Using Computational Similarity to Analyze the Performance Data of the NAS Parallel Benchmarks

Erich Strohmaier
Computing Center, University of Mannheim
D–68131 Mannheim, Germany
RUM 44/95

April 1995

Abstract

One of the major problems in using complex application benchmarks is the difficulty to relate the measured timing information to basic properties of the hardware and of the application used. In this paper we extend the concept of computational similarity invented by Roger Hockney to analyze complex benchmarks. We demonstrate this method for the published performance data of the NAS Parallel Benchmarks. We find that it is only to a very limited amount possible to get information about the details of the timing relations. For a detailed study, access to the actual implementation from the different vendors would be necessary to get analytical timing relations.

Keywords: Benchmarks; performance; characterization; dimensionless ratios; dimensionless numbers; computational similarity; scaling; speedup; Amdahl's Law; NAS Parallel Benchmark.

1 Introduction

The NAS Parallel Benchmarks have been developed at the NASA Ames Research Center. In the last three years extensive performance data has been reported for parallel machines based on the NAS Parallel Benchmarks (NAS PB) [5], [6]. But unfortunately we have to stay with the table of the results without being able to reduce the amount of information. This could be done by introducing a limited number of parameters which hopefully will be characteristic for the system and the algorithms e.g. the hard- and software used. In previous studies [2], [3] we showed that there is strong statistical evidence that only four of the eight NAS Parallel Benchmarks are independent of each other. This is an indication that a more condensed description of the measured data should be possible.

With his concept of computational similarity [1] Roger Hockney showed a possible way to express the scaling behaviour of a whole class of applications by the same set of parameters. Along the line of his concept we developed a generalized version of his ideas [4]. We did this in a way which, in our opinion, is especially well suited to study the timing behaviour of complex applications. The method is also usable if analytical timing relations are not deducable. The central ideas of computational similarity using dimensionless quantities to characterize the timing and scaling are in our case still usable. Hereby we show that many details of the applications and of the systems can not be deduced from measurements alone. In order to do this, analytical timing relations are needed. This fact also means that certain timing relations might not be distinguishable at all.

In this paper we use this method to analyze the NAS PBs. We find that only very little information can be gained from timing information of such highly tuned applications. For a better understanding of the published data you definitely would need access to the implementation used by the vendors.

After the introduction of our extension of the concept of computational similarity in chapter 2, we briefly discuss several important basic timing relations in chapter 3. In chapter 4 we apply our method for Amdahl's Law to the NAS PB, while in chapter 5we try to find the best fitting models for the NAS PBs. Our results are summarized in the final chapter 6.

2 The Concept of Computational Similarity

The basic idea of Roger Hockney for the concept of computational similarity was to introduce the dimensionless ratios of certain parameters to the timing relation. These ratios characterize the hard- and software used. He showed this concept for the timing model including the three most important terms, namely computation, communication and message startup. For each of these terms there exists a hardware $(r^s_\infty, r^c_\infty, t^c_0)$ and a related software parameter $(s^s(N;p), s^c(N;p), q^c(N;p))$ which depends on the problem size and on the number of processors and p. The timing relation is then given by

$$T(N;p) = \frac{s^s(N;p)}{r^s_\infty} + \frac{s^c(N;p)}{r^c_\infty} + t^c_0 q^c(N;p) \quad (1)$$

We now start with the generalized version of this timing relation.

$$T(N;p) = \sum_{j=1}^{n} t_j = \sum_{j=1}^{n} \frac{w_j(N;p)}{r_j} \quad (2)$$

If we divide computational work in parts that can be overlapped with communication, this timing relation is universal.

We now split the different works $w_j(N;p)$ into a sum of products. One factor $u_i(N;p)$ of each term contains all the dependencies on p and can, but in most cases will not depend on N. The other factor $s_j^i(N)$ then depends only on N. Furthermore, we will call u_i the characteristic functions as they contain all the dependencies on p.

$$T(N;p) = \sum_{j=1}^{n} \sum_{i=1}^{m} \frac{s_j^i(N) u_i(N;p)}{r_j} \tag{3}$$

This split is slightly different from the one used in [1].

We now normalize equation 3 such that each characteristic function has a dimensionless parameter as factor which can vary between 0 and 1. This can be achieved by a normalization with the time $\frac{w_T}{r_T}$. This is the quotient of a total work w_T which depends only on N and by a total performance r_T. w_T and r_T are not defined otherwise. So for symmetry reasons we choose this way for writing down a time. This free parameter introduced has to be fixed by an additional condition on the new dimensionless parameters.

$$\begin{aligned}
T'(N;p) &= \frac{T(N;p)}{\frac{w_T}{r_T}} = \sum_{j=1}^{n} \sum_{i=1}^{m} \frac{s_j^i(N)}{w_T} \frac{r_T}{r_j} u_i(N;p) && (4) \\
&= \sum_{i=1}^{m} \left(\sum_{j=1}^{n} \delta_j^i(N) \right) u_i(N;p) \quad \text{with} \quad \delta_j^i(N) = \frac{s_j^i(N)}{w_T} \frac{r_T}{r_j} && (5) \\
&= \sum_{i=1}^{m} \delta_i(N) u_i(N;p) \quad \text{with} \quad \delta_i = \sum_{j=1}^{n} \delta_j^i(N) && (6)
\end{aligned}$$

under the additional condition

$$\sum_{i=1}^{m} \delta_i(N) = 1 \tag{7}$$

By this condition the degree of freedom introduced by the arbitrary normalization is fixed.

