The Speedup-Test

Observing speedups with rigorous statistics

Sid-Ahmed-Ali Touati, Julien Worms, Sébastien Briais

University of Versailles Saint-Quentin en Yvelines

Tutorial Outline

1. General introduction
2. Common observed non rigorous experimental methodology
3. Different kinds of observed speedups
4. Statistical significance of the observed speedups
5. Proportion of accelerated programs
6. Conclusion on the Speedup-Test methodology
7. The Speedup-Test Software: tool demo

1. General introduction

1. Non reproducible research results
2. Variability of execution times
3. Related literature on statistics and performance analysis
4. The nature of the collected and analysed data
5. Reminders on some basic definition in statistics

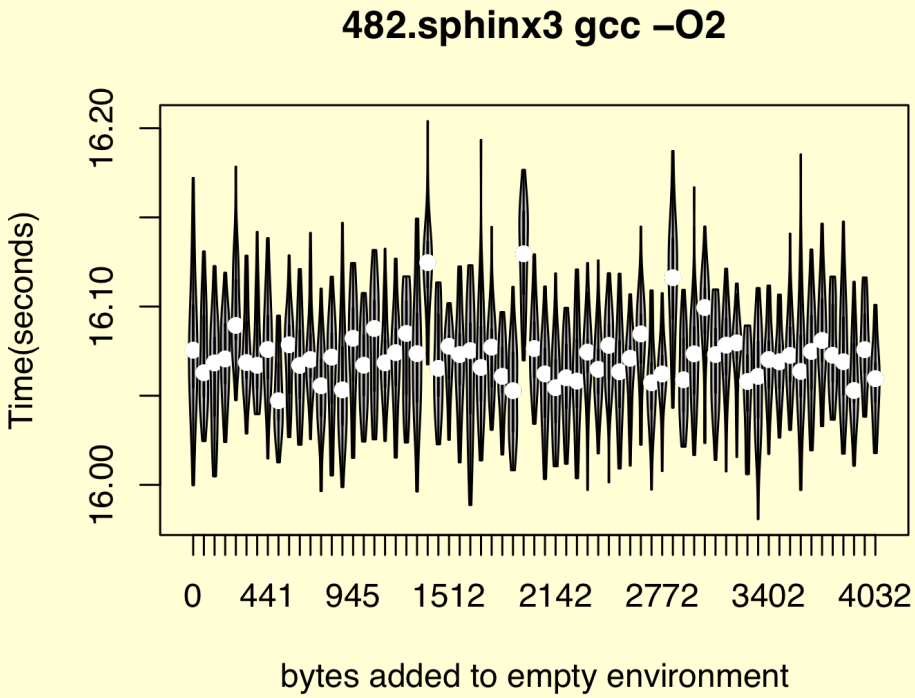
1.1 Non reproducible research results

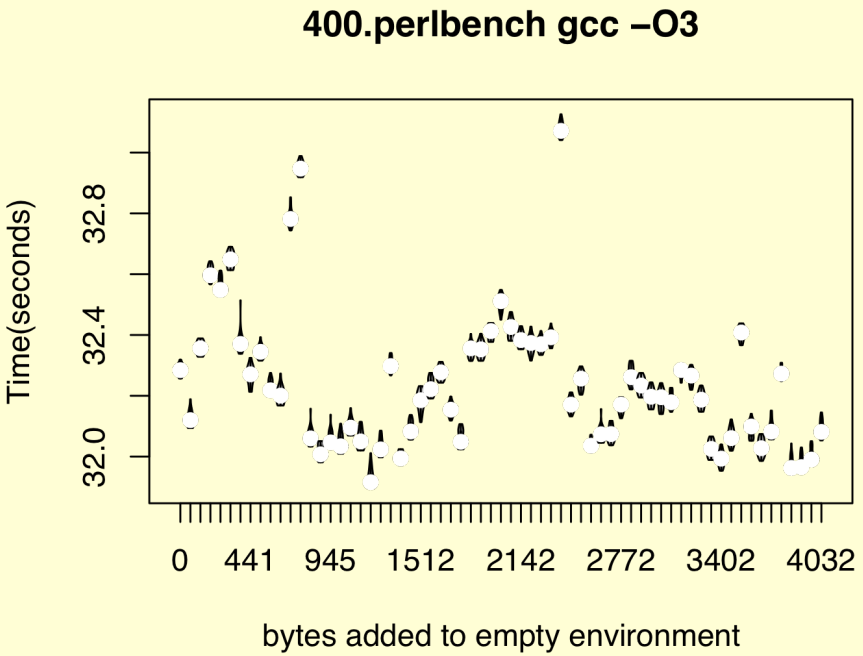
- Non usage of mathematics.
- Non release of software and data.
- Hide experimental details.
- Do not write formal algorithms.
- Usage of deprecated machines and software, exotic OS.
- Doing wrong statistics.

