On the Costs and Benefits of Stochasticity in Stream Processing*

Raj R. Nadakuditi and Igor L. Markov, University of Michigan
{rajnrao,imarkov}@eecs.umich.edu

Abstract

With the end of clock-frequency scaling, *parallelism* has emerged as the key driver of chip-performance growth. Yet, several factors undermine efficient simultaneous use of on-chip resources, which continue scaling with Moore's law. These factors are often due to sequential dependencies, as illustrated by Amdahl's law.

Quantifying achievable parallelism can help prevent futile programming efforts and guide innovation toward the most significant challenges. To complement Amdahl's law, we focus on stream processing and quantify performance losses due to stochastic runtimes. Using spectral theory of random matrices, we derive new analytical results and validate them by numerical simulations. These results allow us to explore unique benefits of stochasticity and show that they outweigh the costs for software streams.

Categories and Subject Descriptors: B.8.2 **[Performance and Reliability]** Performance Analysis and Design Aids
General terms: Algorithms, Design, Performance, Theory
Keywords: Stream computing, latency, stochasticity

1 Introduction

Numerous studies and industry practice show that parallel processing can significantly improve power-performance trade-offs and boost chip performance beyond clock-frequency limitations. Some applications naturally exhibit parallelism, but most resist it. Achieving efficient parallelism through hardware engineering and improved software stack is a key challenge in electronic system design [3].

Past experience with attempts at greater parallelism suggests a recurring pattern — *diminishing returns* — exemplified by Amdahl's law [2]. This law assumes a chain of tasks and upper-bounds the expected overall performance improvement when only one task is improved. It was generalized for multiple active tasks in [1]. A key result is that narrow focus on component improvement usually gives a smaller benefit than intuitively expected. Amdahl's law also shows that each new processor contributes less *usable power* than the previous processor. Applied to software programs with sequential dependencies, Amdahl's law helps determine where speed-ups would be most beneficial.

Single-chip and full-system performance can be scaled significantly through *streaming* — a form of parallelism achieved by processing several dependent tasks simultaneously on unrelated data, such that job $k+1$ can commence before job k is finished. Streaming is effective in reconfigurable systems [11] and when each processing stage is implemented in dedicated hardware [24], e.g., 200 specialized stages in modern GPU pipelines. Wireless communications, cryptography, and video decoding are processed by deep pipelines with such dedicated stages as FFT, DCT, convolution, Viterbi coding, AES, motion estimation. Dedicated circuits and task-specific CPUs with ISA extensions offer greater performance and lower power than generic CPUs.

To limit idle time and power consumption of streams, stage execution times must be balanced. For example, an in-order pipeline with execution times 1,2,3 is 33% idle. Perfect balance can be impossible with irregular input [8, 11, 24], e.g., audio frames with a *busy signal* decode faster than normal voice frames; some video frames exhibit less motion than others. The performance of GPGPU programs processing irregular data is greatly affected by stochasticity due to (i) long graphics pipelines and (ii) increasing user-hardware separation encouraged by CUDA programming. Stochastic processing rates for the IBM/Sony/Toshiba Cell processor [8] can be traced to data dependencies, nonuniform memory access, cache misses, etc. Additionally, randomized algorithms (simulated annealing, Fiduccia-Mattheyses netlist partitioning and Boolean SAT solvers with random restarts) exhibit stochastic runtimes.

Our work quantifies losses in stream processing efficiency due to stochastic execution times. We analytically derive new trends and observe very good fits to numerical simulations. One remarkable trend is observed in an Amdahl-like setting with a single bottleneck in a sequential chain of processing stages, except that all stages can be active at once when processing streaming data. Here, we analytically derive and numerically confirm an unexpected *phase-transition* — speeding up a bottleneck (by allocating greater CPU resources) brings (i) diminishing returns until the threshold is reached and (ii) no returns past the threshold, *even when the bottleneck is improved*. These trends hold for a broad range of stage-time distributions.

In addition to the costs of stochasticity in stream processing, its benefits should be quantified as well. To this end, stochastic runtimes of randomized algorithms offer a unique opportunity for parallelism — mean latencies can be reduced by launching independent runs, waiting for the first run to complete, and terminating remaining runs. Our analytical results enable *a comparison of costs and benefits of stochasticity* in improving bottlenecks of software streams.

The remaining material is organized as follows. Basic concepts and terminology are reviewed in Section 2 along with relevant literature. Section 3 shows how to calculate end-to-end latency of deterministic streams and contrasts the use of *queuing theory* and *random-matrix theory* in the analysis of stochastic streams. Sections 4 and 5 derive the cost of stochasticity for balanced and unbalanced streams, resp. The assumption of exponential distributions made to derive key results is overcome in Section 6. In Section 7, we quantify the benefits of stochasticity for software streams and compare them to the costs. Conclusions are given in Section 8.