If we now assume that the details (that means the functions $s_j^i(N)$) of the timing relation are unknown, we still can use equation 6 to fit the measured time. We then get numerical values for the $\delta_i(N)$. This is why we will not be able to get back the split of the $\delta_i(N)$ into the $\delta_j^i(N)$ and we will in general not be able to derive the amount of different kinds of works done in the calculation. This fact seems to be even worse as in $\delta_i = \sum_{j=1}^{n} \delta_j^i(N)$ different types of works will influence one fitted value $\delta_i(N)$ for a characteristic function $u_i(N;p)$.

Thus we can state that in general *all timing relations which lead to the same set of characteristic functions are not distinguishable by timing information regardless which type of work they represent!*

In many cases, however, a meaningful interpretation should be possible if the timing relation contains only a few dominating contributions.

For discussion of scaling we can get the optimal number of processors by

$$\left.\frac{\partial T'(N;p)}{\partial p}\right|_{p=\tilde{p}} = \sum_{i=1}^{m} \delta_i(N) \left.\frac{\partial u_i(N;p)}{\partial p}\right|_{p=\tilde{p}} = 0 \tag{8}$$

The optimal self-speedup is then given by

$$\tilde{S_p} = \frac{T'(N;1)}{T'(N;\tilde{p})} \tag{9}$$

3 Different Simple Timing Relations

After the development of the theoretical basis for our work, we now will briefly show and discuss the timing relations for different basic operations on a parallel system. These are in general given by

$$t_i = \frac{w_i(N;p)}{r_i} = \sum_{j=1}^{n} \frac{s_j^i(N)}{r_j} u_i^j(N;p) \tag{10}$$

3.1 Parallel work

This type of work is defined to be ideally parallelizable on all processors

$$t_p = \frac{s_p(N)}{r_p}\frac{1}{p} \quad \text{with} \quad u_p = \frac{1}{p} \tag{11}$$

3.2 Sequential work

Here the execution time is independent of the number of processors used

$$t_s = \frac{s_s(N)}{r_s} \quad \text{with} \quad u_p = 1 \tag{12}$$

3.3 Overall startup or global overhead

This is a general overhead and thus constant. It differs from sequential work as no useful work is done. Thus the unit of work is different.

$$t_O = const \quad \text{with} \quad u_O = 1 \tag{13}$$

3.4 Pipeline startup

The performance of pipelines or vectorunits can be described by

$$t_v = t_0 + \frac{s_v(N)}{r_v}\frac{1}{p} \quad \text{with} \quad u_v^1 = 1 \quad \text{and} \quad u_v^2 = \frac{1}{p} \tag{14}$$

Here we assume that the startup work can not be parallelized. For the first time we have a simple timing relation which will influence two terms in the overall timing relation of a more complex calculation.

3.5 Reduction operation as a binary tree operation

Here we assume that $N \geq p$. Then the first part of the computation can be done in parallel. The last p operations are done in a binary tree-like operation

$$t_r = \frac{s_r(N)}{r_r}\left(\frac{N-p}{p} + \frac{log(p)}{p}\right) \quad \text{with} \quad u_r^1 = 1, \quad u_r^2 = \frac{1}{p} \quad \text{and} \quad u_r^3 = \frac{log(p)}{p} \tag{15}$$

This is a another simple example where a single operation contributes to different characteristic functions u_i. The limitation to $N \geq p$ could be removed by using step function $\Theta(x)$ as seen in the next examples.

3.6 Limited parallelism

If the maximum amount of parallelism in the calculation can be smaller than the number of processors, we have to distinguish two possible cases. We formulate this by using the step function $\Theta(x)$, which is 0 if $x \leq 0$ and 1 otherwise.

$$t_{pl} = \frac{s_{pl}(N)}{r_{pl}}\frac{1}{p_{crit} - (p_{crit} - p)\Theta\left(p_{crit} - p\right)} = \begin{cases} \frac{s_{pl}(N)}{r_{pl}}\frac{1}{p} & : \quad p_{crit} > p \\ \frac{s_{pl}(N)}{r_{pl}}\frac{1}{p_{crit}} & : \quad p_{crit} \leq p \end{cases} \tag{16}$$

Thus we have potential contributions to the characteristic functions

$$u_{pl}^1 = 1 \quad \text{and} \quad u_{pl}^2 = \frac{1}{p}$$

3.7 Broadcast implemented as a binary tree communication

If we assume that the broadcast is implemented in a binary tree communication pattern, the execution time goes up with $log(p)$

$$t_b = \frac{s_b(N)}{r_b}log(p) \quad \text{with} \quad u_b = log(p) \tag{17}$$

3.8 Pairwise communication without contention

As long as no network contention happens this kind of work is also ideally parallelizable but has fixed execution time!