1.2 Variability of execution times

- Long running sequential applications (SPEC2000, SPEC2006)
  - Negligible variations
- Toy and kernel benchmarks, small loops, parallel applications, multicores
  - Observed important variations

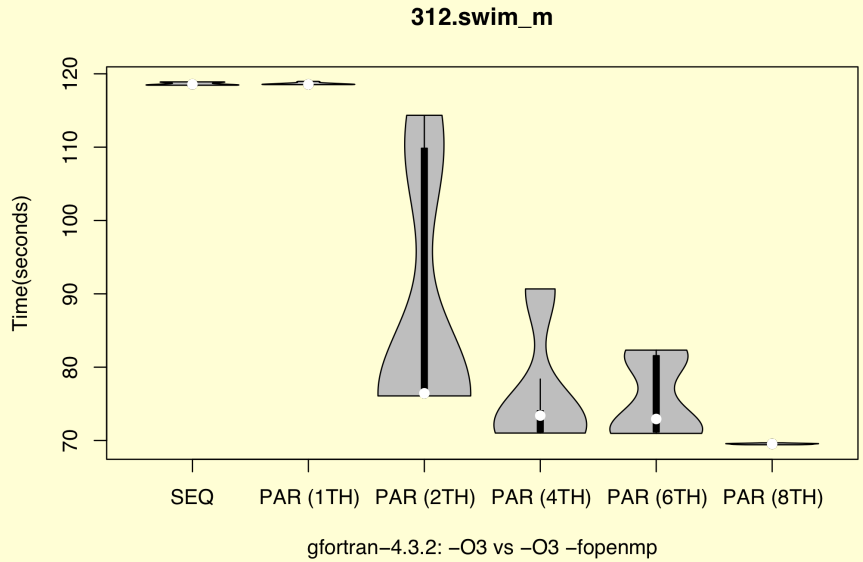
1.2 Variability of execution times

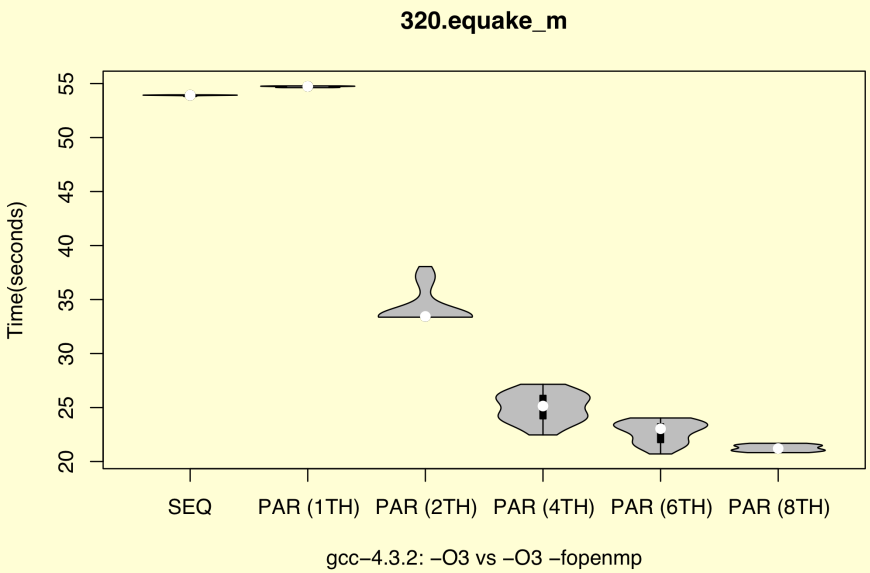




Abdelhafid MAZOUZ and Sid-Ahmed-Ali TOUATI and Denis BARTHOU. *Study of Variations of Native Program Execution Times on Multi-Core Architectures*. IEEE