2 End-to-end Latency Analysis

Given a stream with n simultaneously active stages shown in Figure 1, we evaluate its performance on a batch of m independent jobs (audio or video frames, network traffic, etc). Each job starts at the first stage and advances sequentially through the remaining stages — once job k has been processed by stage j, it is queued up for stage $j+1$ and processed once job $k-1$ clears that stage (this is formalized in Section 3). Executions occur in-order, and inter-stage FIFOs are assumed sufficiently large (in practice, buffer contents can be spilled to secondary storage [11, Section 4.5]). Our key performance metric is *end-to-end latency* (**EEL**) $l(m,n)$ — the completion time of the last (m-th) job at the last (n-th) stage. Figure 1 illustrates a three-stage stream and the emergence of idle periods between jobs. Unlike in [10], (i) no end-to-end latency deadlines are imposed and, (ii) our FIFO inter-stage queuing model does not provision for explicit communication, simplifying EEL computations.

Stochastic stage completion times arise in several contexts [8, 11, 24]: (i) sensitivity of runtime to the complexity of input data, (ii) non-determinism due to randomized algorithms, shared resources, interrupts, and cache misses, as well as (iii) the lack of accurate information about (possibly deterministic) stage completion times. These diverse circumstances are analytically modeled by random varibles for stage completion times, making EEL a random variable.

The main objective in this work is *to quantify the impact of the probability distributions of individual stage times on the end-to-end latency statistic*. We seek to characterize the mean end-to-end latency (**MEEL**), the variance, and whenever possible provide a complete analytical description of EEL via its probability distribution.

Closest related work by Rajsbaum and Sidi [25] and, more recently, by Lipman and Stout [20], studied the impact of random processing times and transmission delays on the average number of computational steps executed by a processor in the network per unit time when attempting to synchronize over a distributed network. Our work differs in two notable ways. First, instead of bounds, we obtain exact answers. Second, while we start off our exposition in terms of exponential probability distributions, we later conclude that the specific form of the probability distributions matters less than anticipated. In particular, the new scaling phenomena we discover for the EEL statistic hold for a broad class of stage-time probability distributions.

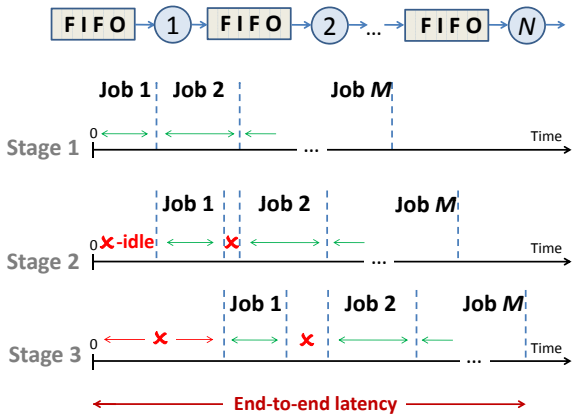

Figure 1: A timing diagram of a stream processor. Idle periods are indicated with red crosses.

Performance bottlenecks are of particular interest in our work, for the same reasons as they are in Amdahl's law. However, in the context of stream processing with balanced stages and *stochastic stage times*, the time distribution of a bottleneck stage may exhibit a *greater variance* or *longer tail*. This observation motivates designers to collect runtime statistics as in [7] so that such bottleneck stages can be identified and their impact mitigated, e.g., by allocating additional compute resources.[1] In practice, each stage may exhibit a different runtime distribution, whereas hardware designers, compiler experts and software developers have no simple way to locate bottlenecks. Even with existing profiling tools, pinpointing the "features" of runtime distribution (large variance, long tail) that affect end-to-end latency most remains difficult. Indeed, bottleneck identification and mitigation in stochastic streams have so far been more art than science. Design trade-offs to satisfy power constraints and resource limitations have been performed by trial and error.

3 Mathematical Background

Deterministic processing streams. Let $\Delta(k,n)$ be the stage time and $l(k,n)$ the end-to-end latency (EEL) for job $k \geq 1$ at stage $n \geq 1$. *In-order execution* (Section 2) means that job k can only be active at stage n after $(k-1,n)$ and $(k,n-1)$ complete. These constraints suggest an $O(nk)$-time dynamic programming algorithm for computing $l(k,n)$, based on the following recursion and straightforward *memoization*.

$$l(k,n) = \max\{l(k-1,n), l(k,n-1)\} + \Delta(k,n) \qquad (1)$$

with $l(k,0) = l(0,n) = 0$. To solve this recursion, consider the lattice $\{(i,j) \in \mathbb{Z}^2 | 1 \leq i \leq k, 1 \leq j \leq n\}$. The following solution can be proven by induction [16, Section 2],

$$l(k,n) = \max_{\pi(k,n)} \sum_{(i,j)\in\pi} \Delta(i,j) \qquad (2)$$

Here, $\pi(k,n)$ is the set of all monotonic lattice paths of length $k+n-1$ from $(1,1)$ to (k,n), as shown in Figure 2. These monotonic paths capture all possible critical paths during stream's execution. We give *closed-form expressions* for $l(k,n)$ for two cases in discussions after Formulas 6 and 12, and contrast them with results for the stochastic case.

[1] If stage times are independent, then processing the same data at the same stage on multiple processors (and using the first available result) can reduce variance and shorten the tail of the resulting time distribution.

