On the Definition of Speedup

Wolfgang Ertel*

TR-93-069

December 1993

Abstract

We propose an alternative definition for the *speedup* of parallel algorithms. Let A be a sequential algorithm and B a parallel algorithm for solving the same problem. If A and/or B are randomized or if we are interested in their performance on a probability distribution of problem instances, the running times are described by random variables T^A and T^B. The speedup is usually defined as $E[T^A]/E[T^B]$ where E is the arithmetic mean. This notion of speedup delivers just a number, i.e. much information about the distribution is lost. For example, there is no variance of the speedup. To define a measure for possible fluctuations of the speedup, a new notion of speedup is required. The basic idea is to define speedup as $M(T^A/T^B)$ where the functional form of M has to be determined. Also, we argue that in many cases $M(T^A/T^B)$ is more informative than $E[T^A]/E[T^B]$ for a typical user of A and B. We present a set of intuitive axioms that any speedup function $M(T^A/T^B)$ must fulfill and prove that the geometric mean is the only solution. As a result, we now have a uniquely defined speedup function that will allow the user of an improved system to talk about the average performance improvement as well as about its possible variations.

Keywords: Speedup, variation of running time, geometric mean, functional equations, randomized algorithms.

---

*Current address: International Computer Science Institute, 1947 Center Street, Berkeley CA 94704, ertel@icsi.berkeley.edu. Research supported by an ICSI Postdoctoral Fellowship.

1 Introduction

Although recently various new performance measures for parallel algorithms and architectures have been proposed (see e.g. [SG91]), the *speedup* defined as the ratio of the running times of sequential and parallel execution still is one of the most common figures for the performance evaluation of parallel algorithms and architectures.

There are two obvious scenarios where the speed of an algorithm is described by a distribution of running times. In case of a determistic algorithm the distribution of running times may be taken over a sample of different problem instances (i.e. different inputs). In case of a randomized algorithm[1], even for a fixed problem instance, the running time may depend on the random seed and therefore is described by a probability distribution.

Let T_1 and T_p be the random variables for the running time on one and on p processors. The "speedup of the average running times" S_0 is usually defined as

$$S_0(k) := \frac{E(T_1)}{E(T_p)},$$

i.e. as the ratio of the expected values. This definition has been widely used for many years by empirical as well as theoretical scientists. Especially in the analysis of randomized algorithms for combinatorial problems the speedup is a measure of great interest. For example, there is an ongoing discussion about the reasons for superlinear speedup (see e.g. [MG85, RK88, Nat89, SMV88, Ert92, GK93]). In [GK93] and [Ert92] e.g. it was shown, that for certain combinatorial search algorithms there is a strong correlation between the variance of the probability distribution of running times and the occurrence of superlinear speedup.

If the variables T_1 and T_p both have only one value, say t_1 and t_p, then there is no doubt, that t_1/t_p is the right relative performance measure. If, however, T_1 and T_p have nontrivial distributions, then there are two problems with S_0. First, often the expectation of the relative performance T_1/T_p is of interest, which S_0 does not provide. Second, the speedup S_0 gives information about the relative average performance of the two systems, but carries no information about possible variations of the "speedup". In other words, since S_0 is just a number, there is no underlying distribution and consequently nothing can be said about the deviation of the ratio T_1/T_p from the value $S_0(p)$.

This leads to the more general question: What is the right speedup definition for any purpose? To answser this question we will introduce in Section 2 a new speedup definition and compare it with the standard definition. For two most common scenarios where a distribution of running times occurs we show in Section 3, how to compute the distribution of the ratios. In Section 4 we will prove that this definition

[1]A randomized algorithm receives in addition to its input a sequence of random bits, called the *seed*.

is uniquely determined, i.e. it is the unique solution of a set of obvious functional equations.