Workshop on Multi-Core Computing Systems (MuCoCoS 2010). Krakow, Poland, February 15, 2010.

1.2 Variability of execution times





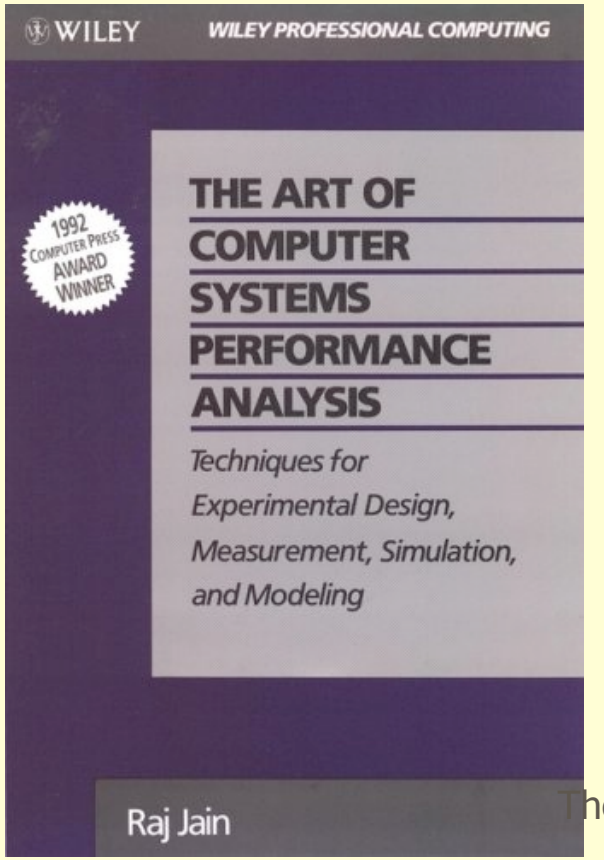
Abdelhafid MAZOUZ and Sid-Ahmed-Ali TOUATI and Denis BARTHOU. *Study of Variations of Native Program Execution Times on Multi-Core Architectures*. IEEE Workshop on Multi-Core Computing Systems (MuCoCoS 2010). Krakow, Poland, February 15, 2010.

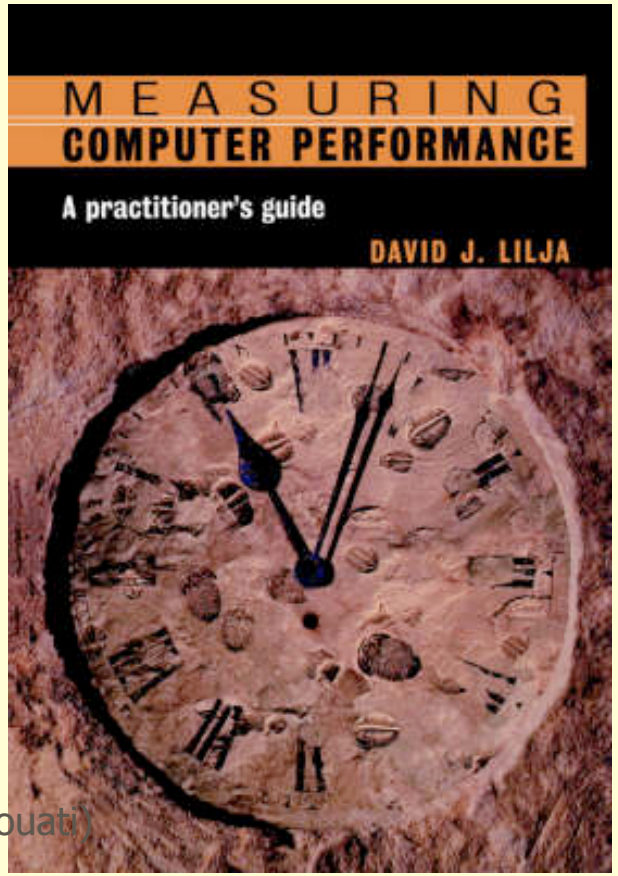
1.2 Why execution times vary?