$$t_{pc} = \frac{s_{pc}(N)}{r_{pc}} \quad \text{with} \quad u_{pc} = 1 \tag{18}$$

3.9 Critical section

Here we assume that all p processors try to enter the critical section at the same time. Thus the execution time scales up with p.

$$t_{cs} = \frac{s_{cs}(N)}{r_{cs}} p \quad \text{with} \quad u_{cs} = p \tag{19}$$

3.10 Barrier synchronization

There are different ways to implement a barrier synchronization. It could be implemented in a single global instance. Then we would get a similar behaviour to the one of a critical section. It could also be implemented by using local instances and a combination operation on them. This would lead to a behaviour like a broadcast communication.

$$u_{bs} = p \quad \text{or} \quad u_{bs} = log(p) \tag{20}$$

Thus this is an example where the characteristic function which we have to use depends on the system used.

3.11 Central resources used by all processors

Here the same argument applies as for critical sections.

$$t_{cr} = \frac{s_{cr}(N)}{r_{cr}} p \quad \text{with} \quad u_{cr} = p \tag{21}$$

3.12 Cache effects

We assume for simplicity reasons that the amount of work for moving data from and to the cache is proportional to the data out of cache per processor. This work is assumed to be parallelizable! For describing the different cases possible we use the step function $\Theta(x)$.

$$t_{ca} = \frac{s_{ca}(N)}{r_{ca}} \frac{1}{p} \Theta\left(\frac{f(N)}{p} - d_{crit}\right) = \begin{cases} 0 & : \frac{f(N)}{p} \leq d_{crit} \\ \frac{s_{ca}(N)}{r_{ca}} \left(\frac{f(N)}{p^2} - \frac{d_{crit}}{p}\right) & : \frac{f(N)}{p} > d_{crit} \end{cases} \tag{22}$$

$$\text{with} \quad u^1_{ca} = \frac{1}{p^2} \quad \text{and} \quad u^2_{ca} = \frac{1}{p}$$

Here again a single operation influences the δ_i of different characteristic functions u_i. A more realistic description of cache behaviour may be found in [8].

3.13 Overlap between communication and computation

Here we look at the communication work which can only partially be overlapped with computational work depending on the problem size per processor. Again we use a step function $\Theta(x)$ to describe whether all communication can be overlapped or not. We also assume that the amount of communication work per processor is independent of the processor number p.

$$t_{co} = \frac{s_{co}(N)}{r_{co}} \Theta\left(N_{crit} - \frac{f(N)}{p}\right) = \begin{cases} 0 & : \; N_{crit} \leq \frac{f(N)}{p} \\ \frac{s_{co}(N)}{r_{co}}\left(N_{crit} - \frac{f(N)}{p}\right) & : \; N_{crit} > \frac{f(N)}{p} \end{cases} \tag{23}$$

$$\text{with} \quad u^1_{co} = 1 \quad \text{and} \quad u^2_{co} = \frac{1}{p}$$

Here again we get negative contributions to the parameter δ_i of the characteristic function $u_i = \frac{1}{p}$.

3.14 Overview on the basic operations

In table 1 we finally show which kind of work contributes to the different characteristic functions seen in the previous examples. We do not list the step functions but list their possible contributions if their arguments are positive.

3.15 Timing Relations used for analyzing the NAS PBs

Looking at table 1 we now can easily write down a more general timing relation in the form of equation 6

$$T'(N;p) = \sum_{i=1}^{m} \delta_i(N) u_i(N;p) \tag{24}$$

with the set of characteristic functions

$$u_1 = \frac{1}{p^2} \tag{25}$$

$$u_2 = \frac{1}{p} \tag{26}$$

$$u_3 = \frac{log(p)}{p} \tag{27}$$

kind of work	u_i	$\frac{1}{p^2}$	$\frac{1}{p}$	$\frac{log(p)}{p}$	1	$log(p)$	p
parallel	t_p		$+$				
serial	t_s				$+$		
overhead	t_O				$+$		
vectorpipe	t_v		$+$		$+$		
reduction	t_r		$+$	$+$	$-$		
lim. parallelism	t_{pl}		$(+)$		$(+)$		
broadcast	t_b					$+$	
pairwise comm.	t_{pc}				$+$		
critical section	t_c						$+$
barrier	t_{bs}					[+]	[+]
central resource	t_{cr}						$+$
cache	t_{ca}	$(+)$	$(-)$				
comm. overlap	t_{co}		$(-)$		$(+)$		

Table 1: The contributions of different kinds of work to different characteristic functions u_i are shown. '+' indicates positive contributions and '-' negative ones. Brackets '()' indicate the cases where a step function applies. '[]' indicate a system-dependent selection of the characteristic function.

$$u_4 = 1 \tag{28}$$
$$u_5 = log(p) \tag{29}$$
$$u_6 = p \tag{30}$$

Here we sorted the characteristic functions by their influence on the speedup. For simplicity reasons we do not include the step functions seen in the previous chapter. However, we assume that their arguments might be positive and that these cases contribute to the times measured.

