# The Impact of Noise on the Scaling of Collectives: The Nearest Neighbor Model

**[Extended Abstract]**

Nisheeth K.Vishnoi[1]

Computer Science Division
University of California Berkeley, CA 94720.
nisheeth.vishnoi@gmail.com

**Abstract.** This paper presents a theoretical study of the impact of noise on the scaling of a cluster when the processors participate in “local” collectives with their nearest neighbors. The model considered here is an extension of that introduced in [9] for understanding the effect of noise on the scaling of “global” collectives in large clusters. In this paper, the scaling is studied with respect to three fundamental aspects: (1) the distribution of noise: whether it is heavy or light tailed; (2) the temporal independence of noise; (3) the topology of the cluster. When the noise has a “light” tail and is temporally independent, it is shown that the cluster scales well, i.e., the slowdown per phase is just proportional to the (logarithm of the) maximum degree of the communication topology. This implies that for popular topologies such as grids and toruses the slowdown per phase is just a constant factor, which is independent of the number of processors. In the light tailed case, assuming only a weak temporal independence, a general upper bound is derived in terms of an “expansion” parameter of the communication topology. For grid-like graphs this establishes an exponential speedup compared to what was shown for global collective operations in [9].

## 1 Introduction

*Motivation.* It has been observed by several researchers, see [1–3], that the “throughput” of several high performance computing (HPC) clusters running scientific applications drops as the number of processors in the cluster increases. It has been suggested in [1, 2] that one of the main causes of this is the “noise” in the processors of the cluster in the form of overheads such as daemons and interrupts. Given the exorbitant amount of resources invested in building such systems, it becomes extremely important to understand the reasons for this loss in efficiency and, if possible, rectify it. As a first formal step towards achieving this, the impact of noise on the scaling of these clusters was explained via a stochastic model in [9]. They abstracted a typical scientific parallel application in which each node in the cluster repeatedly performs a “phase” which consists of a computation stage followed by a “global” collective stage. A collective involves all processors coming to a common state once they are through with their computation stage, from where on they resume the next phase. Hence, a large amount of noise in one of the processor, which would result in an untimely completion of the work assigned to it, may end up slowing down everyone. The results of [9] showed, in particular, that even in the most favorable case when the noise in each processor is distributed according to the exponential distribution, the per phase throughput drops by a factor of $\Omega(\ln N)$, where $N$ denotes the number of processors. Or, if each processor would do $w$ work per phase when there was no noise, it would end up doing only $O(w/\ln N)$ work per phase. Hence, as the number of processors tends to infinity, the throughput goes to zero, somewhat defeating the purpose why such clusters are in place. The main objective of such a study is to understand the problem of noise and the ways in which it has the potential to degrade system performance. The end goal being to reduce the impact of this necessary evil, and hence, to improve the performance of HPC systems.

*This Work.* In this paper we study the impact of noise on the performance of a cluster when the processors participate in "local" collectives, i.e., with their nearest neighbors in the communication topology. We suitably extend the stochastic model of [9] who did exactly this, albeit, for global collectives. Before we go on, let us briefly (at the expense of being imprecise) outline the model.[1] Consider a parallel program with $N$ threads running on a system with $N$ processors. The system is dedicated to running the same program repeatedly, each such run being referred to here as a *phase*. Each phase consists of two stages (1) a computation stage: In this stage each processor is supposed to do $w$ amount of work. (2) A communication stage: In this stage each processor communicates with a specified set of processors, referred to as its *neighbors*. This communication pattern is referred to as the communication topology and is captured by a directed graph on $N$ processors: $G$. But, because of noise, each processor has to devote an overhead time which in phase $j$ for the $i$-th processor is $\delta_{i,j}$ and is assumed to be distributed according to a random variable $\delta$ with mean $wf$, where $f$ is a fixed number in the interval $[0, 1]$. It can be safely assumed that the noise is independent spatially, i.e., in the absence of coordinated OS policies, the noise across different processors is uncorrelated. It sometimes may be difficult to argue, and not true, that the noise is uncorrelated temporally, i.e., with $j$, although one may expect noise to be uncorrelated in the short range and long range. In this setup, we explore the scaling, or per phase work per processor with respect to three fundamental aspects: (1) the tail of the distribution of noise: $\mathbf{Pr}[\delta \geq t]$; (2) the temporal independence of noise: the correlation patterns of $\{\delta_{i,j}\}$ with $j$; (3) the communication topology: $G$.

*Our Results.* Our results can be informally summarized in the following points. To see the precise technical statement, the reader is referred to the corresponding theorem.

[1] A detailed description of the model appears in Section 2.

