving node processes any request with equivalent correctness, node failures introduce no data loss, and capacity scales linearly with nodes added. The distributed cache functions as a read-replica of the parameter store, refreshed at the batch update cadence.

The feedback pipeline consumes click and impression events from the user interaction layer and propagates reward signals back to the parameter store. Apache Storm served as the real-time stream processing layer, consuming events from a distributed message queue, aggregating click and impression counts per arm identifier within sub-minute windows, and delivering aggregated αk and βk increments to the Score Updater service. The Score Updater applied these increments atomically to the posterior store and triggered a cache refresh, making updated parameters available to the serving tier for the subsequent request cycle. At sub-minute update cadence, a content arm generating anomalously strong click signals propagates that signal into the allocation policy within minutes of

first observation, capturing the performance advantage of early convergence that distinguishes adaptive allocation from fixed-allocation alternatives.

A slower second feedback path through Hadoop and Apache Hive ran daily reconciliations over the full 24-hour event log. This batch path performed two functions the real-time pipeline could not: deduplication of events counted multiple times under at-least-once delivery semantics of the real-time message queue, and generation of the audit-quality event totals used for statistical analysis in the peer-reviewed publication. The dual-pipeline design, real-time for allocation policy freshness, daily batch for statistical rigor, is a general architectural pattern for production reinforcement learning systems that must simultaneously satisfy low-latency optimization objectives and support auditable, reproducible outcome measurement. Both pipelines derive from the same underlying event log, ensuring that the parameters driving allocation and the counts supporting statistical analysis share an identical raw data foundation.