4 Fitting Amdahl's Law to the NAS PBs

For deriving his well known law [9], Amdahl used only two kinds of works namely serial and parallel. This gives the timing relation

$$T'(N;p) = \delta_1 + \delta_2 \frac{1}{p} \tag{31}$$

with the additional condition $\delta_1 + \delta_2 = 1$. Identifying δ_2 as parallelization ratio α we recognize the usual formulation of Amdahl's Law. The only difference is that we use dimensonless time.

Thus the self-speedup is given by

$$\tilde{S_p} = \frac{T'(N;1)}{T'(N;\tilde{p})} = \frac{1}{(1-\delta_2) + \frac{\delta_2}{p}} \tag{32}$$

To relate this dimensionless time to physical time we need a reference parameter r_1. This gives

$$T(N;p) = \frac{T'(N;p)}{r_1} = \frac{1}{t_1}\left((1-\alpha) + \frac{\alpha}{p}\right) \tag{33}$$

Also r_1 can be interpreted as a dimensionless reference preformance on a certain machine for which $r_1 = 1$ is fixed.

We fitted Amdahl's Law to the NAS PB to look whether this simple model for performance is already able to explain the measured performances or whether there is statistical room for more sophisticated models. As we wanted to calculate error terms we did this only for systems for which performance data of at least three different system sizes are reported. We also allowed α values greater than one, which does not make sense in a rigid application of Amdahl's Law. The results are shown in tables 2 and 3 for the parametrization given in equation (33). Some of the systems show α values slightly greater than one. This can be seen as an indication of superlinear speedups. In this case Amdahl's Law is too limited and can not be extrapolated to unlimited processor numbers.

The resulting parameters shown in tables 2—3 give a good characterization and overview on the different systems and on the implementations of the benchmarks. For instance, it is quite easy to see extraordinarily good or bad implementations and results.

In most cases Amdahl's Law fits very well to the data, giving small error bounds for any prediction typically in the range of a few percent. Examples for the fitted curves for some systems and class A problem sizes are given in fig 1 and 2. Here we also show as examples the error bounds for the fits of the BT and FT benchmarks. The errors for BT are typically quite small while the errors for FT are sometimes quite big.

5 General Fit of Characteristic Functions to the NAS PB

We now use the general timing relation 24 to fit the timing information of the NAS PBs. As the number of observations is quite small we are, of course, not able to fit the full model in a meaningful way. Rather than that we start with timing models with one characteristic function. After that we try to fit all combinations of two characteristic functions to the data. In table 4 we show for all systems the best two models based on only one characteristic function. In table 5 we show the best model with two characteristic functions.

r_1 of Class A	EP	MG	CG	FT	IS	LU	SP	BT
CM2	0.00409	0.00103	0.00217	0.00185	0.00085	0.00121	0.00084	0.00151
CM5	0.18736	0.03593	0.02024	0.09819	0.00778	0.02894	0.06697	0.10990
CM5E	0.35747	0.19657	0.03644	0.17422	0.06013	0.08540	0.09302	0.17805
KSR1	0.07652	0.03702				0.04660	0.03918	0.05632
Meiko CS2	0.20828	0.20783		0.16815				
nCube 2s	0.02351	0.00897	0.00633	0.00677	0.00764	0.00403	0.00538	0.00981
SGI PowChal	0.51472		0.48834	0.51727		0.41571	0.55089	0.57377
IBM SP1		0.15804	0.06619	0.08340	0.09351	0.13906	0.13248	0.21735
IBM SP2	0.35536	0.43788	0.38369	0.21532	0.25259	0.36609	0.33067	0.41855
SPP1000	0.32955	0.10930	0.06169	0.16566	0.14220	0.15734	0.18916	0.29769
Cray T3D	0.22689	0.13125	0.04925	0.15273	0.08238	0.10443	0.13640	0.20387
VPP500	2.78729	3.95379	2.25813	2.66994	4.99801		2.54733	5.07530
Paragon XP	0.19190	0.05103	0.10621	0.07056		0.03230	0.04044	0.06887
YMP C90	2.72625	3.03040	3.45228	3.08355	3.58960	2.21656	2.41429	2.16728
YMPel		0.30116	0.28088	0.31413	0.27584	0.22234	0.27992	0.23899

Table 2: Single processor performances r_1 in NAS PB units obtained by a fit of Amdahl's Law to all NAS PBs for class A problem size. Missing values indicate measurements for only two or less system sizes.