1. For the case when the noise is temporally independent and light tailed, we prove that time per phase is "sharply concentrated" around the quantity $w + wf(\ln \mathit{\Delta}) + \tau$, here $\mathit{\Delta}$ is exactly one more that the largest in-degree of the communication topology $G$. (See Theorem 7.)
2. For the case when the noise is "weakly" temporally dependent, light tailed and the growth rate of the topology is bounded, we prove that the time per phase is upper bounded by the quantity $w + wf(\ln \ln N) + \tau$. (See Theorem 8.)
3. For the case when the noise is heavy tailed and temporally independent, we show that the topology has little effect and the time taken per phase is almost the same as if the processors were involved in a global barrier. (See Section 4.) This has been proved independently in [11].
4. We also prove sharp results when the communication patterns are multi-round, such as a binary tree, or one that arises in the Fast Fourier Transform. (See Theorems 9 and 10.)

*Related Work.* Our work is a natural follow-up to that of [9]. The work of [10] takes an in-depth look into the theoretical model of the impact of noise on the collectives studied in [9] by validating its accuracy against data collected from production clusters. Recently, we came across the work of Lipman and Stout [11] who have, independently of this work, considered a problem which reduces to a similar stochastic problem considered in our paper. The main result of their paper is a tight bound for the case when the noise is light tailed, temporally independent and the topology is a directed cycle. This can be seen as a special case of our first result.

The problem on how to alleviate the problem of noise has been looked by several research teams [3–8]. Our theoretical understanding could potentially be coupled with ideas from these works to alter the noise in the systems, as a function of the communication topology, and help improve the performance of these systems considerably.

*Organization.* In Section 2 we present a description of the model considered in the paper. In Section 3 we analyze the model presented in Section 2. The results presented in Section 3 are general and rigorous. In Section 4 we describe the results obtained in Section 3.2 for the light and heavy tailed distributions. The canonical example of light tailed distribution we consider is the exponential distribution, while for the heavy tailed distribution, we consider the Pareto distribution. In Section 5 we present the results pertaining to multiple communications per phase. For the lack of space, most of the proofs are deferred to the full version of this paper.

## 2 The Setup

### 2.1 Modeling the Application

In this section we present the stochastic model considered in this paper. Since it is an extension of the model presented in [9], we choose to keep the terminology as similar as possible. The basic setup is a parallel program with $N$ threads running on a system with $N$ processors. Typically, the system is dedicated to running the same program repeatedly, each such run being referred to here as a *phase*, albeit on different inputs every time. Each phase is a composition of two kinds of stages:

- Computation stage: In this stage the processor does computation without any message exchange or I/O activity.
- Communication stage: Once the processor has finished its computation stage, it enters the communication stage. In this stage each processor communicates with a specified set of processors, referred to as its *neighbors*. In this stage there is negligible computation except that associated to communication.

For the sake of simplicity, as has been done in the previous papers on this, we assume that for every processor, each phase just consists of a computation stage followed by a communication stage. Later, we will consider the case of FFT where this is not the case, and each phase comprises of a number of alternations of computation stages with communication stages. Thus, every phase is characterized by the amount of work assigned to each processor and the pattern of communication that occurs between them. Formally, in phase $j \geq 1$, let $W_{i,j}$ be the work assigned to processor $1 \leq i \leq N$, which would be completed by it in (deterministic) time $w_{i,j}$ was there no noise. The communication in phase $j$ can be captured by a directed graph (possibly with loops) $G_j([N], E_j)$. Here, $E_j$ consists of the directed edges along which communication happens. We represent an edge as $(i_1, i_2)$, meaning that $i_1$ communicates a message to $i_2$. The time this communication takes is $c_{i_1,i_2,j}$. This completes the overview of a phase. Now we proceed to a detailed quantitative description of a phase.

**A Phase.** Let $t^s_{i,j}$ denote the time when the $i$-th thread begins phase $j$, and let $t^f_{i,j}$ denote the time when it ends the computation stage in the $j$-th phase. Let $W_{i,j}$ denote the amount of work carried out by thread $i$ in the computation stage of the $j$-th phase. If the system is noiseless, the time required by processor $i$ to finish work $W_{i,j}$ in its $j$-th phase will be a deterministic quantity, which we denote by $w_{i,j}$. This quantity typically depends on several characteristics of the processor such as its clock frequency, its architectural parameters, and the state of the node (such as cache contents) just before the $j$-th phase is entered. Therefore, $t^f_{i,j} - t^s_{i,j} = w_{i,j}$. Due to the presence of noise, the time taken by processor $i$ to finish the work $W_{i,j}$ is typically not a constant. There will be a variable component that represents the time consumed to service the daemons and other asynchronous events. This is captured by a random variable $\delta_{i,j}$. More precisely, $t^f_{i,j} - t^s_{i,j} = w_{i,j} + \delta_{i,j}$. Let $f_{i,j} \in [0,1]$ be the fraction representing the system overhead for the processor, i.e., let $f_{i,j} := \frac{E[\delta_{i,j}]}{w_{i,j}}$. Thus, we may think of the noise as a random variable $\eta_{i,j} := \frac{\delta_{i,j}}{f_{i,j} w_{i,j}}$ with mean one. Thus, we may write the wall-clock time taken by processor $i$ for the $j$-th phase as $t^f_{i,j} - t^s_{i,j} = w_{i,j}\left(1 + f_{i,j}\eta_{i,j}\right)$.