- Timing measure errors.
- Background tasks.
- OS process scheduling policy.
- Binding of process/threads on processors/cores.
- Interrupts.
- Input/output.
- Starting loader address of the stack (unix shell environment size).
- Branch predictor initial state.
- Cache effects.
- Non deterministic dynamic instruction scheduling and register renaming.
- Temperature of the room (DVS).

1.3 Related literature

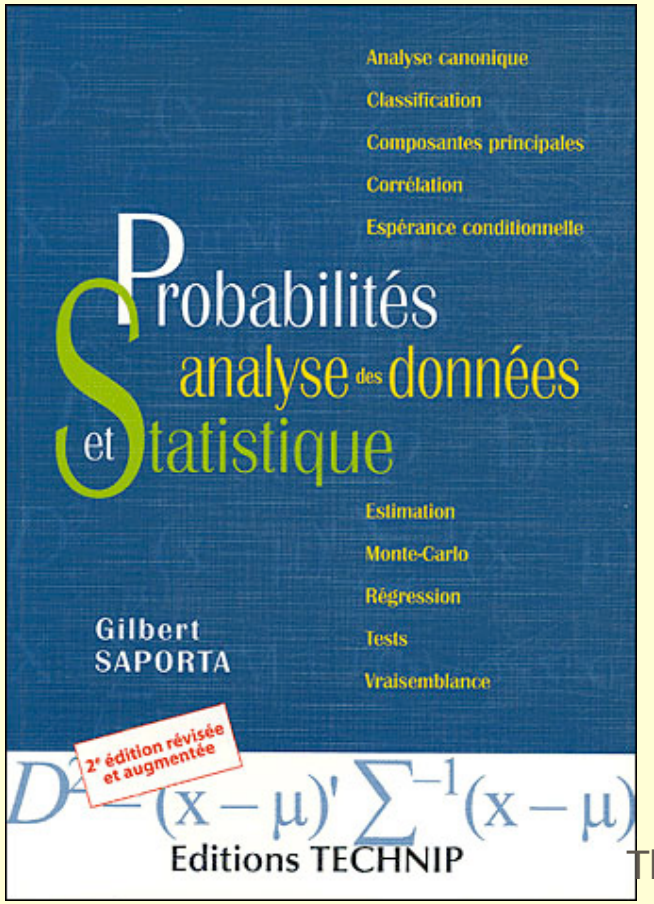
Interesting popular, accessible and digest books