α of Class A	EP	MG	CG	FT	IS	LU	SP	BT
CM2	0.99988	0.99984	0.99937	0.99976	0.99939	0.99898	0.99934	0.99978
CM5	0.99988	0.99780	0.99727	0.98578	0.99968	0.99281	0.99089	0.99122
CM5E	0.99932	0.99592	0.99821	0.99039	0.99910	0.99092	0.99563	0.99697
KSR1	1.00008	0.99856				0.98677	0.99421	0.99754
Meiko CS2	0.99741	0.99439		0.98897				
nCube 2s	1.00000	0.99984	0.99877	1.00001	0.99985	0.99935	0.99959	0.99973
SGI PowChal	0.99930		0.91776	0.86025		0.98009	0.98286	0.99578
IBM SP1		0.99498	0.98871	0.99700	0.99021	0.98916	0.98699	0.99262
IBM SP2	1.00005	0.99583	0.95535	0.99733	0.99484	0.98805	0.99102	0.99445
SPP1000	0.99894	0.96935	1.00027	0.97542	0.93409	0.96209	0.96072	0.98246
Cray T3D	0.99999	0.99950	0.99858	0.99926	0.99788	0.99879	0.99941	0.99980
VPP500	0.99901	0.97157	0.90730	0.99368	0.68633		0.94725	0.99857
Paragon XP	0.99985	0.99704	0.97843	0.99517		0.99661	0.99551	0.99648
YMP C90	0.99873	0.92711	0.96165	0.96262	0.97633	0.94180	0.99533	0.98383
YMPel		0.79131	0.87243	0.91428	0.91003	0.87466	0.82094	0.88249

Table 3: Parallelization ratios α obtained by a fit of Amdahl's Law to all NAS PBs for class A problem size. Missing values indicate measurements for only two or less system sizes.

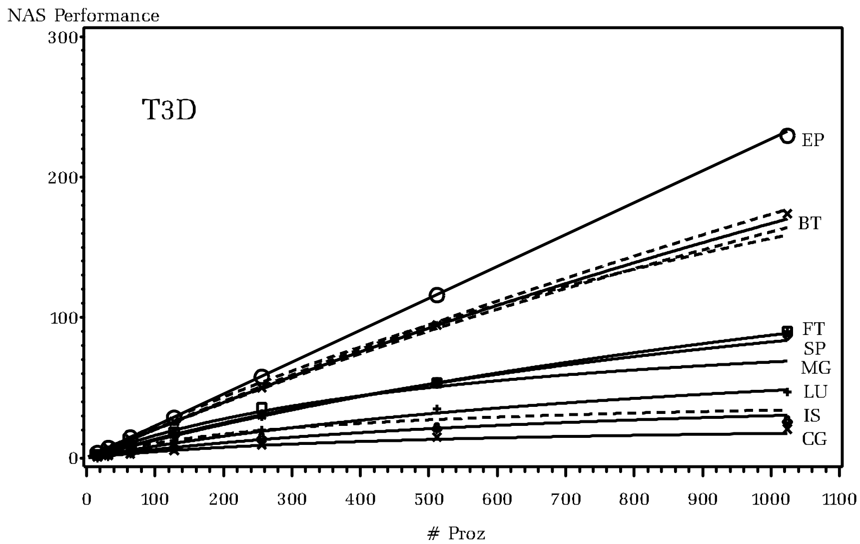

Figure 1: Fit of Amdahl's Law for all NAS PB for the T3D. Error bounds are only shown for BT and FT.

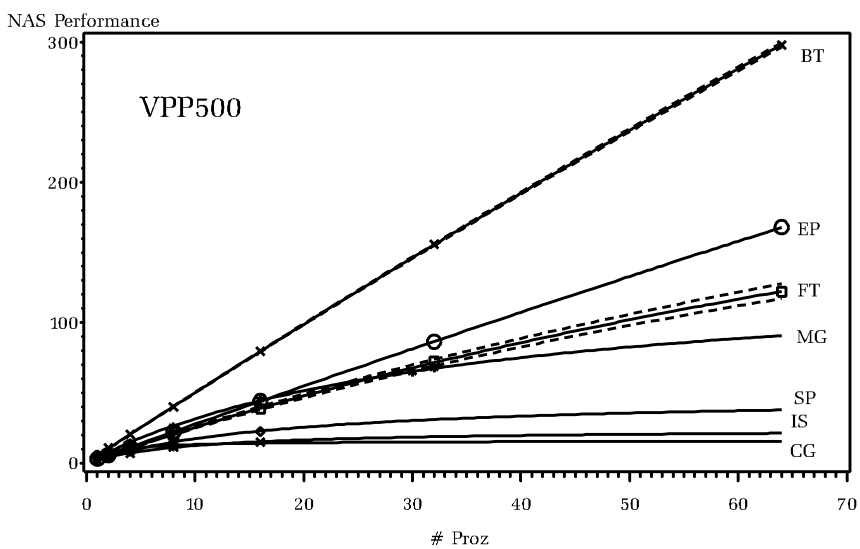

Figure 2: Fit of Amdahl's Law for all NAS PB for the Fujitsu VPP500. Error bounds are only shown for BT and FT.