For $j \geq 1$, phase $j+1$ starts for the $i$-th processor when it has completed phase $j$. The first phase starts at time zero for all processors. The $j+1$-th phase ends for the $i$-th processor when it has completed its computation in the $j+1$-th phase, as well as, all the processors $i'$ such that $(i', i) \in E_{j+1}$ have ended the computation in their $j+1$-th phase. Define $N_j(i)$ to be the set of processors which have an edge directed towards $i$ in $G_j$. With this notation, one can define the time taken by the $i$-th processor to complete the $j+1$-th phase, denoted by $T_{i,j+1}$ as

$$\max\left\{T_{i,j} + w_{i,j}(1 + f_{i,j}\eta_{i,j}), \max_{i' \in N_{j+1}(i)} \{T_{i',j} + w_{i',j}(1 + f_{i',j}\eta_{i',j}) + \tau_{i',i}\}\right\} \quad (1)$$

Here, $\tau_{i',i}$ denotes the communication time between processors $i$ and $i'$. (We assume this is symmetric and independent of $j$.) This completes a description of a generic phase in the most general setting.

*Performance Measure.* Given this description of a phase, a natural measure of performance is the amount of time taken by each processor to complete $n$ phases. More formally, given an $\varepsilon > 0$, to be thought of as very small, one would be interested in "eff"[2] which is defined to be the smallest number such that $\mathbf{Pr}\left[\frac{\max_{i=1}^{n}\{T_{i,n}\}}{n} \leq \text{eff}\right] \geq 1 - \varepsilon$. Thus, with probability at-least $1 - \varepsilon$, each processor finishes $n$ phases in time $n \cdot \text{eff}$.

### 2.2 Simplifying Assumptions

Now we present some simplifications, which make the model amenable for theoretical analysis and, yet, not render it unrealistic. The justifications for these assumptions are presented in detail in the paper [9], and we will only discuss them here very briefly. Several assumptions have been verified for real systems in [10]. Of course, one can make the model more and more *real* by removing some of these assumptions, but then the theoretical analysis of the model also becomes considerably difficult.
*(1) Identical Communication.* $G_j = G$ are the same for all $j \geq 1$. Also, as in [9], we assume that each message transmission between a pair of processors takes time $\tau$, which is referred to as the *one-way latency.*
*(2) Balanced Load.* $W_{i,j} = W$ for all $i, j$. This means that each thread is supposed to do the same amount of work in its compute stage. For instance, each thread is supposed to be multiplying two matrices of the same size.
*(3) Identical Processors.* $w_{i,j} = w$ for all $i, j$. This means that all the processors are identical in their computational power. Hence, in the noiseless case, given that $W_{i,j} = W$, $w_{i,j} = w$ for all processors.
*(4) Stationary and Balanced Overheads.* $f_{ij} = f$ for all $i, j$. In a typical systems, the processors are assigned an application for the lifetime of the application and running any other application on the node is avoided. Thus, the only interference

[2] This quantity, to be thought of as the efficiency of the system with respect to the application, will depend on the application and $\varepsilon$ .

is due to the background processes or daemons. The amount of daemon activity is not expected to change over time. Thus, we may assume $f_{i,j} = f_{i,j'}$, for all $i, j$ and $j'$. We further assume that $f_{i,j} = f_{i',j}$, for all $i, i'$ and $j$. (See [9] for more on this assumption.)
*(5) Identical Noise.* $\eta_{i,j} \sim \eta$ for all $i, j$. Recall that we have arranged $\eta$ and $f$ such that $\mathbf{E}[\eta] = 1$.
*(6) Spatial Independence.* $\{\eta_{i,j} : 1 \leq i \leq N\}$ are independent for each $j$. This assumption is crucial to our results. This can be justified as, in a typical system under consideration, there is no coordinated scheduling policy to synchronize processes across different processors.
*(7) t-Temporal Independence.* For the simplest of our results, we will assume that the the random variables $\{\eta_{i,j}\}$, are independent, i.e., apart from spatial independence, there is temporal independence as well. This may not be necessarily true as some of the daemons could be somewhat periodic, and we do expect weak correlation patterns between these random variables across different phases. In general we may only assume limited independence. To this effect, we say that the process is $t$ temporal independent, if for all $1 \leq i \leq N$ and $j \geq 1$, the set of random variables $\{\eta_{i,j'} : j \leq j' \leq j + t\}$ are independent. Typically, we will assume that $t \ll N$.