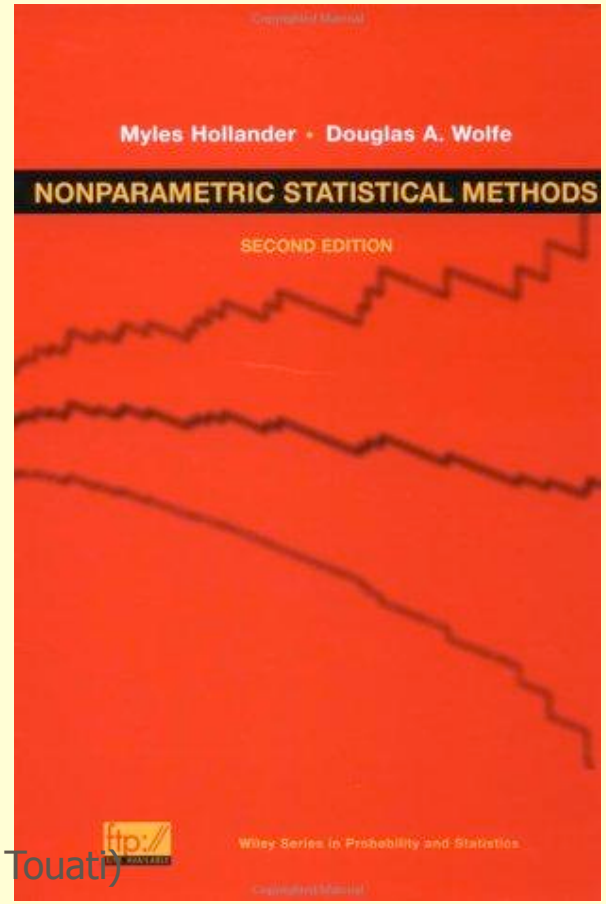




1.3 Related literature

More rigorous books on statistics





1.3 Related literature

- The Speedup-Test is based on the rigorous books: assertions on risk levels are proved under clear hypothesis.
- We will show you some examples of misunderstandings and miss-usage of statistics if we do not follow the rigorous books.

1.4 The nature of the collected and analysed data

UNIVERSITÉ DE VERSAILLES SAINT-QUENTIN-EN-YVELINES

1. Execution times are considered in a *continuous* time unit.
2. If discrete values, the statistics we describe will consider them as continuous values.
3. Discrete and continuous data are two separate fields in statistical theory.

1.5 Reminders on some basic definitions in statistics

- *C(I)*: a code with its input data *I*
- *C'(I)*: an optimised version of *C(I)*
- Executing multiple times *C(I)* and *C'(I)* lead to multiple observed execution times. So we have two samples

$\mathcal{X}=\{x_1,\ldots,x_n\}$

$\mathcal{Y}=\{y_1,\ldots,y_m\}$

1.5 Reminders on some basic definitions in statistics

- X and Y are the random variables describing all the possible execution times of *C(I)* and *C'(I)*
- The density functions of X and Y are unknown, we have just observed two samples $\mathcal{X}$ and $\mathcal{Y}$
- Theoretical means (average): μ_X, μ_Y
- Theoretical medians: *med(X), med(Y)*

1.5 Reminders on some basic definitions in statistics

- Theoretical variances: $\sigma_X^2, \sigma_Y^2,$
- Cumulative distribution function (CDF)

$$F_X(a) = P[X \leq a]$$

- Probability density function is the derivative function of F_X

1.5 Reminders on some basic definition in statistics

- By considering two samples $\mathcal{X}$ and $\mathcal{Y}$
- Sample means: $\overline{X}, \overline{Y}$
- Sample medians: $\overline{med(X)}, \overline{med(Y)}$
- Sample variance: s_X^2, s_Y^2

Doing statistics is not simply reporting the observations on samples, but statistics help to conclude on the theoretical unknown information (distribution functions, means, medians, variance) based on the samples

Tutorial Outline

1. General introduction
2. Common observed non rigorous experimental methodology
3. Different kinds of observed speedups
4. Statistical significance of the observed speedups
5. Proportion of accelerated programs
6. Conclusion on the Speedup-Test methodology
7. The Speedup-Test Software: tool demo

2.1 Biased experiments

- Independence of the successive runs of the same program and input *C(I)*
  - Loop around a kernel
  - Back-to-back executions
  - Playing with hidden influencing factors
  - Background workload
- OS services (DVS) may accelerate/ decelerate the whole system
- Multiple inputs of the same program