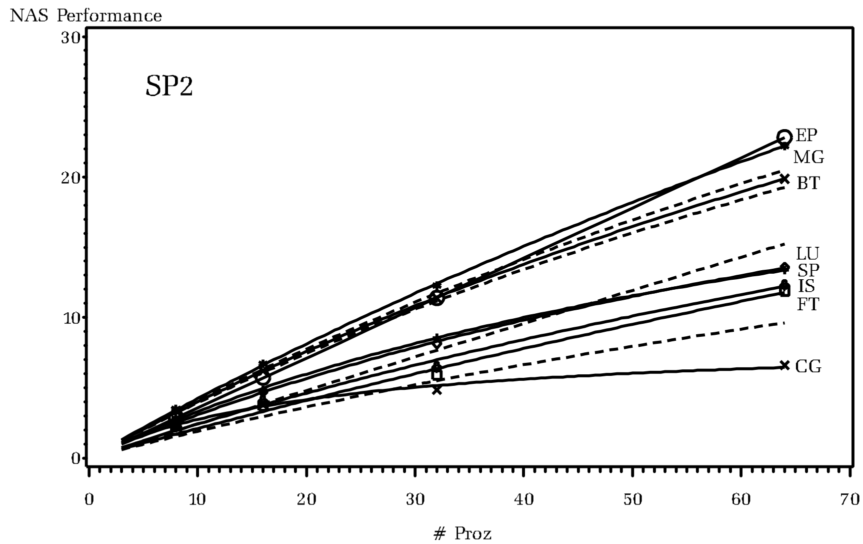

Figure 3: Fit of Amdahl's Law for all NAS PB for the SP2. Error bounds are only shown for BT and FT.

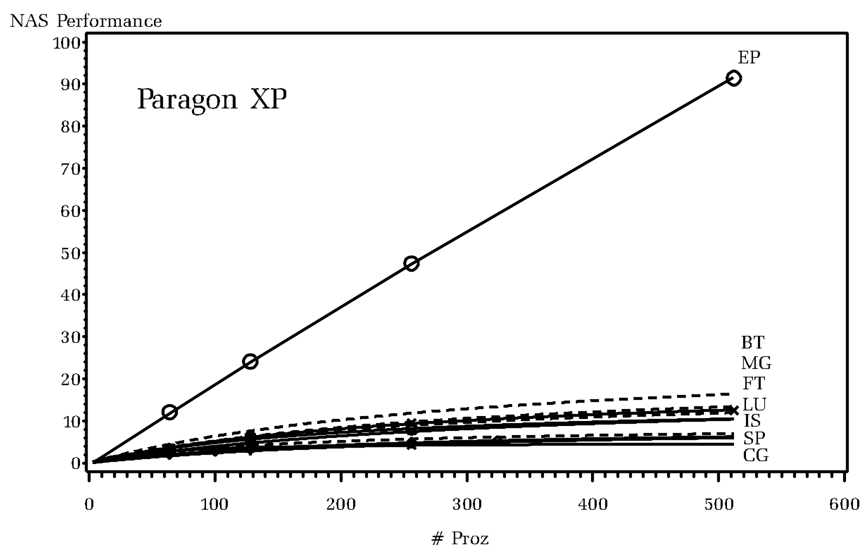

Figure 4: Fit of Amdahl's Law for all NAS PB for the Paragon XP running OSF1.2. Error bounds are only shown for BT and FT.

u_i	EP		MG		CG		FT	
System	best	sec.	best	sec.	best	sec.	best	sec.
CM2	$1/p$	$log(p)/p$	$log(p)/p$	$1/p$	$log(p)/p$	$1/p$	$log(p)/p$	$1/p$
CM5	$1/p$	$log(p)/p$	$log(p)/p$	$1/p$	$1/p$	$log(p)/p$	$log(p)/p$	$1/p$
CM5E	$1/p$	$log(p)/p$	$log(p)/p$	$1/p$	$1/p$	$log(p)/p$	$log(p)/p$	$1/p$
KSR1	$1/p$	$log(p)/p$	$1/p$	$log(p)/p$				
Meiko CS2	$1/p$	$log(p)/p$	$log(p)/p$	$1/p$			$log(p)/p$	$1/p$
nCube 2s	$1/p$	$log(p)/p$	$1/p$	$log(p)/p$	$log(p)/p$	$1/p$	$1/p$	$log(p)/p$
SGI PowChal	$1/p$	$1/p^2$			$1/p$	$1/p^2$	$1/p$	$1/p^2$
IBM SP1			$1/p$	$log(p)/p$	$log(p)/p$	$1/p$	$1/p$	$log(p)/p$
IBM SP2	$1/p$	$log(p)/p$	$1/p$	$log(p)/p$	$log(p)/p$	$1/p$	$1/p$	$log(p)/p$
SPP1000	$1/p$	$1/p^2$	$1/p$	$1/p^2$	$1/p$	$1/p^2$	$1/p$	$1/p^2$
Cray T3D	$1/p$	$log(p)/p$	$1/p$	$log(p)/p$	$1/p$	$log(p)/p$	$1/p$	$log(p)/p$
VPP500	$1/p$	$1/p^2$	$1/p$	$log(p)/p$	$1/p$	$1/p^2$	$1/p$	$log(p)/p$
Paragon XP	$1/p$	$log(p)/p$	$log(p)/p$	$1/p$	1	$log(p)/p$	$1/p$	$1/p^2$
YMP C90	$1/p$	$1/p^2$	$1/p$	$1/p^2$	$1/p$	$1/p^2$	$1/p$	$1/p^2$
YMPel			$1/p$	$1/p^2$	$1/p$	$1/p^2$	$1/p$	$1/p^2$