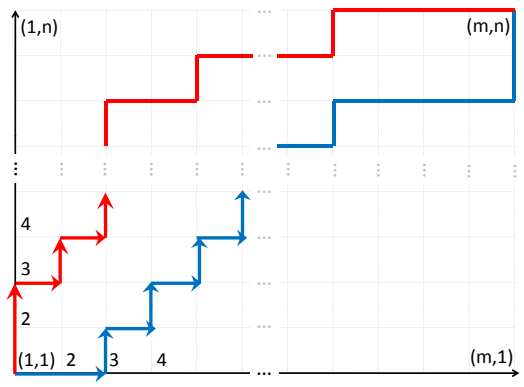

Figure 2: Monotonic paths $(1,1) \rightarrow (m,n)$ used to model end-to-end latency from Figure 1 by Formula 2 with $k = m$.

Stochastic queuing theory [16] studies the statistics of Formula 2 when $m \gg n$ and *vice versa*. For stream processing, this assumption can be justified in the traditional setting where the number of stream stages remains limited, i.e., $n = O(1)$, but the number of jobs is large. Under these assumptions EEL is *normally* distributed via the law of large numbers [16]. Consequently, the asymptotic scaling of the mean is straightforward, and the impact of a small number of bottlenecks is what one would intuitively expect. We note that the "interacting-particle system" interpretation [26] used by queuing theory simplifies the analysis by neglecting the interaction between stages — this is a reasonable assumption when $m \gg n$ or $n \gg m$, but not when n and m are both small or when both are large.

Numerous parallel cores can be useful in deep streams when the number of streaming jobs is sufficiently high. To this end, the RAMP project at Berkeley is developing a massive FPGA-based emulator to study large-scale behavior of many-core systems [11], recently reaching the 1008-processor milestone [6]. However, current supercomputers integrate 300,000 cores, and "supercomputers with 100 million cores are coming by 2018" [28]. This motivates our focus on *analytical estimates*. When both n and m are large in the stream model of Section 2, the interactions between stochastic stage-time distributions accumulate, and the assumptions made in queuing theory are no longer valid (see discussion after Formula 6). The Gaussian distribution predicted by queuing theory transitions into the *type-2 Tracy-Widom* distribution studied in the *spectral theory of random matrices* [18, 19], and the asymptotic scaling of variance changes as well. Figure 3 contrasts the two distributions.

The type-2 Tracy-Widom distribution (TW_2) describes the largest eigenvalue of random Hermitian matrices [15] and arises in combinatorics. If π is a random n-element permutation, then the length of the *longest increasing subsequence* of π converges (with appropriate scaling and re-centering) to the TW_2 distribution as $n \to \infty$ [4]. For exponentially-distributed stage times, Formula 2 is related to the LONGEST INCREASING SUBSEQUENCE PROBLEM. Empirical evidence in [12, 14] suggests viewing the TW_2 distribution as a nonlinear variant of the law of large numbers for EEL. Thus, we use TW_2 and related mathematics to perform accurate analysis of stochastic streams.

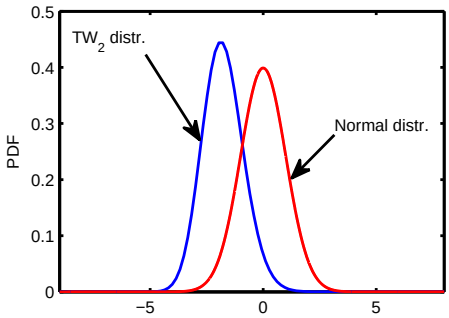

Figure 3: The Tracy-Widom and normal distributions.

4 Analysis of Balanced Stochastic Streams

The research strategy pursued in this work is to initiate analysis in terms of balanced *exponentially* distributed stage times. For $\lambda > 0$, we consider the distribution with the pdf

$$f(t;\lambda) = (1/\lambda)\exp(-t/\lambda), \qquad t \geq 0, \tag{3}$$

where the mean and standard deviation are λ. We extend key results to a broader class of probability distributions in Section 6, but note here that exponential distributions are the *worst* from an information-theoretic perspective.

In a practical setting, we might not know the entire stage-time distribution, but we can usually estimate its mean. From the many probability distributions with a given mean, we distinguish the unique distribution that maximizes the Shannon entropy[2] because it offers *the most random* probabilistic model subject to what is known. Among all probability distributions supported on $t \geq 0$ with mean λ, the *exponential distribution* exhibits maximum entropy [9, Chapter 11]. This *worst-case* information-theoretic argument was previously used by Rajsbaum and Sidi [25] to motivate the focus on exponential distributions in a setting related to ours.