2.2 Outlier elimination (min, max)

UNIVERSITÉ DE VERSAILLES SAINT-QUENTIN-EN-YVELINES

- Never remove outliers from your samples, unless they correspond to wrong executions (bugs, crashes, etc.)
- Why should we keep outliers (min, max) ?
  - If it appears in the sample, then it is not a rare event
  - Nothing guarantee that outliers are accidents
  - We know how to make fair statistics in presence of outliers
  - The variability of the collected observations of the heart of the information in statistics

2.3 Using statistical tests without checking the hypothesis

- Use of the Student's t-test without checking if the populations are normally distributed
- The central limit theorem does not prove anything for using the Student's t-test on large samples for any distribution
- In practice we use some statistical tests on large samples, but the declared risk level may not be correct.

2.4 Confusion between discrete and continuous data

- Most of the statistical tests have proved risk levels for continuous data only.
- Hopefully, an execution time can be considered as continuous quantity.
- If the time is measured it with discrete events (clock cycles, hardware performance counters), that is fine, but we should assume it as continuous (real values) not discrete (integer values).

Tutorial Outline

1. General introduction
2. Common observed non rigorous experimental methodology
3. Different kinds of observed speedups
4. Statistical significance of the observed speedups
5. Proportion of accelerated programs
6. Conclusion on the Speedup-Test methodology
7. The Speedup-Test Software: tool demo

3.1 Single program with fixed input

- $\mathcal{X} = \{x_1,\ldots,x_n\}$ the observed execution times of *C(I)*
- $\mathcal{Y} = \{y_1,\ldots,y_m\}$ the observed execution times of *C'(I)*
- X/Y is a "random" value in theory, people want to have a single number

3.1 Single program with fixed input

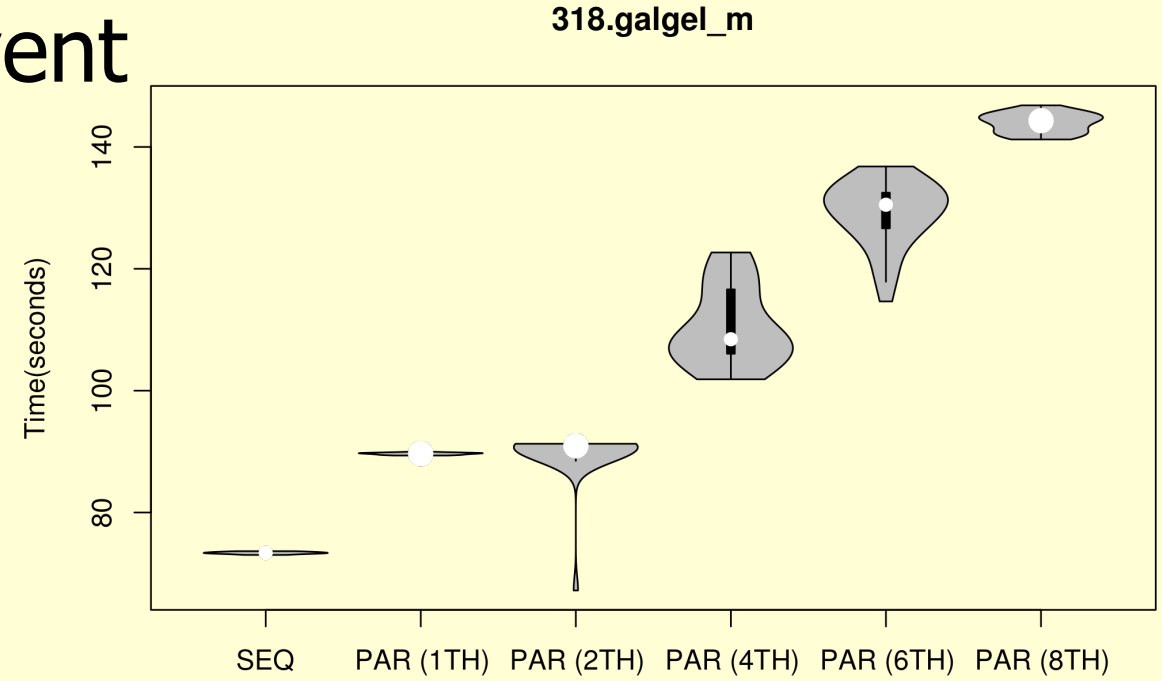
Speedup of the minimum execution time $sp\min(C,I) = \frac{\min_i x_i}{\min_j y_j}$