u_i	IS		LU		SP		BT	
System	best	sec.	best	sec.	best	sec.	best	sec.
CM2	$log(p)/p$	$1/p$	$log(p)/p$	$1/p$	$log(p)/p$	$1/p$	$log(p)/p$	$1/p$
CM5	$log(p)/p$	$1/p^2$	$log(p)/p$	$1/p$	$log(p)/p$	$1/p$	$log(p)/p$	$1/p$
CM5E	$1/p$	$log(p)/p$	$log(p)/p$	$1/p$	$log(p)/p$	$1/p$	$log(p)/p$	$1/p$
KSR1			$log(p)/p$	$1/p$	$log(p)/p$	$1/p$	$log(p)/p$	$1/p$
Meiko CS2								
nCube 2s	$1/p$	$log(p)/p$	$log(p)/p$	$1/p$	$log(p)/p$	$1/p$	$1/p$	$log(p)/p$
SGI PowChal			$1/p$	$1/p^2$	$1/p$	$1/p^2$	$1/p$	$1/p^2$
IBM SP1	$1/p$	$log(p)/p$	$log(p)/p$	$1/p$	$log(p)/p$	$1/p$	$1/p$	$log(p)/p$
IBM SP2	$1/p$	$log(p)/p$	$log(p)/p$	$1/p$	$1/p$	$log(p)/p$	$1/p$	$log(p)/p$
SPP1000	$1/p$	$1/p^2$	$1/p$	$1/p^2$	$1/p$	$1/p^2$	$1/p$	$1/p^2$
Cray T3D	$1/p$	$log(p)/p$	$1/p$	$log(p)/p$	$1/p$	$log(p)/p$	$1/p$	$log(p)/p$
VPP500	$1/p$	1			$1/p$	$1/p^2$	$1/p$	$1/p^2$
Paragon XP			$log(p)/p$	$1/p$	$log(p)/p$	$1/p$	$log(p)/p$	$1/p$
YMP C90	$1/p$	$1/p^2$	$1/p$	$1/p^2$	$1/p$	$1/p^2$	$1/p$	$1/p^2$
YMPel	$1/p$	$1/p^2$	$1/p$	$1/p^2$	$1/p$	$1/p^2$	$1/p$	$1/p^2$

Table 4: Best and second best fitting characteristic functions u_i for a single parameter fit of the NAS PBs of class A.

u_i System f	EP best	MG best	CG best	FT best
CM2	$1/p$, $log(p)$	$log(p)/p$, p	$log(p)$, $1/p$	p, $1/p$
CM5	$1/p$, 1	$1/p$, $log(p)/p$	$1/p$, p	$1/p^2$, 1
CM5E	$1/p$, 1	$1/p$, $log(p)/p$	$1/p$, $log(p)$	$1/p^2$, 1
KSR1	$1/p$, $1/p^2$	$1/p^2$, $log(p)/p$	,	,
Meiko CS2	$1/p$, 1	$1/p$, $log(p)$	,	$1/p$, $log(p)$
nCube 2s	$1/p$, 1	$1/p^2$, $log(p)/p$	1, $1/p$	$1/p$, $log(p)/p$
SGI PowChal	$1/p$, $log(p)/p$	,	$1/p$, $1/p^2$	$1/p$, 1
IBM SP1	,	$1/p$, $log(p)/p$	$log(p)/p$, $1/p$	$1/p$, $log(p)/p$
IBM SP2	$1/p$, 1	$1/p$, 1	1, $1/p$	$log(p)/p$, $1/p^2$
SPP1000	$1/p$, $log(p)/p$	$1/p$, 1	$1/p$, $1/p^2$	$1/p$, 1
Cray T3D	$1/p$, 1	$1/p$, $1/p^2$	$1/p$, 1	$1/p$, 1
VPP500	$1/p$, 1	$1/p$, 1	$1/p$, 1	$1/p$, $log(p)/p$
Paragon XP	$1/p$, 1	$1/p$, p	1, $1/p$	$1/p$, p
YMP C90	$1/p$, 1	$1/p$, p	$1/p$, $log(p)$	$1/p^2$, $log(p)/p$
YMPel	,	$1/p$, p	$1/p$, $log(p)$	$1/p^2$, $log(p)/p$