The cost of stochasticity. Assume n stages with times that are independent and exponentially distributed with parameter $\lambda > 0$. Let $G_{n,m}$ be an $n \times m$ complex-valued matrix with independent, normally distributed entries with mean 0 and variance 1. Let $S_{n,m} = G_{n,m}G^*_{n,m}$. Johansson [18] shows that $l(m,n)$ and the largest eigenvalue $\lambda_{\max}$ of $S_{n,m}$ have the same distribution:

$$l(m,n) \sim \lambda_{\max}(S_{n,m}) \qquad \forall m,n. \tag{4}$$

Since the TW_2 distribution asymptotically describes $\lambda_{\max}$ [19], we are able to highlight the important qualitative trends of $l(m,n)$ (derivation omitted due to page limitation).

$$E\left[l(n,m)\right] = \lambda(\sqrt{n}+\sqrt{m})^2 - 1.7711\,\lambda\frac{(\sqrt{m}+\sqrt{n})^{4/3}}{(mn)^{1/6}} \tag{5}$$

$$\mathrm{Var}\left[l(n,m)\right] = 0.8132\lambda^2\left(\frac{(\sqrt{m}+\sqrt{n})^{4/3}}{(mn)^{1/6}}\right)^2 \tag{6}$$

Figure 4 illustrates predicted scaling behavior. To this end, note that m identical jobs streamed through n stages with identical *deterministic* latencies λ take $\lambda(n+m)$ time. But MEEL in the stochastic case scales as $\lambda(\sqrt{n}+\sqrt{m})^2 = \lambda(n+m+2\sqrt{nm})$.

Hence, the cost of stochasticity scales as $2\lambda\sqrt{nm}$.

Observe that for $n \gg m$ or $m \gg n$, the term $2\lambda\sqrt{nm}$ is asymptotically negligible because $2\sqrt{nm} = o(n+m)$, but it may contribute up to 50% of EEL when $n = \Theta(m)$. This first-order result is alluded to in the seminal paper on queuing theory by Glynn and Whitt [16]. However, the *law of vanishing returns* stated next is new and exploits results from random-matrix theory [5].

[2] A single number that is commonly used to measure the amount of uncertainty contained in a probability distribution [9].

n	m	MEAN Experiment	MEAN Theory	VARIANCE Experiment	VARIANCE Theory
5	5	13.1024	12.3685	9.4351	15.0981
10	10	30.9954	30.3849	18.6033	23.9668
20	20	68.3172	67.8858	33.0268	38.0449
40	40	145.0274	144.7371	55.1251	60.3926
80	80	300.9902	300.7699	90.0644	95.8673
160	160	615.9515	615.7717	148.8302	152.1799
320	320	**1249**.4124	**1249**.4742	236.0294	241.5705
480	480	**1885**.7545	**1885**.0567	311.7331	316.5469
640	640	**2521**.6221	**2521**.5399	374.6064	383.4693
1000	1000	**3955**.4348	**3955**.3710	506.5496	516.3498

Table 1: Empirical mean and variance of end-to-end latency, computed over 1000 Monte-Carlo trials, compared to theoretical predictions in Formulas 5 and 6, respectively.

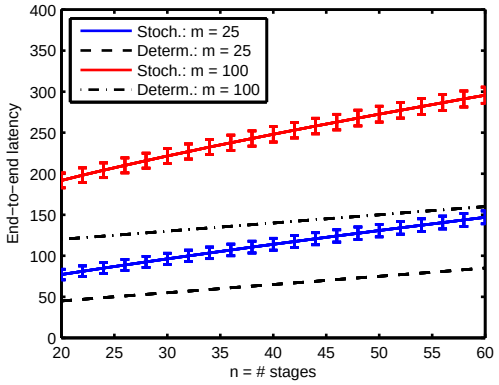

Figure 4: Theoretical scaling of mean end-to-end latency with the number of stages for exponentially distributed stage times. Solid lines illustrate Formula 5, and error bars give standard deviation according to Equation 6. For comparison, dashed lines show latencies in a deterministic stream.

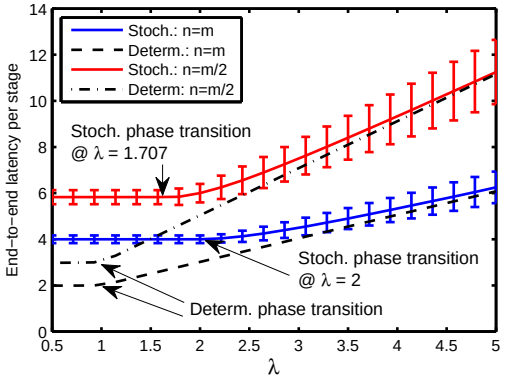

Figure 5: The effect of a *single bottleneck* stage with $\lambda_1 > 0$. End-to-end latency (normalized per stage) given by Formula 7 exhibits a *phase transition* at the critical value $\tau = 1 + \sqrt{\frac{n}{m}}$. Error bars show standard deviation as per Formula 6. Dashed lines give a deterministic baseline as in Figure 4.