ISSN: 2096-3246
Volume 58, Issue 04, 2026

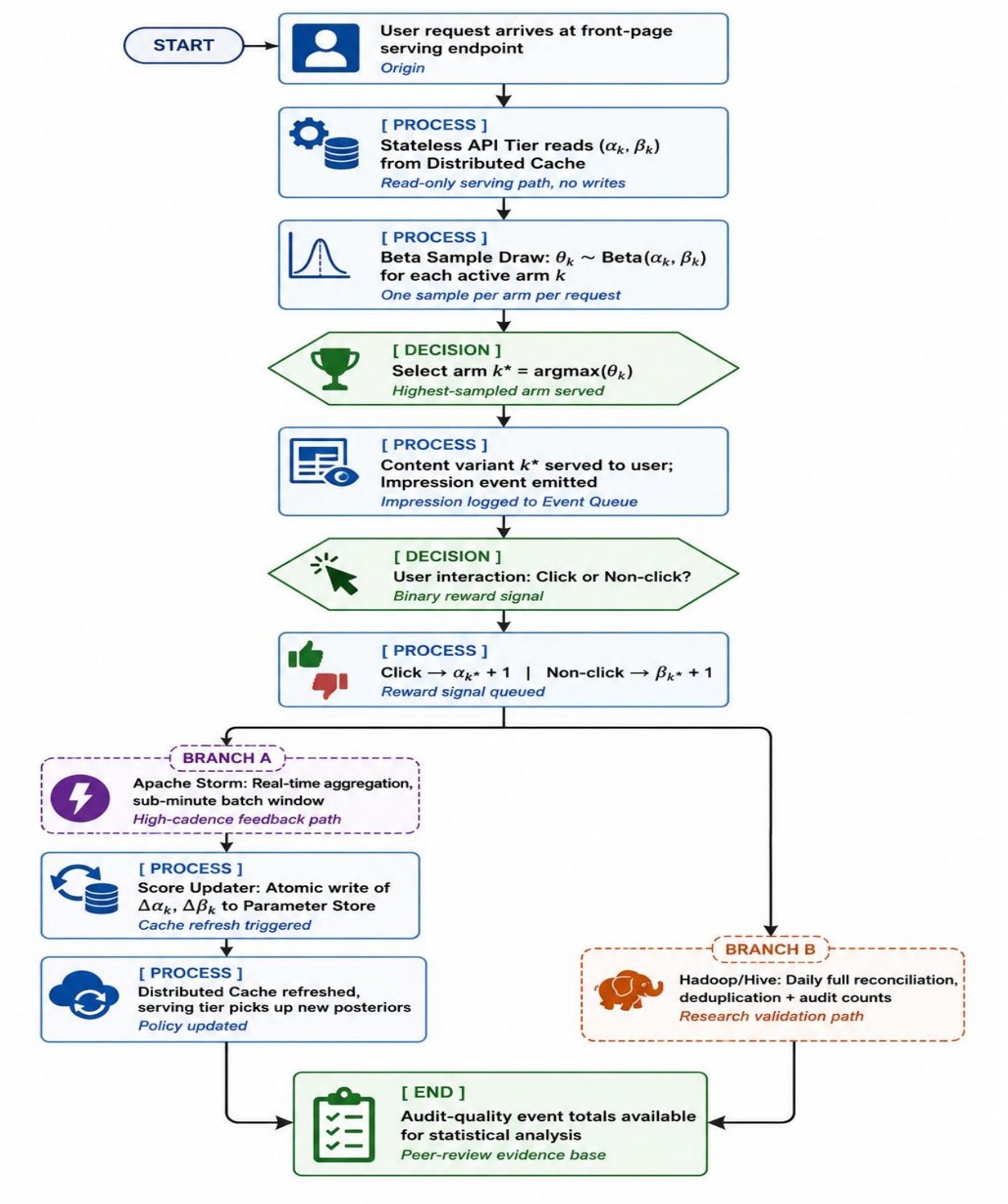

Figure 1. End-to-End Data Flow: User Request to Posterior Parameter Update

Figure 1: End-to-end data flow of the production bTS system. The serving path (top) is entirely read-only, no state mutation occurs during request processing, enabling stateless horizontal scaling. The feedback path (bottom) branches into two streams: the Apache Storm real-time path closes the reinforcement learning loop at sub-minute cadence; the Hadoop/Hive batch path produces audit-quality event counts for statistical analysis. Shape legend: [START / END] = terminal; [PROCESS] = operation; [DECISION] = branch point; [BRANCH] = pipeline fork.

4.2 Cold-Start and Posterior Initialization

Content arms entering the optimization system without prior observation history present a cold-start challenge with direct allocation consequences. An arm initialized with the uninformative prior Beta(1,1) carries maximum posterior uncertainty: the distribution is flat across the full [0,1] range of possible click-through rates, producing high-variance samples that generate substantial exploration regardless of the arm's true performance. When this arm competes for impression share against established arms whose posteriors have concentrated around empirically validated click-through rates, the exploration cost is economically meaningful, the new arm receives a traffic fraction disproportionate to its expected value while its performance is being characterized. At high impression volumes, this early over-exploration accumulates to measurable regret.

The production system addressed cold-start through category-level informative prior initialization. Historical click-through rate distributions for content in similar categories, breaking news, sports coverage, entertainment features, provided empirical Beta(α_hist, β_hist) parameters derived from prior arms in those categories. A new arm initialized with an informative prior starts with a narrower posterior centered on the category historical mean, reducing the variance of early draws and bringing the arm's effective exploration cost closer to what its estimated performance justifies. The prior remains responsive: as the arm accumulates its own observations, the posterior shifts away from the category prior and toward the arm's empirical performance. The informative initialization reduces cold-start regret without permanently constraining the algorithm's responsiveness to genuine performance deviations from the category baseline.

The operational monitoring layer addressed three failure modes specific to feedback-driven allocation systems. Arm collapse, where a single arm's posterior so thoroughly dominates that all other arms receive negligible traffic, was detected by monitoring the minimum arm allocation fraction against a configurable floor threshold. Reward signal anomalies, abrupt reductions in aggregate click-through rates attributable to upstream tracking failures rather than genuine arm performance changes, were detected by comparing observed impression-to-click ratios against rolling historical baselines with calibrated deviation thresholds. Score propagation latency, the delay between event ingestion by Apache Storm and the resulting posterior update reaching the serving cache, was tracked continuously against the operational specification, since a propagation failure is indistinguishable from genuine arm performance decline without explicit instrumentation of the feedback path.