2 From "Speedup of the Average" to "Average Speedup"

First we will introduce a slightly more general notation which will be applicable to comparing the running times of any two systems A and B rather than just sequential and parallel algorithms. Let T^A and T^B be the random variables for the running times of A and B with the possible values $\alpha_1, \ldots, \alpha_n \in \mathbb{R}_+$[2] and $\beta_1, \ldots, \beta_m \in \mathbb{R}_+$. The respective probability densities will be denoted by p_i^A and p_i^B. Let

$$E[T^A] \equiv \sum_{i=1}^{n} \alpha_i \, p_i^A \qquad \text{and} \qquad E[T^B] \equiv \sum_{i=1}^{m} \beta_i \, p_i^B,$$

be the expected values of T^A and T^B. The *speedup of the average running times* is defined by

$$S_1(T^A, T^B) := \frac{E[T^A]}{E[T^B]}. \tag{1}$$

As already mentioned above, this definition has the disadvantage, that there is no way to define the "variation of the speedup". This leads to a short excursion about the purpose of the "speedup". Among others there are two different, typical scenarios:

1. The designer of an improved system B (e.g. a parallel algorithm) wants to compare its performance with the original system A. Since this system will be used very often, he may be interested in the reduction of cost it produces in the long run. Therefore, in this case the sum of the running times of system A has to be related to the sum of the running times of system B. Thus S_1 is perfectly appropriate for this purpose. Also, variations in the running times are not of great interest, since the relative variations of running times will cancel out, if the sample size is big enough. For these reasons we could also call S_1 the "designer speedup", as opposed to the "user speedup" which is motivated as follows.

2. The user of the two systems A and B may have a different view. He wants to know, if he runs B once, how much it is faster (or slower) than A. Thus, he wants to know the mean of the ratio T^A/T^B what we might call the "average speedup". But even if the user knows the mean ratio of T^A and T^B, he might ask how certain it is to observe this value. In other words, the user really should have some knowledge about the variation of the speedup.

[2]For the rest of this paper let $\mathbb{R}_+$ denote the positive real number without zero.

Let the "average speedup" S_2 be defined as the mean value M of the ratio of the two variables T^A and T^B. Thus, we define

$$S_2(T^A, T^B) = M\left(\frac{T^A}{T^B}\right) \tag{2}$$

where M can be any mean, in particular the arithmetic mean. In the following we will show that the geometric mean is the only useful instantiation of M. Before, however, we have to make clear what T^A/T^B means.

3 The Appropriate Sampling Strategy

Depending on the source of randomness in our statistical running time data T^A and T^B there are two obvious ways of defining the new random variable $X = T^A/T^B$.

3.1 Scenario 1: The Running Times T^A and T^B are Correlated via the Input

A typical scenario involves the comparison of two different deterministic algorithms (or architectures) A and B on a set of inputs (benchmarks). Here, the definition of X is obvious, since the only figure of interest is the ratio α_i/β_i of the running times of A and B on each of the inputs $I_1, \ldots, I_k$. Hence we have the same number of different running times for A and B. If we define k as the number of different ratios α_i/β_i, we have $k = n = m$. Although in this scenario in many cases it would be sufficient to have equal weights, i.e. $p_1^A = \ldots = p_k^A = p_1^B = \ldots = p_k^B = 1/k$, we will work with a probability distribution of inputs.

Therefore we define for $i = 1, \ldots, k$

$$x_i \equiv \alpha_i/\beta_i \tag{3}$$

and

$$p_i \equiv Pr(X = x_i) = Pr\left(\frac{T^A}{T^B} = x_i\right) = \sum_{j=1}^{n} \delta\left(\frac{\alpha_j}{\beta_j}, x_i\right) \cdot p_i^A \tag{4}$$

Motivated by the task of comparing different RISC processors on a set of benchmark problems, this scenario has been investigated in [FW86] and it was proved that the geometric mean is the only "reasonable" figure to summarize the relative performance on all the benchmarks in one number. Unfortunately however, the uniqueness proof in [FW86] is only valid for this scenario and not for the case of two uncorrelated random variables T^A and T^B, which me must consider e.g. when A and/or B are randomized algorithms.