A law of vanishing returns. Suppose that $n-1$ stage times are independent and exponentially distributed with parameter $\lambda = 1$, but the one remaining *bottleneck* stage exhibits exponentially distributed stage-time with $\lambda_1 > 1$. Equation 4 still holds, except that results from [5] imply a *phase transition* (derivation omitted):

$$E\left[\frac{l(m,n)}{n}\right] \approx \begin{cases} (1+\sqrt{\frac{m}{n}}) & \text{if } \lambda_1 \leq 1+\sqrt{\frac{n}{m}} \\ \lambda_1 \frac{m}{n}\left(1+\frac{n/m}{\lambda_1-1}\right) & \text{otherwise.} \end{cases} \quad (7)$$

$$\mathrm{Var}\left[\frac{l(m,n)}{n}\right] \approx \begin{cases} \frac{0.8132}{n^2}\left(\frac{(\sqrt{m}+\sqrt{n})^{4/3}}{(mn)^{1/6}}\right)^2 & \lambda_1 \leq 1+\sqrt{\frac{n}{m}} \\ \lambda_1^2 \frac{m}{n^2}\left(1-\frac{n/m}{(\lambda_1-1)^2}\right) & \text{otherwise.} \end{cases} \quad (8)$$

Here, we have normalized the mean and variance *per stage*, so that distribution-dependent higher-order terms can be neglected. Figure 5 illustrates this emergent scaling behavior: when the mean of the bottleneck-stage time is below the critical threshold $\tau = 1+\sqrt{n/m}$, then, surprisingly, the end-to-end latency of the system becomes insensitive to changes in λ_1. The same holds for $o(n)$ bottlenecks. This result can be interpreted as an analog of Amdahl's law, for stream processing with stochastic runtime distributions.

Numerical validation of the formulas presented so far was performed by extensive Monte-Carlo simulations in Matlab. Table 1 shows excellent agreement between analytical results and numerical simulations. Figure 6 graphically illustrates empirical accuracy of our bottleneck predictions. Notice that the errors decrease as parameters grow — this is expected for asymptotic estimates. The variances in Table 1 are always *over-estimated*, betraying (distribution-dependent) higher-order terms missing from our estimates.

5 Analysis of Unbalanced Stochastic Streams

We now generalize the previous setting by assuming that the n stage times are independent and exponentially distributed with *different* parameters $\lambda_1, \ldots, \lambda_n$. In Section 6, we show how these results provide insight for the setting where the streams have balanced means but unbalanced variances.

The cost of stochasticity. As in the context of Relation 4, let $X = G_{n,m}G_{n,m}^*$ but now let Σ be a diagonal matrix with entries $\lambda_1, \ldots, \lambda_n$ (parameters of exponential stage-time distributions). Let $S_{n,m} = \Sigma^{1/2} X \Sigma^{1/2}$. Applying the *random-matrix theory* [15] to $S_{n,m}$ we obtain

$$\frac{\lambda_{\max}(S_{n,m}) - m\mu_{n,m}}{\sqrt[3]{m}\,\sigma_{n,m}} \xrightarrow{D} \mathrm{TW}_2, \quad (9)$$

where $\xrightarrow{D}$ denotes *almost sure* convergence and TW_2 is the type-2 Tracy-Widom distribution from Section 3. Here, $\mu_{n,m}$ and $\sigma_{n,m}$ are given by

$$\mu_{n,m} = \frac{1}{c}\left(1 + \frac{1}{m}\sum_{i=1}^{n}\frac{\lambda_i c}{1-\lambda_i c}\right) \quad (10)$$

$$\sigma_{n,m} = \frac{1}{c^3}\left(1 + \frac{1}{m}\sum_{i=1}^{n}\left(\frac{\lambda_i c}{1-\lambda_i c}\right)^3\right), \quad (11)$$

where c is the unique solution in $[0, 1/\max(\lambda_1, \ldots, \lambda_n)]$ of the equation

$$\frac{1}{m}\sum_{i=1}^{n}\left(\frac{\lambda_i c}{1-\lambda_i c}\right)^2 = 1. \quad (12)$$

We are able to prove (derivation omitted due to page limitation) that Relation 4 *also holds for unbalanced stochastic streams*, facilitating accurate analysis of the distribution

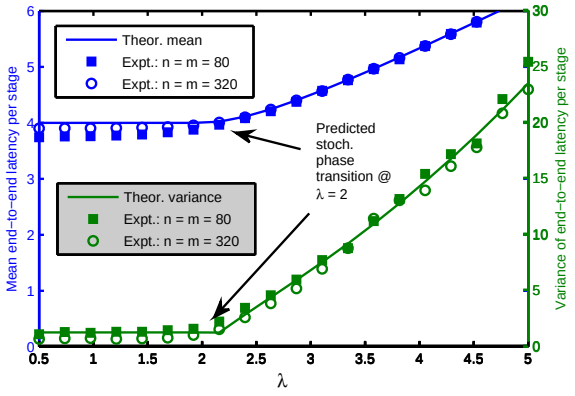

Figure 6: Empirical evaluation of analytical predictions (solid lines) for the mean (left axis, Formula 7) and the variance (right axis, Formula 8) against λ of one bottleneck. Datapoints are averaged over 1000 Monte-Carlo trials.