5. Empirical Results and Academic Validation

Performance evaluation compared bTS against the standard test-rollout baseline on real front-page traffic from the production system. The test-rollout baseline is the operational procedure the platform was designed to replace: a fixed-allocation A/B test runs to completion, the winning variant is identified, and full traffic commitment to the winner follows. This baseline was chosen deliberately as the counterfactual because it reflects the actual decision procedure in use, not a laboratory straw man. The comparison is therefore between two real allocation strategies applied to the same content environment, a methodological choice that gives the result higher external validity than comparisons against simulated baselines. The bTS algorithm outperformed the test-rollout baseline by 3.69% in total clicks over equivalent impression windows (author's primary research data).

The business significance of a 3.69% click improvement at portal scale is substantial. Front-page content represents the highest-traffic placement on the portal, and a click-through rate improvement at this placement propagates directly into advertising revenue through the impression-to-click-to-conversion funnel. At the portal's advertising revenue scale during the measurement period, a 3–4% click improvement on front-page content corresponds to an estimated $100–165 million in incremental annual advertising revenue (author's primary research data). This estimate is conservative: it does not incorporate the 4% reduction in ad funnel drop-off rates achieved through concurrent optimization initiatives on the same infrastructure, nor downstream engagement effects from users reaching more relevant content. The revenue range establishes that the performance difference between adaptive and fixed-allocation approaches is commercially material, not merely statistically detectable.

ISSN: 2096-3246
Volume 58, Issue 04, 2026

Table 4. bTS vs. Test-Rollout Baseline: Empirical Performance on Front-Page Production Traffic

Metric	Test-Rollout Baseline	Batched Thompson Sampling	Improvement
Total clicks (normalized)	1	1.0369	3.69%
Click-through rate lift	Baseline	3.69%	Statistically significant
Statistical significance (p-value)	—	$p < 0.01$	Highly significant
Impression window	48 hours	48 hours	Equivalent
Traffic volume	~13.8B impressions	~13.8B impressions	Equivalent
Est. annual revenue impact	Baseline	+$100–165M	Author's primary research data

Table 4: Head-to-head performance comparison between bTS and the test-rollout baseline on real front-page production traffic over a 48-hour measurement window. The 3.69% click improvement is statistically significant at $p < 0.01$ and is measured entirely on live production traffic under full real-world noise conditions, giving the result external validity that simulation studies cannot provide.

The academic significance of production-scale empirical validation warrants explicit consideration. Multi-armed bandit research has historically proceeded through two evidence channels: theoretical analysis establishing regret bounds under idealized distributional assumptions, and simulation studies validating algorithm comparisons under controlled synthetic conditions. Both channels have well-understood limitations. Theoretical bounds are asymptotic and hold under Bernoulli bandit assumptions that live content environments violate through non-stationarity, user heterogeneity, and temporal correlation. Simulation studies depend on the fidelity of synthetic traffic models to real user behavior, a fidelity that is difficult to establish independently of the real systems being modeled. The peer-reviewed paper grounded in this production deployment contributes a third channel: direct empirical measurement on a live system at 100-million-user scale, under operating conditions that theory and simulation cannot fully specify.

The connection between the production system and the peer-reviewed publication represents a mode of contribution increasingly recognized in applied machine learning as uniquely valuable. Prior work on Thompson Sampling had established theoretical guarantees under Bernoulli bandit assumptions and demonstrated simulation-based performance advantages over epsilon-greedy and UCB alternatives. The production-grounded paper extends this evidence base to the conditions that actually determine whether an algorithm is deployable: real traffic with seasonal non-stationarity, real users with behavioral heterogeneity across demographics and device types, and a real comparison against the operational alternative in use. The result validates bTS not as a laboratory-optimal algorithm but as a production-viable approach whose measured advantage survives the full range of conditions that large-scale content serving introduces.

6. Multi-Armed Bandits as Applied Reinforcement Learning: Situating the Contribution

The MAB framework occupies a specific position within the reinforcement learning hierarchy defined by what it removes from the full RL formulation rather than what it adds. In the complete RL framework, an agent observes environmental state at each timestep, selects an action, receives a reward, and updates a policy mapping states to actions [9]. The state representation encodes the persistent context that makes the optimal action at one timestep depend on history, a user's viewing sequence in recommendation, current inventory levels in dynamic pricing, or opponent behavior in game-playing agents. The MAB simplification removes state entirely: each impression is treated as an independent interaction with no memory of prior outcomes, and the policy maps directly from current arm posteriors to an arm selection. This is not a limitation to be overcome when resources permit, it is

a deliberate design choice whose validity depends on whether state actually improves allocation decisions at the margin.