Speedup of the mean execution time $spmean(C,I) = \overline{X}/\overline{Y}$

Speedup of the median execution time $spmedian(C,I) = \overline{med}(X)/\overline{med}(Y)$

3.2 Why the minimum execution time is a bad metric

The minimum execution time may be a rare event



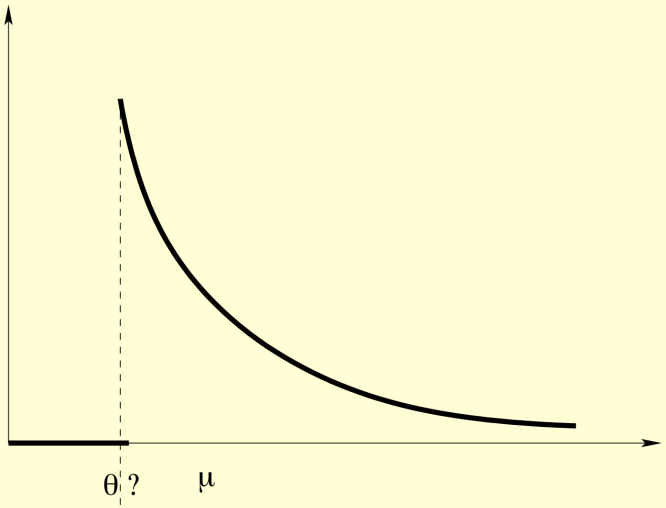
gfortran–4.3.2: –O3 vs –O3 –fopenmp

Abdelhafid MAZOUZ and Sid-Ahmed-Ali TOUATI and Denis BARTHOU. *Study of Variations of Native Program Execution Times on Multi-Core Architectures*. IEEE Workshop on Multi-Core Computing Systems (MuCoCoS 2010). Krakow, Poland, February 15, 2010.

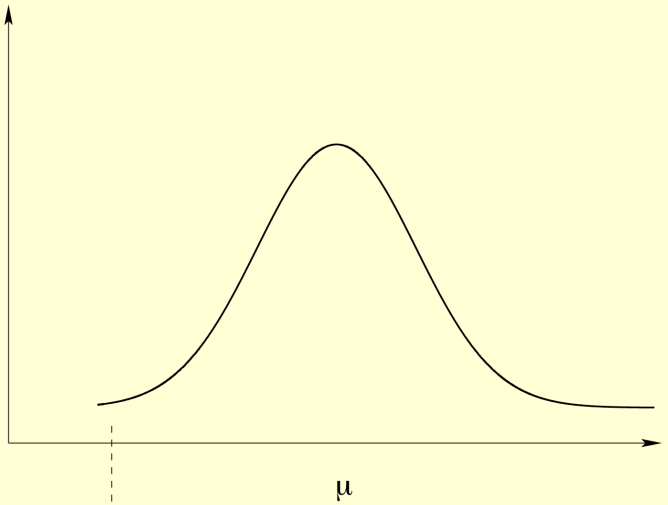
3.2 Why the minimum execution time is a bad metric

- Nothing guarantee that the minimum execution time is a consequence of the optimisation technique
- Extreme values (min and max) computed from samples are highly variable: contrary to the average and the median, extreme values are not normally distributed when we consider multiple samples

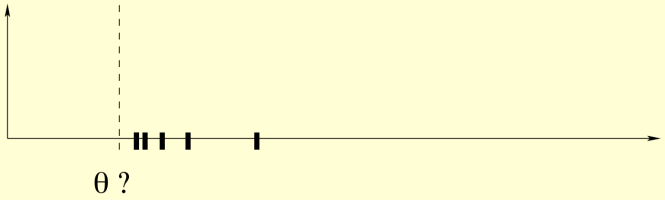
3.2 Why the minimum execution time is a bad metric