## 3 Analysis

### 3.1 The Simplified Problem

In this section we present the problem at hand with the simplifications made in the previous section. Applying assumptions (1)-(5) to Equation (1), we obtain that for $j \geq 0$,

$$T_{i,j+1} := \max\left\{T_{i,j} + w(1 + f\eta_{i,j}), \max_{i' \in N_{j+1}(i)}\{T_{i',j} + w(1 + f\eta_{i',j}) + \tau\}\right\}, \quad (2)$$

where $T_{i,0} = 0$ for all $1 \leq i \leq N$. The communication graph $G(V, E)$, and the parameters $w, f$ and $\tau$ are fixed for the rest of the paper. This graph contains loop edges of the form $(i, i)$ for all $1 \leq i \leq N$. The graph does not contain multiple edges in the same direction between a pair of vertices. Given $\varepsilon > 0$, recall that the goal is to give tight estimate of the quantity "eff" such that $\mathbf{Pr}\left[\frac{\max_{i=1}^{n}\{T_{i,n}\}}{n} \leq \text{eff}\right] \geq 1 - \varepsilon$. Since $w, f$ and $G$ are fixed, this quantity is just a function of $n, \varepsilon$ and $\eta$.

### 3.2 General Results

We proceed to give general bounds on this quantity as a function of the random variable $\eta$. First we need a few definitions.

**Definition 1** *A walk $\mathcal{W}$ of length $n$ in a directed graph $G(V, E)$ consists of a sequence of (possibly repeated) edges $e_1, e_2, \ldots, e_n$ such that, if $e_k = (i_k, j_k)$,* *then for all* $1 \leq k < n$, $j_k = i_{k+1}$. *The starting vertex of a walk is* $i_1$ *while the ending vertex is* $j_n$. *With abuse of notation, a walk* $\mathcal{W}$ *will be denoted by* $i_1, i_2, \ldots, i_n, i_{n+1}$, *where* $i_{n+1} := j_n$.

The maximum in-degree of $G$ is denoted by $\Delta$ (this includes the self-loop at each vertex). $G$ is said to be $\Delta$ regular if all vertices have in-degree $\Delta$. Let $\mathcal{W}_{i,n}$ denote the number of walks in $G$ of length $n$ that end at $i$, or $\mathcal{W} = i_1, \ldots, i_{n+1}$ such that $i_{n+1} = i$. It follows from the definitions that $|\mathcal{W}_{i,n}| \leq \Delta^n$, and if $G$ is $\Delta$ regular, then this is an equality.

**Definition 2** *For a vertex* $i$ *in* $G$, *let* $B(i,d)$ *denote the ball of radius* $d$ *centered at* $i$. *Formally,* $B(i,d)$ *contains all vertices* $i'$ *such that there is a directed path of length at-most* $d$ *from* $i'$ *to* $i$. *For* $G(V,E)$, *let* $B_G(d)$ *denote* $\max_{i \in V} |B(i,d)|$.

**Definition 3** *Let* $\eta$ *be a random variable such that* $\mathbf{E}[\eta] = 1$, *and let* $\sigma > 0$ *and* $n$ *be given. Let* $\{\eta_i\}_{i=1}^n$ *be* $n$ *independent copies of* $\eta$. *Then* $\eta$ *is said to be* $p(\eta, n, \sigma)$*-tailed if* $\mathbf{Pr}\left[\sum_{i=1}^n \eta_i \geq \sigma n\right] \leq p(\eta, n, \sigma)$.

Let $M_\eta^d$ denote the the random variable which is distributed according to the maximum of $d$ independent copies of $\eta$.

*Temporal Independence.*Assuming that the random variables $\eta_{i,j}$ are independent for all $i$ and $j$, we obtain the following results. The first follows from Equation (2) via a direct application of union bound.

**Theorem 4 (Upper Bound).** *For* $\sigma > 0$ *and* $n \geq 1$, *let* $\eta$ *be* $p(\eta, n, \sigma)$*-tailed, and the maximum in-degree of a vertex in* $G$ *be* $\Delta$. *Then, with probability at-least* $1 - p(\eta, n, \sigma)\Delta^n$, $\frac{\max_{i=1}^N \{T_{i,n}\}}{n} \leq wf\sigma + w + \tau$.