3.2 Scenario 2: The Running Times T^A and T^B are Uncorrelated

Now let us assume the two random variables T^A and T^B are uncorrelated, both having their own distribution. This happens for example if A and B are randomized algorithms and if the experiments are performed such that the seeds for any two runs of A and B are independently chosen at random. Here we have no indication which particular values α_i/β_j we must select to compute the distribution of X. Therefore we have to work with all possible ratios and define the set $\mathcal{X}$ of possible values for the quotient variable $X = T^A/T^B$ by

$$\mathcal{X} \equiv \{\alpha_i/\beta_j \in I\!R_+ : i \in \{1,\ldots,n\}, j \in \{1,\ldots,m\}\} \tag{5}$$

with cardinality $k \equiv |\mathcal{X}| \leq n\,m$.[3] For any ratio $x_i \in \mathcal{X}$ the probability to observe x_i is

$$\begin{aligned} p_i &\equiv Pr(X = x_i) = Pr\left(\frac{T^A}{T^B} = x_i\right) \\ &= \sum_{j=1}^{n}\sum_{l=1}^{m} \delta\left(\frac{\alpha_j}{\beta_l}, x_i\right) \cdot p_j^A \cdot p_l^B \quad \text{where} \quad \delta(x,y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

3.3 Other Scenarios

These two scenarios describe just two out of many possibilities for computing the quotient variable X. Of course there are many other situations, where the computation of X is different. For example one can imagine a deterministic algorithm A and a randomized algorithm B and vice versa. Also, in case of randomized algorithms one might sample over different inputs. However, in all these cases the computation of X is obvious.

Now, that we know how to compute the distribution of X, we get back to the question of finding the appropriate mean M.

4 The Right Mean

The first choice for M usually is the arithmetic mean, i.e. $M(X) = \sum_i x_i\,p_i$. However, as the following example shows, the arithmetic mean does not behave as one would expect a speedup function to do. Suppose we have two identical randomized algorithms A and B and for either algorithm we measure the same running times 1 and 10 with probability 1/2. Thus, if we use the method of Scenario 2 to compute X, we get the values 1, 1/10, 10, 10/10 and the following distribution:

x_i	1	$\frac{1}{10}$	10
p_i	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$

[3] Note that $|\mathcal{X}|$ may be less than $n\,m$ due to multiple occurrences of certain ratios α_i/β_j.

The arithmetic mean of this distribution is 3.025. Since both algorithms have the same performance, we get $T^A = T^B$ and one would expect $S(T^A, T^A) = M(T^A/T^A) = 1$. As can easily be checked, the geometric mean of these four numbers is 1. The reason for this is that the inverse ratios 1/10 and 10 cancel out when they are multiplied rather than added. In [FW86] a number of examples obtained from measurements of the relative speed of RICS processors are used to point out the problems which arise if the arithmetic mean is used to summarize relative scores. From these examples one can derive a set of obvious requirements on M which are listed below.

First, however, for any $k \geq 2$ we introduce the notation

$$M(\vec{x};\vec{p}) \equiv M(x_1,\ldots,x_k;p_1,\ldots,p_k) \equiv M(X)$$

for the mean value of the random variable X with the possible values $x_1,\ldots,x_k$ and probabilities $p_1,\ldots,p_k$. In case of equal probabilities $p_1 = p_2 = \ldots = p_k = 1/k$ we define a new k-ary function

$$F_k(x_1,\ldots,x_k) \equiv M(x_1,\ldots,x_k;1/k,\ldots,1/k).$$

In the remainder of this section we will restrict ourselves to this case of equal probabilities, since this will make the proofs much easier and clearer. This is no real restriction, since for any k a $2k$-ary function M^* can be defined in terms of a function F_r, where $r \geq k$, as follows. If we have rational probabilities

$$p_1 = \frac{d_1}{n_1},\ldots,p_k = \frac{d_k}{n_k} \qquad (d_i \in I\!N,\ n_i \in I\!N).$$

and define η as the least common denominator of $p_1,\ldots,p_k$, there is for all i a unique δ_i with $d_i/n_i = \delta_i/\eta$. We get $\delta_1 + \ldots + \delta_k = \eta$ and

$$M(x_1,\ldots,x_k;\frac{d_1}{n_1},\ldots,\frac{d_k}{n_k}) = M(x_1,\ldots,x_k;\frac{\delta_1}{\eta},\ldots,\frac{\delta_k}{\eta}).$$

This allows us to define

$$M^*(x_1,\ldots,x_k;\frac{\delta_1}{\eta},\ldots,\frac{\delta_k}{\eta}) \equiv F_\eta(\underbrace{x_1,\ldots,x_1}_{\delta_1\text{times}},\underbrace{x_2,\ldots,x_2}_{\delta_2\text{times}},\ldots,\underbrace{x_k,\ldots,x_k}_{\delta_k\text{times}}) \qquad (6)$$