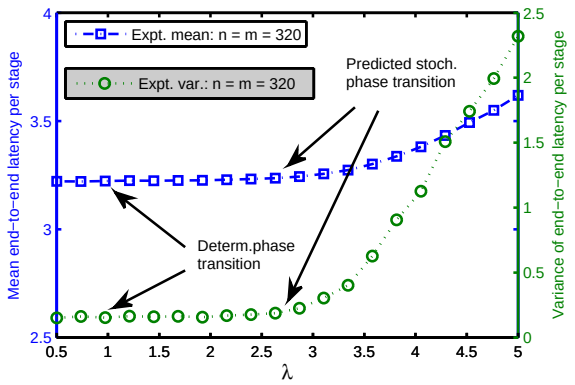

Figure 7: The law of vanishing returns for an unbalanced stochastic stream with normally-distributed stage-times. Empirical datapoints are overlaid against theoretical predictions (lines) for the mean (left axis) and the variance (right axis) of end-to-end latency.

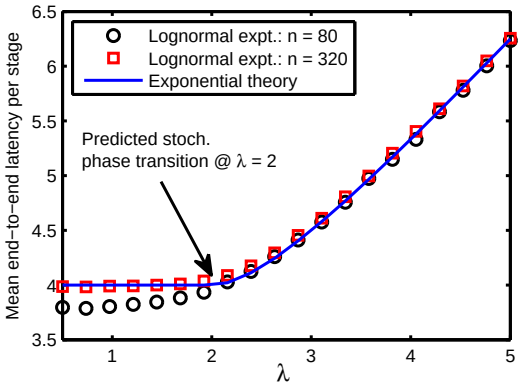

Figure 8: Theoretical predictions for MEEL with *exponentially* distributed stages and a single bottleneck (λ) compared to simulation results averaged over 1000 independent trials for *log-normally* distributed stages. Equally good fits were produced up to $n = 1000$ (not shown).

of $l(m,n)$. To this end, note that m identical jobs streamed through n stages with deterministic latencies $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ take $\sum_{i=1}^{n} \lambda_i + m\lambda_1$ time. In contrast, in the stochastic case, MEEL scales with $m\mu_{n,m}$ as given by Formula 10, and the *cost of stochasticity* scales as

$$\frac{1}{c}\left(m + \sum_{i=1}^{n} \frac{\lambda_i c}{1-\lambda_i c}\right) - \left(\sum_{i=1}^{n} \lambda_i + m\lambda_1\right) > 0 \qquad (13)$$

where c is the solution of Equation 12.

A law of vanishing returns. Suppose that $n-1$ stage-times are independent and exponentially distributed with parameters $\lambda_1, \ldots, \lambda_{n-1}$, while the bottleneck stage time is exponentially distributed with parameter $\lambda_n > \max\{\lambda_i\}$. To describe the bottleneck's impact on end-to-end latency, we recall a very recent result in random-matrix theory [22]. It establishes that the largest eigenvalue $\lambda_{\max}(S_{n,m})$ of $S_{n,m}$, constructed above, experiences a *phase transition* at the threshold $\tau > 0$ that is a solution of the equation [15]

$$\frac{n}{m} = \frac{1}{n}\sum_{i=1}^{n-1}\left(\frac{\lambda_i}{\tau - \lambda_i}\right)^2. \qquad (14)$$

When $\lambda_n > \tau$,

$$E\left[\frac{l(m,n)}{n}\right] \approx \frac{1}{n}\left(1 + \frac{1}{m}\sum_{i=1}^{n-1}\frac{\lambda_i}{\lambda_n - \lambda_i}\right) \qquad (15)$$

The variance can be similarly computed, but we omit it here. To translate the results of random-matrix theory into the context of unbalanced streams, we build on sophisticated mathematical techniques from [5] (details omitted due to page limitations) generalized from [18]. We are able to prove that the above threshold applies to the end-to-end latency of an unbalanced stochastic stream. Specifically, if the mean bottleneck-stage time is below the critical threshold τ, then MEEL of the full system is insensitive to changes in λ_n. This result also covers the case of $o(n)$ bottlenecks in an unbalanced stream. We have extended all stated results to accommodate time-varying λ_i parameters.

6 Extension to a Broader Class of Distributions

So far, our results assume *exponential* stage-time distributions. We now offer several types of evidence suggesting that these results hold for a broader class of distributions.

Theoretical considerations. Similar generalizations have been extensively studied in random-matrix theory and are exemplified by the well-known *universality conjecture* [12]. This conjecture considers matrix $S_{n,m}$ in Relation 4 and replaces the Gaussian distribution by an arbitrary distribution f_δ with the same mean and variance. The claim is that the largest eigenvalue will be described by the same TW_2 distribution, as long as the fourth moment of f_δ is bounded. This conjecture is supported by numerical data [12, 14], is commonly viewed as a nonlinear law of large numbers for max-eigenvalues, and mirrors what has been recently proven for min-eigenvalues by Tao and Vu [27]. We state an analogous nonlinear law of large numbers for MEEL.