This theorem says that if for some $\sigma$, $p(\eta, n, \sigma)$ goes to zero faster than $1/\Delta^n$, then the per phase efficiency is at-most $wf\sigma + w + \tau$ with high probability. Indeed, for light tailed distributions, such as exponential distribution, this is true for $\sigma = \ln \Delta$. In fact for such distributions, one can show that this is the best we can hope for. We present a lower bound technique which, when applied to the exponential distribution shows that each phase will take time $wf\sigma + w + \tau$ on an average. (See Theorem 7.)

Now we present a general lower bound which is more convenient to state for regular graphs. This can be generalized to the case when the graph is not regular, but we omit it here.

**Theorem 5 (Lower Bound).** *Let* $G$ *be a* $\Delta$ *regular graph,* $n \geq 1$ *and for* $1 \leq j \leq N$, *let* $M_j$ *be i.i.d.* $M_\eta^\Delta$ *random variable. Then, for all* $1 \leq i \leq N$, $T_{i,n} \geq wf \sum_{j=1}^n M_j + wn$.

It is possible to incorporate the dependency of the lower bound on $\tau$ via a slightly more involved argument. We omit the easy proofs of these theorems from this version of the paper and focus on what they imply in Section 4.

*Limited Temporal Independence.* Consider now the case when the noise random variables $\eta_{i,j}$ $t$-temporally independent. In this case, Theorem 4 can no longer be expected to hold. Here we provide an upper bound by a *stochastic embedding* technique which takes into account the topology of the communication graph. The basic idea is to consider $t$ phases at a time, which we refer to as a meta-phase. If the graph $G$ has the property that a large delay at a node does not end up affecting processors further than distance $t$ from it, then the meta phase ends much faster. Imagine the following stochastic process. Every processor draws $t$ samples from its noise distribution. Because of the spatial independence and $t$-temporal independence, all these samples are i.i.d according to $\eta$. We denote these noise distributions as $\eta_{i,j}$. The meta-phase ends when all processors have finished $t$ phases. Consider a node $i$, and recall that $B(i,r)$ denotes the set of processors from which $i$ is reachable by a path of length at-most $r$. Recall also that $B_G(r)$ denotes the size of the largest ball of radius $r$ in $G$. Let $\zeta_j := \max_{i' \in B(i,r)} \eta_{i',j}$, where $\eta_{i',j}$ are i.i.d. according to $\eta$. Let $p(s) = \mathbf{Pr}[\sum_{j=1}^{t} \zeta_j \geq s]$. Hence, $\mathbf{Pr}[T_{i,t} \geq wfs + wt + t\tau] \leq p(s)$. Thus, using a union bound, one obtains the following theorem.

**Theorem 6 (Limited Independence Upper Bound).** *Let $\sigma \geq 0$ and $t \geq 1, r \geq 0$ be integers. Let $\eta$ be the distribution of noise which is $t$-temporally independent. Further, let $Y_1, \ldots, Y_t$ be i.i.d. according to $M_\eta^{B_G(r)}$, and $Y := \sum_{i=1}^{t} Y_i$. Then $\mathbf{Pr}\left[\frac{\max_{i=1}^{N}\{T_{i,t}\}}{t} \leq wf\sigma + w + \tau\right] \geq 1 - N\mathbf{Pr}[Y \geq \sigma t]$.*

This theorem says that even in the case of limited temporal independence, as long as the noise has the property that the sum of a small number of them have a light tail, one can still upper bound the per phase time by something much better than what one would expect in the global collective case. Of course, this requires that the communication graph is not expanding in the sense that the number of neighbors in a radius $r$ grow slowly as a function of $r$ for all the processors. This is indeed true for $d$-dimensional grids and toruses for fixed $d$.

## 4 Results for Representative Distributions

In this section we explain Theorems 4, 5 and 6 for the context of light tailed and heavy tailed distributions for noise. We pick the canonical examples of these two cases: the exponential and the Pareto respectively.

### 4.1 Distributions

*Exponential.* An exponential distribution Exp(1) has the following distribution: $\forall x \geq 0,\ \mathbf{Pr}[\text{Exp}(1) \leq \text{x}] = 1 - \exp(-\text{x})$. First we note some important properties of this distribution. Let $X_1, \ldots, X_n$ be i.i.d. according to Exp(1), then

1. $Y := \sum_{i=1}^{n} X_i$ is distributed according to $\Gamma(n,1)$ which has the following p.d.f. $f(x;n,1) := x^{n-1}\frac{\exp(-1)}{(n-1)!}$, for $x > 0$. The moment generating function is $(1-t)^{-n}$ for $t < 1$. It follows from Chernoff Bounds that for all $\Delta > 1$, for all $0 < \delta \leq \delta_0(\Delta)$, and all $n \geq n_0(\delta)$, $\mathbf{Pr}[Y \geq (1+\delta)n \ln \Delta] \leq \exp(-(1+\delta/2)n \ln \Delta)$.