This definition represents the interpretation of the probabilities as normalized frequencies, or in other words, the weight of the value x_i must be proportional to its probability.[4] If the p_i are real numbers, this definition makes no sense. However, every real number p_i is the limit of a sequence $(q_{ik})_{k\in I\!N}$ of rational numbers. If we require M to be continuous in the last k arguments, we can define

$$M(x_1,\ldots,x_k;p_1,\ldots,p_k) \equiv \lim_{n\to\infty} M^*(x_1,\ldots,x_k;q_{1n},\ldots,q_{kn}) \qquad (7)$$

[4]Since below we will require F to be symmetric in its arguments, the weight of x_i is naturally represented by the number of occurrences of x_i.

If we assume for a moment that F_k is the geometric mean, i.e. $F_k = \sqrt[k]{x_1 \cdot \ldots \cdot x_k}$, equation (6) and (7) rewrite to

$$M^*(x_1, \ldots, x_k; \frac{\delta_1}{\eta}, \ldots, \frac{\delta_k}{\eta}) = \left(x_1{}^{\delta_1}\, x_2{}^{\delta_2} \ldots x_k{}^{\delta_k}\right)^{\frac{1}{\eta}} = \prod_{i=1}^{k} x_i{}^{(\delta_i/\eta)}$$

and

$$M(\vec{x}, \vec{p}) = G(\vec{x}, \vec{p}) \equiv \prod_{i=1}^{k} x_i{}^{p_i} \tag{8}$$

as one would expect. Since we know (see below) that the geometric mean is the only solution for F_k, the geometric mean is also the only solution for M.

In the following we will keep the arity k of F_k fixed (but arbitrary), so we can omit the index k, i.e. we define $F \equiv F_k$.

4.1 A Uniqueness Proof for Scenario 1

Now we are ready to give a first characterisation of F for the case of correlated running times T^A and T^B, as described in 3.1. If we define a second random variable Y with values $y_1, \ldots, y_k$, our requirements are

A1 (Reflexivity): $F(x, \ldots, x) = x$

A2 (Multiplicativity): $F(x_1, \ldots, x_k) \cdot F(y_1, \ldots, y_k) = F(x_1 y_1, \ldots, x_k y_k)$

A3 (Symmetry): For any vector $\vec{x}$ of length k and any permutation π of $\{1, \ldots, k\}$

$$F(x_1, \ldots, x_k) = F(x_{\pi(1)}, \ldots, x_{\pi(k)})$$

A1 is obvious. To understand A2, imagine a scenario, where a deterministic algorithm A is being improved, resulting in C, which again is improved, resulting in B. Together with (2) this implies that the overall speedup of B relative to A is the product of the two single speedups, i.e.

$$S(T^A, T^B) = S(T^A, T^C) \cdot S(T^C, T^B).$$

In the context of randomized algorithms (see 3.2) A2 is not applicable, since in general the sample size of X and Y need not be the same. This shows that $A2$ is a very strong (restrictive) axiom which will make the proof of Lemma 1 very easy. A formal verifikation of the strength of A2 can be found in [Rob90], Theorem 1. One might think of generalisations of $A2$; however as we will see below there is a very natural, but weaker axiom A2', which is still sufficient to characterize the geometric mean. Requirement A3 ensures that for fixed $x_1, \ldots, x_k$ the order of the values x_i does not affect the resulting speedup.

We will first use the relatively strong set of axioms A1...A3 to characterize the geometric mean. In [FW86] and [Acz66] a similar proof has been given for the same set of axioms.

Lemma 1 *The only function $F : \mathbb{R}_+^k \to \mathbb{R}_+$ that fulfills requirements A1,A2 and A3 is the geometric mean*

$$F(x_1, \ldots, x_k) = \sqrt[k]{x_1 \cdot \ldots \cdot x_k}.$$

Proof: It is easy to see that the geometric mean fulfills A1,...,A3. The following sequence of equations holds for any solution f of A1,...,A3:

$$\begin{aligned} f(x_1, \ldots, x_k)^k &= f(x_1, \ldots, x_k) \cdot f(x_2, \ldots, x_k, x_1) \cdot \ldots \cdot f(x_k, x_1, \ldots, x_{k-1}) \\ &= f(x_1 \cdot \ldots \cdot x_k, \ldots, x_1 \cdot \ldots \cdot x_k) \\ &= x_1 \cdot \ldots \cdot x_k. \end{aligned}$$