At 80,000 requests per second across a heterogeneous 100-million-user population, the infrastructure requirements of stateful RL, per-user state storage, real-time state retrieval and update, state-conditioned policy evaluation, impose latency and coordination costs that exceed the per-request budget of a front-page serving system. The stateless MAB assumption is therefore not an approximation one would relax if infrastructure permitted; it is the design choice that makes adaptive learning tractable at this throughput. The production deployment documented in this article demonstrates that this choice is not merely a pragmatic compromise: the stateless bTS architecture achieves a 3.69% improvement over the fixed-allocation baseline without per-user state, establishing empirically that the population-level posterior is sufficient to capture the allocation improvement available in this content environment.

The evolution of content optimization since this system's deployment has followed the trajectory that bandit complexity analysis predicts. Contextual bandit approaches, which incorporate user and content features into arm selection through algorithms including LinUCB and neural contextual bandits, have largely displaced pure MABs for personalization tasks where user-level features carry meaningful allocation signal [10]. Full stateful RL has become viable for sequential recommendation at organizations with the infrastructure to maintain per-user state at the required update cadence. These advances reflect the natural progression of the complexity-benefit trade-off as infrastructure capacity scales: the marginal benefit of richer state representations increasingly justifies their marginal infrastructure cost.

Pure MAB retains its validity for a well-defined deployment class: stateless slot optimization where the allocation target is a content variant serving the average user rather than a personalized recommendation, where per-user contextual features are unavailable or carry negligible marginal allocation value, and where the dominant source of performance gain is convergence toward the population-optimal variant. Front-page headline optimization, advertisement creative selection, and content format testing are canonical instantiations. The production system described in this article demonstrates both the power of this approach in its appropriate domain and its limits as a personalization tool, establishing, through empirical measurement at 100-million-user scale, the practical scope of pure MAB as applied reinforcement learning.

Table 5. Applied RL Spectrum: MAB Variants, State Requirements, and Deployment Contexts

Approach	State Requirements	Computational Cost/Request	Infrastructure Complexity	Recommended Deployment Context
Pure MAB (Thompson Sampling)	None, stateless	O(k) sample draws	Low	Stateless slot optimization; headline/ad creative selection
Epsilon-Greedy MAB	None, stateless	O(k) comparisons	Low	Simple deployments; not recommended for low-CTR environments
Contextual Bandit (LinUCB)	User/context feature vector	O(k·d) matrix ops	Medium	Personalization with available feature vectors
Full Stateful RL	Per-user state history	O(k·|S|) policy eval	High	Sequential recommendation; game-playing agents

Table 5: Comparative overview of four applied RL approaches across deployment context, state requirements, computational cost, and infrastructure complexity. The deployment decision should be driven by a marginal benefit analysis of contextual features: pure MAB remains optimal when population-level posteriors already capture the available allocation signal, and more complex approaches are justified only when contextual features demonstrably improve allocation beyond that baseline.

Conclusion

The Batched Thompson Sampling platform described in this article demonstrates that Bayesian adaptive allocation is deployable at internet scale without compromising the statistical rigor required for peer-reviewed empirical validation. Serving 100 million users at 80,000 requests per second, the system achieved a 3.69% improvement in total clicks over the standard test-rollout baseline on live production traffic, corresponding to an estimated $100–165 million in incremental annual advertising revenue at the portal's operating scale. The dual-pipeline architecture, combining Apache Storm for sub-minute posterior updates with Hadoop/MapReduce for daily audit-quality reconciliation, provides a reusable design pattern for production reinforcement learning systems that must simultaneously satisfy low-latency optimization objectives and support rigorous outcome measurement.