2. Let $Y := \max_{i=1}^{\Delta} X_i$. Then $Y$ is distributed according to the random variable $\sum_{i=1}^{\Delta} \frac{1}{i} X_i$. (Lemma 1 below). It follows that $\mathbf{E}[Y] = \mathrm{Var[Y]} = \mathrm{H}_\Delta := \sum_{\mathrm{i}=1}^{\Delta} \frac{1}{\mathrm{i}}$. Hence, it follows from Chebyshev's Inequality that if $Y_1, \ldots, Y_n$ are distributed according to $Y$, are pairwise independent, then for any $0 < \delta < 1$, $\mathbf{Pr}[\sum_{i=1}^{n} Y_i \geq (1-\delta) n H_\Delta] \geq 1 - \frac{1}{\delta^2 n H_\Delta}$. It follows from the inequality that $\frac{1}{2\Delta+2} \leq |H_\Delta - \ln \Delta - \gamma| \leq \frac{1}{2\Delta}$ (where $\gamma > 0$ is the Euler-Mascheroni constant) that, for all $0 < \delta \leq \delta_0(\Delta)$, and all $n \geq n_0(\delta)$, $\mathbf{Pr}[\sum_{i=1}^{n} Y_i \geq (1-\delta/2) n \ln \Delta] \geq 1 - \frac{1}{\delta^2 n H_\Delta}$.

**Lemma 1.** *Let $X_1, \ldots, X_k$ be i.i.d. according to* $\mathrm{Exp}(1)$*. Then $\max_{i=1}^{k} X_i$ has the same distribution as $\sum_{i=1}^{k} \frac{X_i}{i}$.*

*Spatial and Temporal Independence.* The fact (1) above implies that for all $\delta > 0$ small enough, when $\eta \sim \mathrm{Exp}(1)$, $p(\eta, n, (1+\delta)\ln \Delta) \leq \frac{1}{\Delta^{(1+\delta/2)n}}$. Hence, by Theorem 4, with probability at-least $1 - \Delta^{-n\delta/2}$, $\frac{\max_{i=1}^{N}\{T_{i,n}\}}{n} \leq wf(1+\delta)\ln \Delta + w + \tau$. While the fact (2) above combined with Theorem 5 implies that for all $\delta > 0$ small enough, with high probability, every processor finishes $n$ phases in time at-least $(1-\delta/2) n \ln \Delta$. These together imply the following theorem.

**Theorem 7.** *For all $\delta > 0$ small enough, for $\eta \sim$* $\mathrm{Exp}(1)$*, and when the communication topology is given by a (regular) digraph $G$ with in-degree $\Delta$, with high probability, the efficiency or the maximum average time per phase for each processor lies between $wf(1-\delta/2)\ln \Delta + w + \tau$ and $wf(1+\delta)\ln \Delta + w + \tau$.*

Thus, for standard communication topologies such as toruses or meshes, this result is optimal.

*Limited Temporal Independence.* Now we show that assuming $O(\ln N)$-temporal independence, and the fact that for $r = O(\ln N)$, the communication graph is not expanding, i.e., $B_G(O(\ln N)) = O((\ln N)^{O(1)})$, $\ln N$ phases will finish in time $O(\ln N \ln\ln N)$. Thus giving an efficiency of $O(\ln\ln N)$ per phase with high probability. Compare this to the case when each phase takes $\Theta(\ln N)$ time in the case of global collectives [9]. Formally, we have the following theorem.

**Theorem 8.** *Let $c_1, c_2 > 0$ be large constants. Let the communication graph $G$ have the property that for all $r \leq c_1 \ln N$, $B_G(r) \leq (\ln N)^{c_2}$. If the noise is distributed according to* $\mathrm{Exp}(1)$ *and is $\ln N$-temporally independent, then with probability at-least $1 - 1/N^{100}$ each processor finishes $\ln N$ phases in time at-most $100 \ln N(wf \ln\ln N + w + \tau)$. Thus, the average time per phase for each processor is at-most $100wf \ln\ln N + w + \tau$ with high probability.*

The proof of this theorem relies on Theorem 6 and the measure concentration inequality for the random variable distributed according to the maximum of exponential distributions and we will include it in the full version of this paper. This is a significant speed-up compared to $\ln N$ per phase and assumes that noise is temporally independent for only about $\ln N$ phases. It would be interesting to see if this is indeed observed for real systems.