For the first equation we used A3 to write $f(x_1, \ldots, x_k)^k$ as the product of f applied to all cyclic permutations of $x_1, \ldots, x_k$. The second equation follows from $k-1$ fold application of A2 and the last line is an application of the reflexivity axiom A1. The lemma follows immediately, since f must be a positive real valued function. □

4.2 A General Uniqueness Proof

We will now give a second set of axioms that characterize the geometric mean. Although there are many more axioms, they are all obvious for characterizing a speedup function regardless which of the two scenarios is being considered.

B1 (Continuity): F is continuous in all its variables.

B2 (Strict Monotony): for $1 \le i \le k$, if $x_i < x_i'$ then
$$F(x_1, \ldots, x_i, \ldots, x_k) < F(x_1, \ldots, x_i', \ldots, x_k)$$

B3 (Bisymmetry): the symmetry of the following function in its k^2 variables,
$$g(x_{11}, \ldots, x_{1n}, \ldots, x_{n1}, \ldots, x_{nn}) \equiv F(F(x_{11}, \ldots, x_{1n}), \ldots, F(x_{n1}, \ldots, x_{nn}))$$

B4 (Homogeneity): For any $c \in \mathbb{R}_+$: $F(c\,x_1, \ldots, c\,x_k) = c\,F(x_1, \ldots, x_k)$

B5 (Reciprocal Property): $$F\left(\frac{1}{x_1}, \ldots, \frac{1}{x_k}\right) = \frac{1}{F(x_1, \ldots, x_k)}$$

B1 and B2 are obvious. B3 basically says that if we split up our data into equal parts and then compute the mean of the means of the parts, the result must not depend on how the data are partitioned. The meaning of B4 in terms of the running times α_i and β_i is: if all the ratios α_i/β_i are increased (decreased) by a constant factor c, then the speedup must increase (decrease) by the same factor c. In particular, multiplying all α_i by c has the same effect as multiplying all β_i by $1/c$. B5 ensures that if we exchange the two algorithms A and B, the speedup is being inverted. This property is central for the proof of Theorem 2, since it excludes the arithmetic mean from the set of solutions of B1–B4.

Any function $Q_\phi : I\!\!R_+^k \rightarrow I\!\!R_+$ is called a *quasiarithmetic mean* if and only if there exists a strictly monotonous function ϕ such that

$$Q_\phi(x_i, \ldots, x_k) = \phi^{-1}\left(\frac{1}{k}\sum_{i=1}^{k}\phi(x_i)\right) \tag{9}$$

For the case of unequal probabilities we define the *quasilinear mean*

$$Ql_\phi(x_i, \ldots, x_k, p_i, \ldots, p_k) = \phi^{-1}\left(\sum_{i=1}^{k} p_i\,\phi(x_i)\right)$$

which we get, if we substitute (9) into (6) and (7). This is a very general notion of *mean*. For $\phi(x) \equiv x$ we get the arithmetic mean, for $\phi(x) \equiv \log(x)$ the geometric mean, and for $\phi(x) \equiv 1/x$ the harmonic mean.

Theorem 2 *The only solution of B1–B5 is the geometric mean*

$$F(x_1, \ldots, x_k) = \sqrt[k]{x_1 \cdot \ldots \cdot x_k}.$$

The proof is based on two lemmas. We will first use B1–B5 to characterise the class of quasiarithmetic means and then show that the only quasiarithmetic mean that fulfills our axioms B4 and B5 is the geometric mean.

It is easy to see that the homogeneity B4 implies the reflexivity A1. As an immediate consequence of B3 and A1 we get the symmetry A2. Thus, in the following we can use A1 and A2 to prove Theorem 2. To characterize the quasiarithmetic mean we use the following lemma which was proved in [Acz46].

Lemma 3 *A Function $F : I\!\!R_+^k \rightarrow I\!\!R_+$ is solution of A1, A2, B1, B2 and B3 if and only if it is a quasiarithmetic mean.*

There are other characterisations of the quasiarithmetic mean, (see e.g.[AA86], or [AS83], which replace B3 by different properties, but at least in the present context, they are much less natural than B3. The next lemma completes the proof of Theorem 2.