Three contributions are worth distinguishing. The architectural contribution is the demonstration that decoupling the posterior update cadence from the serving cadence, the defining property of bTS relative to online Thompson Sampling, introduces negligible regret penalty on real traffic while eliminating the synchronization constraints that make per-event updates infeasible at 80,000 requests per second. The empirical contribution is the validation of Thompson Sampling's performance advantage over fixed-allocation baselines under the full noise conditions of a live production environment, extending the evidence base for Bayesian adaptive allocation beyond laboratory settings. The scholarly contribution is the model of a large-scale production system serving as an empirical laboratory for applied machine learning research, a mode of contribution whose external validity exceeds what simulation or controlled experiments can achieve.

For practitioners evaluating adaptive allocation infrastructure, the central finding is that the performance advantage of bTS is not a function of algorithmic sophistication but of the architectural decision to close the reward feedback loop continuously rather than at experiment conclusion. Any system capable of propagating reward signals back to the allocation policy at sub-minute cadence captures the majority of the adaptive allocation benefit. The infrastructure required to implement this feedback loop, a stream processing layer, a parameter store, and a cache refresh mechanism, is modest relative to the revenue impact demonstrated here. At the throughput levels where fixed-allocation testing generates material regret, Bayesian adaptive allocation is not merely preferable; it is the economically rational design choice.

References

1. Lihong Li, et al., "A contextual-bandit approach to personalized news article recommendation," ACM Digital Library, 2010. [Online]. Available: https://dl.acm.org/doi/10.1145/1772690.1772758
2. Daniel J. Russo, et al., "A tutorial on Thompson sampling," Foundations and Trends in Machine Learning, 2018. [Online]. Available: https://www.emerald.com/ftmal/article-abstract/11/1/1/1332412/A-Tutorial-on-Thompson-Sampling?redirectedFrom=fulltext
3. Shipra Agrawal, Navin Goyal, "Analysis of Thompson Sampling for the Multi-armed Bandit Problem," Proceedings of the 25th Annual Conference on Learning Theory, PMLR, 2012. [Online]. Available: https://proceedings.mlr.press/v23/agrawal12.html
4. Peter Auer, et al., "Finite-time Analysis of the Multiarmed Bandit Problem," Machine Learning, 2002. [Online]. Available: https://link.springer.com/article/10.1023/A:1013689704352
5. Ron Kohavi, et al., "Online controlled experiments at large scale," ACM Digital Library, 2013. [Online]. Available: https://dl.acm.org/doi/10.1145/2487575.2488217
6. Ramesh Johari, et al., "Peeking at A/B Tests: Why it matters, and what to do about it," ACM Digital Library, 2017. [Online]. Available: https://dl.acm.org/doi/10.1145/3097983.3097992
7. Olivier Chapelle, Lihong Li, "An Empirical Evaluation of Thompson Sampling," NeurIPS Proceedings, 2011. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2011/file/e53a0a2978c28872a4505bdb51db06dc-Paper.pdf
8. Nicolò Cesa-Bianchi and Gábor Lugosi, "Combinatorial bandits," Journal of Computer and System Sciences, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022000012000219?via%3Dihub

9. R.S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," IEEE Transactions on Neural Networks, 1998. [Online]. Available: https://ieeexplore.ieee.org/document/712192
10. [Alekh Agarwal, et al., "Taming the Monster: A Fast and Simple Algorithm for Contextual Bandits," arXiv, 2014. [Online]. Available: https://arxiv.org/abs/1402.0555
11. Peter Auer, "Using Confidence Bounds for Exploitation-Exploration Trade-offs," Journal of Machine Learning Research, 2002. [Online]. Available: https://www.jmlr.org/papers/volume3/auer02a/auer02a.pdf
12. Volodymyr Mnih, et al., "Human-level control through deep reinforcement learning," Nature, 2015. [Online]. Available: https://www.nature.com/articles/nature14236
13. Ian Osband, et al., "Deep Exploration via Bootstrapped DQN," arXiv, 2016. [Online]. Available: https://arxiv.org/pdf/1602.04621
14. William R. Thompson, "On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples," Biometrika, 1933. [Online]. Available: https://www.jstor.org/stable/2332286
15. Tor Lattimore and Csaba Szepesv´ari, "Bandit algorithms," Cambridge University Press, Cambridge, 2020. [Online]. Available: https://tor-lattimore.com/downloads/book/book.pdf