*Pareto.* In this section we consider the case when the noise has a heavy tail. This is unlike the exponential case and the noise looks more like the uniform distribution. A natural and very popular way to model data which has heavy tail is the so-called Pareto distribution. The Pareto random variable $X_{\text{par}}^a$ with parameter $a$ has the following distribution: $\forall x \geq 1, \quad \mathbf{Pr}[X_{\text{par}}^a \leq x] = 1 - \frac{1}{x^a}$. The Pareto distribution has mean $\frac{a}{a-1}$. To make this random variable with unit mean, we let $\eta$ be $\frac{a-1}{a} X_{\text{par}}^a$. The reason why when the noise is distributed according to Pareto the system will invariably slow down is very simple. After $t \leq N$ phases, a processor $i$ starts depending on processors which are connected to it by a directed path of length $t$. Thus, the number of independent copies of $\eta$ after $t$ phases on which $i$-th processor depends is $P(i,t) := \sum_{r=0}^{t} |B_G(i,r)|$. It follows from the distribution of the maximum of Pareto distribution that, with high probability, there is one of these which will be at-least $\Omega(P(i,t)^{1/a})$. Hence, the $i$-th processor will take time at-least $\Omega(\frac{1}{t} P(i,t)^{1/a})$ on an average per phase. If $P(i,t) = \Omega(t^{\beta})$ for some $\beta > a$, then this quantity is at-least $\Omega(t^{\beta/a-1})$. When $G$ is a ring, as noted in [11], $\beta = 2$, and hence, for $1 < a < 2$, the slowdown is $\Omega(N^{2/a-1})$. We do not discuss it further here as the main idea already appears in the paper of Lipman and Stout [11].

## 5 Multiple Communications per Phase

In this section we consider two cases of a complex communication pattern per phase. The first is the complete binary tree and the second is Fast Fourier Transform (FFT). Both are very natural. The binary tree will arise in any divide and conquer type of application, e.g. Merge Sort, while FFT is a standard benchmark for HPCC. For the sake of clarity we would consider the case when the communication delays are negligible. This is just to highlight the impact of noise, and all results can be suitably modified to incorporate the communication component.

### 5.1 Binary Tree

Consider the case when $N = 2^k$, where the processors are labeled $\{0, 1, \ldots, 2^k - 1\}$. Each phase consists of $k$ rounds. In the $i$-th round ($1 \leq i \leq k$), processors $j$ and $j + 2^{i-1}$ communicate for $j = 0 \cdot 2^i, 1 \cdot 2^i, 2 \cdot 2^i, \ldots, \left\lfloor \frac{2^k - 1}{2^i} \right\rfloor \cdot 2^i$. An example of such a communication pattern is given in Figure 1. We assume that the one way latency $\tau \sim 0$, and hence, only focus on the delay due to synchronization. Let $\eta_{i,j}$ be the random variable denoting the noise incurred by processor $i$ in the $j$-th round. Assume $\{\eta\}_{i,j}$ are i.i.d. according to Exp(1), let $w$ be the work by each processor done per round, and the overhead factor per round is $f$. Thus, time taken to complete one phase is the random variable $wk + wf \max_P \sum_{(i;j) \in P} \eta_{i,j}$. Here the maximization is over all paths that go from a *leaf* to the *root* of the binary tree. There is a path corresponding to each leaf, which is just a processor, and there are exactly $2^k$ of them. Hence, the quantity that we need to estimate is $B_k := \max_P \sum_{(i;j) \in P} \eta_{i,j}$. We prove the following theorem which establishes a remarkable threshold phenomena in the completion time of each phase.

**Theorem 9.** *Let $c_L = 2.678\ldots$ be a solution to the equation $\ln 2 + \ln x - x + 1 = 0$. Then $B_k/k \to c_L$ almost surely as $k \to \infty$. (Here $B_k$ is as defined above when $\eta_{i,j}$ are i.i.d. according to* $\text{Exp}(1)$.*) Thus, the time to complete each phase is almost surely $(1 + fc_L)w \log_2(N + 1)$.*

The upper bound proof involves a tight approximation to the distribution obtained by summing $k$ independent copies of the exponential random variable. The lower bound proof is technically interesting as it uses a result from the theory of branching processes on the behavior of a supercritical Galton-Watson process. It is worth noting that the lower bound argument given earlier for the general case (Theorem 5) does not give the optimal constant, and one needs to appeal to theory of branching processes to obtain the optimal constant. Also, a close look at the proof of Theorem 9 yields that a similar threshold result can be obtained when $\eta_{i,j}$ are i.i.d. according to any distribution for which there is a large deviation inequality.

### 5.2 Fast Fourier Transform

In this section we consider the communication pattern for an application computing the FFT. Here, each phase consists of $k$ rounds. In the $i$-th round, $1 \le i \le k$, processors $j$ and $2^{i-1} + j$ communicate with each other, where $0 \le j < 2^k$. The communication pattern for one processor is a binary tree, as depicted in Figure 1. A phase consists of a binary tree for each processor, except that these binary trees share edges. For instance, the binary tree for processes $j$ and $2^{k-1} + j$ are the same for all $j$. Let $H_k$ denote the time taken for a phase to complete when each processor does $w$ work per round, $f$ is the noise overhead, and the noise in each round is distributed according to Exp(1). Then we have the following theorem on $H_k$.