Lemma 4 *The only homogeneous quasiarithmetic mean that satisfies B5 is the geometric mean.*

A proof for this lemma can be found in [AA86] and [AS83]).

5 Variations of the Speedup

The variance of a random variable X is defined as the second moment of $X - E[X]$, where the expected value $E[X]$ is defined as the first moment of X. As our new speedup definition $S_2(T^A, T^B) = G(\vec{x}, \vec{p})$ uses the geometric mean rather than the first moment, the second moment is no adequate means for measuring the variation of the speedup. Therefore we have to find a different figure to measure deviations from the geometric mean.

Since the geometric mean is a quasilinear mean, and any quasilinear mean $Ql_\phi(X)$ is a function of the expected value of $\phi(X)$, we can use the second moment of $\phi(X) - E[\phi(X)]$ to define the *quasi variance*

$$V_\phi(X) \equiv \phi^{-1}(\, E[\,(\phi(X) - E[\phi(X)])^2\,]\,) = \phi^{-1}(\, E[f^2(X)] - (E[\phi(X)])^2]\,) \qquad (10)$$

and the *quasi standard deviation*

$$D_\phi(X) \equiv \phi^{-1}\left(\sqrt{E[f^2(X)] - (E[\phi(X)])^2}\right).$$

For $\phi(x) \equiv \log(x)$ we get

$$D_{\log}(X) = \exp\left(\sqrt{\sum_{i=1}^{k} p_i\, \log^2(x_i) - \left(\sum_{i=1}^{k} p_i\, \log(x_i)\right)^2}\right). \qquad (11)$$

This definition is a natural measure for the deviation of X from the geometric mean. For example, this function has the desirable property that for $c > 0$ the values $c\,G(\vec{x}, \vec{q})$ and $\frac{1}{c}\,G(\vec{x}, \vec{q})$ have the same weight in the computation of $D_{\log}(X)$.

Of course, there are other measures of variation, which are independent from the particular mean that is being used, e.g. for certain applications the use of quantiles may be the right choice.

6 Continuous Variables

If the variables T^A and T^B are continuous, e.g. real valued, then the definition of S_2 in (2) and G in (8) must be replaced by

$$S_2^c(T^A, T^B) \equiv G^c(\vec{x}, \vec{p}) \equiv \exp\left(\int_{\mathcal{X}} p(x)\, \log(x)\, dx\right), \qquad (12)$$

where $\mathcal{X}$ depends on the sampling strategy. For example the continuous variant of (5) is

$$\mathcal{X} \equiv \{\alpha/\beta \in I\!\!R : \alpha \in I\!\!R,\ \beta \in I\!\!R\}$$

The definition (12) can easily be motivated by the above mentioned fact that $Ql_{\log}(X) = G(X)$, i.e.

$$S_2(T^A, T^B) \equiv \prod_{i=1}^{k} x_i^{p_i} = \exp\left(\sum_{i=1}^{k} p_i \log(x_i)\right).$$

Replacing summation by integration yields (12). Note that the integral over $\mathcal{X}$ in fact is a two-dimensional integral. It is easy to see that S_2^c fulfills the continuous variant of A0...A4. However we do not elaborate on whether S_2^c is the general solution of these properties or not. The formulas for computing variance and standard deviation of a continuous variable can easily be derived from (10) and (11) by replacing summation by integration.

7 Other Speedup Definitions

Sometimes, the user of a refined system B, who is interested in the speedup, wants to know about the variation of the speedup, but only in that portion of the variation caused by the refined system B. A typical example is the parallelisation of a randomized search algorithm A. Here, T^A may have a great variance and thus cause a great variance of the variable T^A/T^B. The user of B, who compares the performance of B with that of A wants to know the speedup and the fluctuation of the speedup caused by the fluctuations of the performance of B. Both can be integrated in the same diagram, if the speedup definition

$$S_4(T^A, T^B) \equiv G\left(\frac{E(T^A)}{T^B}\right) = E(T^A)\ G\left(\frac{1}{T^B}\right)$$

is being used where G and E stand for the geometric and the arithmetic mean. This definition can be seen as a reasonable compromise between S_1 and S_2. Using the geometric mean of $1/T^B$ is necessary due to the same reasons as above. Note that this definition is just a special case of S_2 where the distribution of T^A is collapsed to the single value $E(T^A)$.