Conjecture: Consider two n-stage stochastic streams where stage-time distributions are in *stochastic order*.[3] The first stream exhibits arbitrary stage distributions with means μ_i, variances σ_i^2 and bounded fourth moments. The second stream exhibits exponential stage-time distributions with parameters $\lambda_i = \sigma_i$ and additional *linear shifts* to adjust their means to match μ_i. Then the two streams exhibit the same cost of stochasticity and the same threshold τ below which improvements to MEEL latency vanish.[4]

Empirical evidence for normal distributions. Assume $n-1$ stages with mean $\mu = 1$ and variance $i/(n-1)$ at the i-th stage. Let the bottleneck occur at the n-th stage, normally distributed with mean $\mu = 1$ and variance λ^2. The cost of stochasticity can be computed using Formula 13 with $\lambda_i = i/(n-1)$ and predicts experimental results with 5% accuracy. The phase-transition threshold predicted by Equation 14 matches empirical results, as shown in Figure 7.

Empirical evidence for log-normal distributions with pdf

$$f(t;\mu,\sigma) = \frac{1}{t\sigma\sqrt{2\pi}}\exp\Big(-\frac{(\log t - \mu)^2}{2\sigma^2}\Big), \qquad t > 0. \quad (16)$$

We set $\mu = \log(\lambda/\sqrt{2})$, $\sigma = \sqrt{\log 2}$ to match the mean and variance of the exponential distribution with parameter λ. Our earlier predictions are validated in this case by simulation data shown in Figure 8.

Empirical evidence for *truncated* power-law distributions (not shown here) also confirms phase transitions. However, as the amount of truncation grows, the stochastic phase transition converges to the deterministic phase transition.

[3] For real random variables A and B, $A \leq B$ when $\Pr[A > x] \leq \Pr[B > x]\ \forall x$.

[4] Asymptotic equality neglects distribution-dependent higher-order terms.

7 Comparing Costs to Benefits of Stochasticity

Recall that conclusions can be drawn from Amdahl's law that are relevant to both hardware design and software optimization. In a similar spirit, we now consider *software streams with stage-times* that are randomized even for identical input data. In commercial EDA tool-chains, examples include (i) random restarts in leading DPLL-style SAT-solvers, (ii) the Fiduccia-Mattheyses heuristic for netlist partitioning used with randomized initial partitions, and (iii) the framework of simulated annealing, used in circuit placement and chip floorplanning, where *move selection* during local search is randomized. Additionally, numerical EDA algorithms often exhibit very different convergence for alternative settings and algorithms, and trying multiple settings on identical inputs in parallel was shown useful [13].

Using additional computational cores can reduce the means of the stochastic stage-times without reworking the algorithms. This is achieved by running multiple independent jobs on identical inputs. Due to stochasticity, some jobs will finish earlier, at which point the other equivalent jobs can be terminated. Here we observe that the minimum of s independent, exponentially-distributed random variables with parameters $\lambda_1 \ldots \lambda_s$ (as in Formula 3) is also exponentially distributed, with parameter $1/(1/\lambda_1 + ... + 1/\lambda_s)$. For s independent identical distributions, the mean is λ/s.

In the setting of Section 5, consider an exponentially distributed bottleneck stage with mean λ_n. By the law of vanishing returns, only $\lceil \lambda_n/\tau \rceil < \lceil \lambda_n / \max(\lambda_1, \ldots, \lambda_{n-1}) \rceil$ *identical cores achieve the maximum possible gain*, and no additional independent starts can improve MEEL, despite improving the bottleneck.[5] In Figure 9, this technique is *greedily* applied to two bottlenecks ($\lambda_1 = 15$, $\lambda_2 = 30$). A more effective *balanced allocation* splits s available processors among k bottlenecks as $\Sigma_i^k s_i = s$ so as to minimize $\Sigma_i^k (\lambda_i/s_i)$.

The benefits of stochasticity in software streams can be contrasted with its *costs*. For example, in Figure 5 at $\lambda = 5$ the costs (gaps between solid and dashed lines) are small, but the benefits can produce a net $2\times$ reduction in MEEL.

8 Conclusions

Our work establishes a far-reaching connection between (i) the performance evaluation of stream processing and (ii) the *spectral theory of random matrices* [12, 14, 18, 19, 27]. The analytical models we derived for the costs of stochasticity in stream processing are confirmed by numerical simulations with high accuracy and exhibit previously unknown scaling trends, such as a *a law of vanishing returns*. To the best of our knowledge, relevant results from *queuing theory* [16] only cover the case of balanced streams, and only to the first order. In contrast, our analytical predictions agree with empirical data for both balanced and unbalanced stochastic streams with several types of stage-time distributions, where only the mean and the variance seem to affect key parameters of interest. In the random-matrix setting [22], it has been theoretically established that correlations only affect (negligible) higher-order terms.

We have produced specific guidelines on how to allocate parallel cores to speed-up bottlenecks in stochastic software streams. In this context, we illustrate how the benefits of stochastic runtimes may outweigh their adverse impact on end-to-end latency of stream processors.

[5] Our analysis neglects higher-order terms. Empirically, a very small improvement may be observed, as in Figures 6 and 8.