**Theorem 10.** *Let $c_L = 2.678\ldots$ be a solution to the equation $\ln 2 + \ln x - x + 1 = 0$, and $c_U = 3.692\ldots$ be a solution to $2\ln 2 + \ln x - x + 1 = 0$. Then the following hold: (1) $\limsup_{k\to\infty} H_k/k \le c_U$ almost surely. (2) $\liminf_{k\to\infty} H_k/k \ge c_L$ almost surely. Thus, the time to complete each phase is almost surely bounded between $(1 + fc_L)w \log_2(N + 1)$ and $(1 + fc_U)w \log_2(N + 1)$.*

This theorem establishes that in the case of FFT, inspite of the dependencies among the binary trees of the processors, each phase finishes in time $O(\log N)$.

*Remark 1.* We conjecture that $H_k$ also has threshold behavior as $B_k$ in Theorem 9. The current techniques do not seem powerful enough to resolve this question.

*Acknowledgment.* I would like to thank Rahul Garg for suggesting to look at this model involving local collectives and Daniel Stefankovic for collaborating on this paper at an early stage.

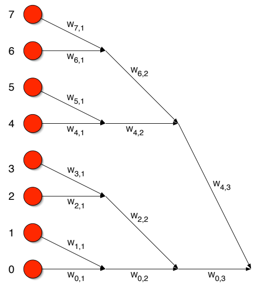

**Fig. 1.** The communication pattern for a binary tree and a single node in FFT

## References

1. R. Gioiosa, F. Petrini, K. Davis, and F. Lebaillif-Delamare, “Analysis of System Overhead on Parallel Computers,” in *The 4th IEEE International Symposium on Signal Processing and Information Technology (ISSPIT 2004)*, (Rome, Italy), Dec. 2004.
2. T. R. Jones, L. B. Brenner, and J. M. Fier, “Impacts of Operating Systems on the Scalibility of Parallel Applications,” Tech. Rep. UCRL-MI-202629, Lawrence Livermore National Laboratory, Mar. 2003.
3. F. Petrini, D. J. Kerbyson, and S. Pakin, “The Case of the Missing Supercomputer Performance: Achieving Optimal Performance on the 8,192 Processors of ASCI Q,” in *ACM/IEEE Conference on Supercomputing (SC’03)*, (Phoenix, Arizona, USA), Nov. 2003.
4. E. Frachtenberg, F. Petrini, J. Fernandez, S. Pakin, and S. Coll, “STORM: Lightning-Fast Resource Management,” in *ACM/IEEE Conference on Supercomputing (SC’02)*, (Baltimore, Maryland, USA), Nov. 2002.
5. A. Hori, H. Tezuka, and Y. Ishikawa, “Highly Efficient Gang Scheduling Implementation,” in *ACM/IEEE Conference on Supercomputing (SC’98)*, (Orlando, FL, USA), Nov. 1998.
6. T. Jones, S. Dawson, R. Neely, W. Tuel, L. Brenner, J. Fier, R. Blackmore, P. Caffrey, B. Maskell, P. Tomlinson, and M. Roberts, “Improving the Scalability of Parallel Jobs by adding Parallel Awareness to the Operating System,” in *ACM/IEEE Conference on Supercomputing (SC’03)*, (Phoenix, Arizona, USA), Nov. 2003.
7. E. Frachtenberg, D. Feitelson, F. Petrini, and J. Fernández, “Flexible Coscheduling: Mitigating Load Imbalance and Improving Utilization of Heterogeneous Resources,” in *International Parallel and Distributed Processing Symposium 2003 (IPDPS03)*, (Nice, France), Apr. 2003.
8. S. Agarwal, G. S. Choi, C. R. Das, A. B. Yoo, and S. Nagar, “Co-ordinated Coscheduling in Time-Sharing Clusters through a Generic Framework,” in *IEEE International Conference on Cluster Computing (CLUSTER’03)*, (Hong Kong), Dec. 2003.
9. S. Agarwal, R. Garg and N. Vishnoi, “The Impact of Noise on the Scaling of Collectives: A Theoretical Approach”, in *HiPC*, 2005.
10. R. Garg, P. De, “Impact of Noise on Scaling of Collectives: An Empirical Evaluation”, in *HiPC* , 2006.
11. J. Lipman, Q.F. Stout, “A performance analysis of local synchronization”, in *Symp. Parallelism in Algorithms and Architectures (SPAA)*, 2006.