u_i System	IS best	LU best	SP best	BT best
CM2	$log(p)/p$, p	$log(p)/p$, 1	$log(p)/p$, 1	$log(p)/p$, p
CM5	$1/p$, $log(p)/p$	$log(p)/p$, 1	$1/p$, $log(p)$	$log(p)/p$, p
CM5E	$1/p^2$, $log(p)/p$	$log(p)/p$, $log(p)$	$log(p)/p$, 1	$1/p$, $log(p)/p$
KSR1	,	$1/p$, p	$log(p)/p$, p	$log(p)/p$, $1/p^2$
Meiko CS2	,	,	,	,
nCube 2s	$1/p$, 1	$log(p)/p$, $log(p)$	$1/p$, $log(p)/p$	$1/p$, $log(p)$
SGI PowChal	,	$1/p$, $log(p)/p$	$1/p$, 1	$1/p$, $log(p)/p$
IBM SP1	$1/p$, 1	$1/p^2$, $log(p)/p$	$1/p^2$, $log(p)/p$	$1/p^2$, $log(p)/p$
IBM SP2	$1/p$, $log(p)/p$	$1/p^2$, $log(p)/p$	$1/p^2$, $log(p)/p$	$1/p$, $log(p)/p$
SPP1000	$1/p$, 1	$1/p^2$, $log(p)/p$	$1/p$, $log(p)$	$1/p$, 1
Cray T3D	$1/p$, p	$1/p$, $log(p)/p$	$1/p$, $log(p)/p$	$1/p$, 1
VPP500	$1/p$, 1	,	$1/p$, $log(p)/p$	$1/p$, $1/p^2$
Paragon XP	,	$log(p)/p$, 1	$log(p)/p$, 1	$log(p)$, $1/p$
YMP C90	$1/p$, p	$1/p$, $log(p)$	$1/p$, $log(p)/p$	$1/p$, 1
YMPel	$1/p$, $log(p)$	$1/p^2$, $log(p)/p$	$1/p$, 1	$1/p$, p

Table 5: The best fit with two characteristic functions u_i of the NAS PBs of class A. Tho ordering of the functions does not have a meaning. In many cases e.g. EP the statistical basis for this two parameter fit is quite low. Thus different combinations of characteristic functions often give fits of similar quality.

All the results we present here slightly depend on the exact type of fit. We tried to fit the times T as well as the speedups S_p and their logarithms $log(T)$ and $log(S_p)$. The major difference between these alternatives is the kind of error minimized. Fitting the times T minimizes absolute errors, fitting $log(T)$ minimizes relative errors. A good decision between these possibilities could be made if an error would be assigned to single measurements. But unfortunately rather than calculating mean values and standard deviations from multiple measurements it is still practice in benchmarking to report best measurements only. We believe that both pieces of information are necessary to analyze timing informations in a scientifically safe way.

From table 4 we see that in most cases pure parallel work or the function which characterizes reduction operations fits the data best. Special attention rises the fact that the shared memory systems (SGI Power Challenge, SPP1000, VPP500, YMP C90, YMPel) in most cases show an influence of the cache function but not the other systems.

At table 5 we also see the influence of the other characteristic functions. In many cases the combination of parallel work and the reduction function fits the data best. Again we see big influence of the characteristic function for cache effects especially for the shared memory systems but also for the other cache-based MPPs. In many cases Amdahl'l Law seems to fit the data best.

6 Conclusions

In this paper we presented an extension of the concept of computational similarity developed by Roger Hockney in [1] for analyzing complex benchmarks. We used this concept to analyze the published performance data of the NAS PBs. We found that for such high tuned applications only a very limited amount of information can be gained by fitting the timing information. We presented the best one and two parameter fits found. Only a few characteristic functions show up in the analyses and seem to be suitable to fit the NAS PBs.

A better and more detailed analysis would only be possible if we could have access to the actual implementation. Then you could try to get an analytical timing relation and fit this directly to the measurements.

References

[1] R. Hochney: Computational Similarity. Proc. Workshop on Performance Evaluation and Benchmarking of Parallel Systems 1994 (December 1994)

[2] H. D. Simon and E. Strohmaier: Amdahl's Law and the Statistical Content of the NAS Parallel Benchmarks. Proc. Workshop on Performance Evaluation and Benchmarking of Parallel Systems 1994 (December 1994)

[3] H. D. Simon and E. Strohmaier: Statistical Analysis of NAS Parallel Benchmarks and LINPACK Results. Proc. HPCN Europe 1995, Milano (Mai 1995)

[4] E. Strohmaier: Extending the Concept of Computational Similarity for Analyzing Complex Benchmarks. Submitted to Parallel Computing, 1995

[5] D. H. Bailey, J. Barton, T. Lasinski, and H. D. Simon (editors): The NAS parallel benchmarks. **NASA Technical Memorandum 103863** NASA Ames Research Center, Moffett Field, CA 94035-1000 (July 1993)

[6] D. H. Bailey, E. Barszcz, L. Dagum, and H. D. Simon: NAS parallel benchmarks results 10-94. **NAS Technical Report RNR-94-001** NASA Ames Research Center, Moffett Field, CA 94035 (October 1994)

[7] Parkbench Committe: Public International Benchmarks for Parallel Computers. **Technical Report CS-93-213** Computer Science Department, Unibersity of Knoxville, Tennessee, (November 1993)

[8] V. S. Getov: Performance Characterisation of the Cache Memory Effect. Proc. Workshop on Performance Evaluation and Benchmarking of Parallel Systems 1994 (December 1994)

[9] G. M. Amdahl: Validity of the single processor approach to achieve large scale computing capabilities. Proc. AFIPS 1967 Spring Joint Comput. Conf. **30** (1967) 483–485