A case that needs very special considertion is that of infinite[5] running times. In this case none of the above presented definitions is applicable. A possible remedy for the case that more than 50% of the runs are finite is the Median. However, the Median is not continuous and does not satisfy the requirements P0–P2 and therefore is of limited use for an objective comparison of systems. If more than half of the running times are infinite, it is very hard to find a reasonable performance measure.

[5]In case of experimental results no infinite running time can ever be observed. However, it is common practice to define all runs above a certain time threshold as having quasi infinite running time.

One way to define the speedup for such a system might be

$$\frac{\frac{\text{number of finite runs of } B}{\text{total number of runs of } B}}{\frac{\text{number of finite runs of } A}{\text{total number of runs of } A}} \cdot S_x(T_*^A, T_*^B)$$

where S_x is any one of the presented speedup definitions and the distributions of T_*^A and T_*^B are defined only for the finite runs. This works as long as T_*^A as well as T_*^B have at least one finite value, i.e. nonzero probability for finite runs.

8 Conclusion

We are now able to give the following recipe to anybody in charge of (either experimentally or theoretically) evaluating the relative performance of two systems A and B with varying running times: If both A and B are executed very often and the total running time of all runs is of interest, then only the ratio of the average running times is the relevant figure, i.e. the classical speedup definition S_1 must be used. If, however, the typical application of A and B involves only one run, the user wants to know which value of the ratio T^A/T^B he can expect to occur in a representative sample. We showed that in this case one has to apply the definition S_2 which uses the geometric mean. The confidence in this value however may depend on the quasi standard deviation $D_{\log}$ of the distribution, which is an appropriate measure for the variation of S_2.

Acknowledgements

I am grateful to Richard Karp for encouraging me to publish this idea and to Janos Aczél for his support on functional equations. I also thank Nati Linial, Ulrich Huckenbeck, Andreas Goerdt, Christoph Goller Tino Gramss and Urs Martin Künzi for helful comments and discussions.

References

[AA86] J. Aczél and C. Alsina. On Synthesis of Judgements. *Socio-Econ. Plann. Sci.*, 20(6):333–339, 1986.

[Acz46] J. Aczél. The Notion of Mean Values. *Norske Vid. Selsk. Forh. (Trondheim)*, 19:83–86, 1946.

[Acz66] J. Aczél. *Lectures on Functional Equations and their Applications*, volume 19 of *Mathematics in Science and Engineering*. Academic Press, 1966.

[AS83] J. Aczél and T. L. Saaty. Procedures for Synthesizing Ratio Judgements. *J. Math. Psych.*, 27:93–102, 1983.

[Ert92] W. Ertel. Random Competition: A Simple, but Efficient Method for Parallelizing Inference Systems. In *Parallelization in Inference Systems*, pages 195–209. LNAI 590, Springer-Verlag, 1992.

[FW86] P. J. Fleming and J. J. Wallace. How not to Lie with Statistics: The Correct Way to Summarize Benchmark Results. *Comm. of the ACM*, 29(3):218–221, 1986.

[GK93] A. Goerdt and U. Kamps. On the Reasons for Average Superlinear Speedup in Parallel Backtrack Search. submitted for publication, 1993.

[MG85] R. Mehrotra and E. F. Gehringer. Superlinear Speedup Through Randomized Algorithms. In *International Conference of Parallel Processing*, pages 291–300. IEEE, 1985.

[Nat89] K. S. Natarajan. Expected performance of parallel search. In *International Conference of Parallel Processing*, pages 121–125, 1989.

[RK88] V. N. Rao and V. Kumar. Superlinear speedup in parallel search. In *Proc. of Foundations of Software Technology and Theor. Comp. Sci.*, New Dehli, 1988.

[Rob90] F. S. Roberts. Merging Relative Scores. *J. Math. Analysis and Applications*, 147:30–52, 1990.

[SG91] X.-H. Sun and J.L. Gustafson. Toward a Better Parallel Performance Metric. *Parallel Computing 17*, pages 1093–1109, 1991.

[SMV88] E. Speckenmeyer, B. Monien, and O. Vornberger. Superlinear speedup for parallel backtracking. In *Supercomputing*, LNCS 297, pages 985–994. Springer Verlag, 1988.