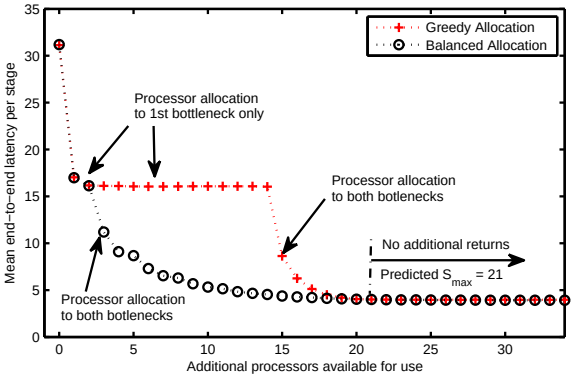

Figure 9: Two strategies for processor allocation in a two-stage stochastic stream with $\lambda_1 = 15$ and $\lambda_2 = 30$.

References

[1] K. Agrawal, C.E. Leiserson, Y. He, W.-J. Hsu, "Adaptive work-stealing with parallelism feedback," *ACM. Trans. Comp. Sys.* 26(3),#7, 2008.

[2] G. M. Amdahl,"Validity of the single-processor approach to achieving large-scale computing," *AFIPS Joint Comp. Conf.*, pp. 483-485, 1967.

[3] K. Asanovic et al, "A view of the parallel computing landscape," *Comm. of the ACM* 52(10), pp. 56-67.

[4] J. Baik, P. Deift, and K. Johansson, "On the distribution of the length of the longest increasing subsequence of random permutations," *J. of Amer. Math. Soc.* 12(4), pp. 1119–1178, 1999.

[5] J. Baik, G. Ben Arous, S. Péché, "Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices," *Annals of Probability*, pp. 1643-1697, 2005.

[6] D. Burke et al, "RAMP Blue: implementation of a manycore 1008-processor system," *Reconfig. Sys. Summer Inst.* (RSSI) 2008.

[7] G. Z. Chrysos et al. "Method for estimating statistics of properties of instructions processed by a processor pipeline," *US Patent 5,809,450*.

[8] R. L. Collins, L. P. Carloni, "Flexible filters: load balancing through backpressure for stream programs," *EMSOFT* 2009, pp. 205-214.

[9] T. M. Cover, J. A. Thomas,"Elements of information theory,"Wiley'06.

[10] A. Davare et al, "Period optimization for hard real-time distributed automotive systems," *DAC* 2007, pp. 283-288.

[11] A. DeHon et al, "Stream computations organized for reconfigurable execution," *Microprocessors & Microsystems* 30(6), pp. 334-354, 2006.

[12] P. Deift, "Universality for mathematical and physical systems," *Int'l Congress of Mathematicians* (I), pp. 125-152,*Eur.Math.Soc.* Zürich '07.

[13] W. Dong, P. Li, "Parallelizable stable explicit numerical integration for efficient circuit simulation," *DAC* 2009, pp. 382-385.

[14] A. Edelman, N. R. Rao, "Random matrix theory," *Acta Numerica*, vol. 14, pp. 233-297, 2005.

[15] N. El Karoui, "Tracy-Widom limit for the largest eigenvalue of a large class of complex sample covariance matrices," *Annals of Probability* 35(2):663-714, 2007.

[16] P. W. Glynn and W. Whitt, "Departures from many queues in series," *The Annals of Applied Probability*, 1(4):546-572, 1991.

[17] M. D. Hill, M. R. Marty, "Amdahl's law in the multicore era," *IEEE Computer*, July 2008.

[18] K. Johansson, "Shape fluctuations and random matrices," *Comm. in Math. Phys.*, 209(2):437-476, 2000.

[19] I. M. Johnstone, "On the distribution of the largest eigenvalue in principal components analysis," *Annals of Statistics*, pp. 295–327, 2001.

[20] J. Lipman, Q.F. Stout, "A performance analysis of local synchronization," *SPAA* 2006, p.260.

[21] J. B. Martin, "Large tandem queueing networks with blocking," *Queuing Systems*, vol. 41(1), pp. 45–72, 2002.

[22] R. R. Nadakuditi, J. W. Silverstein, "Fundamental limit of sample generalized eigenvalue based detection of signals in noise using relatively few signal-bearing and noise-only samples," *J. Sel. Topics in Signal Proc.*, `http://arxiv.org/abs/0902.4250`, 2009.

[23] N. O'Connell, "Random matrices, non-colliding processes and queues," *Séminaire de Probabilités, XXXVI*, vol. 1801, pp. 165-182.

[24] M. Qiu, E. H-M. Sha, "Cost Minimization while Satisfying Hard/Soft Timing Constraints for Heterogeneous Embedded Systems," *ACM Trans. Design Automation* 14(2) 2009.

[25] S. Rajsbaum, M. Sidi, "On the performance of synchronized programs in distributed networks with random processing times and transmission delays," *IEEE Trans. on Par. and Distr. Sys.* 5(9):939–950, 1994.

[26] R. Srinivasan, "Queues in series via interacting particle systems," *Mathematics of Operations Research* 18(1), pp. 39–50, 1993.

[27] T. Tao, V. Vu, "Random matrices: the distribution of the smallest singular values," `http://arxiv.org/abs/0903.0614`.

[28] P. Thibodeau, "Supercomputers with 100 million cores coming By 2018," *Computerworld*, 11/16/09.