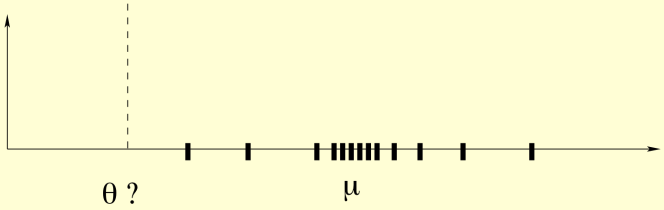
A theoretical exponential distribution



A theoretical gaussian distribution



A sample of observations of an exponential distribution



A sample of observations of a gaussian

3.3. Overall speedup of a set of benchmarks

- Do not use arithmetic mean nor harmonic mean: do not distinguish between long and short running times
- Do not use geometric mean, because we are not faced to a successive speedup on the same benchmark

3.3. Overall speedup of a set of benchmarks

- $W(C_k)$: weight of the benchmark C_k
  - Same weight for all, or fixed by the user, or defined by the profile of usage, or fixed by benchmark institution, etc.

$$\text{overall speedup } S = \frac{\sum_{j=1,b} W(C_j) \times ExecutionTime(C_j, I_j)}{\sum_{j=1,b} W(C_j) \times ExecutionTime(C'_j, I_j)}$$

$$\text{overall gain } G = 1 - \frac{\sum_{j=1,b} W(C_j) \times ExecutionTime(C'_j, I_j)}{\sum_{j=1,b} W(C_j) \times ExecutionTime(C_j, I_j)} = 1 - \frac{1}{S}$$

ExecutionTime(C,I) is the average or the median of observed executions times

Tutorial Outline

1. General introduction
2. Common observed non rigorous experimental methodology
3. Different kinds of observed speedups
4. Statistical significance of the observed speedups
5. Proportion of accelerated programs
6. Conclusion on the Speedup-Test methodology
7. The Speedup-Test Software: tool demo

Statistical significance

- How can we guarantee that the future executions of *C(I) and C(I')* under the same experimental setup would lead to declare that the observed average execution time of *C(I) > C(I')*, or the observed median of *C(I) > C(I') ?*
- Even better, if *C(I)* is not intended to be executed very frequently, how can we guarantee that P[X>Y]>1/2

Hypothesis testing in statistics

- Many tests in statistics are able to decide or not for H_0 a null hypothesis against H_a an alternative hypothesis

Decision of the test	H_0 is the truth	H_a is the truth
H_0 is selected	$1-\alpha$	β
H_a is selected	α	$1-\beta$

Hypothesis testing in statistics

- Most of the statistical tests have proved α risks only
- If the test rejects the null hypothesis H_0, then the risk of error is proved α
- If the test does not reject H_0, we are formally under trouble, but by abuse of usage we say that we accept H_a the alternative hypothesis
  - We say that the confidence level of *accepting* H_a is $1-\alpha$, but this is not proved to be correct

4.1 Statistical significance of the observed speedup of the mean execution time

- The Student's t-test allows to check if $\mu_X > \mu_Y$
- The α risk level is proved only if X and Y are normally distributed
  - The original Student's t-test was designed for normally distributed data with the same variance
  - The Welch's variant of the Student's t-test generalises it to inequal variants

4.1 Statistical significance of the observed speedup of the mean execution time

$$\text{If } X \sim N(\mu,\delta) \text{ then } \sqrt{n}\,\frac{\overline{X}_n - \mu_X}{s_n} \sim t_{(n-1)}$$

- If *X* and *Y* are not normally distributed, one could still use the Student's t-test in practice but
  - The samples sizes must be *large enough*
  - The α risk level is not preserved, but we may know how to bound it if the distribution function of *X* is known

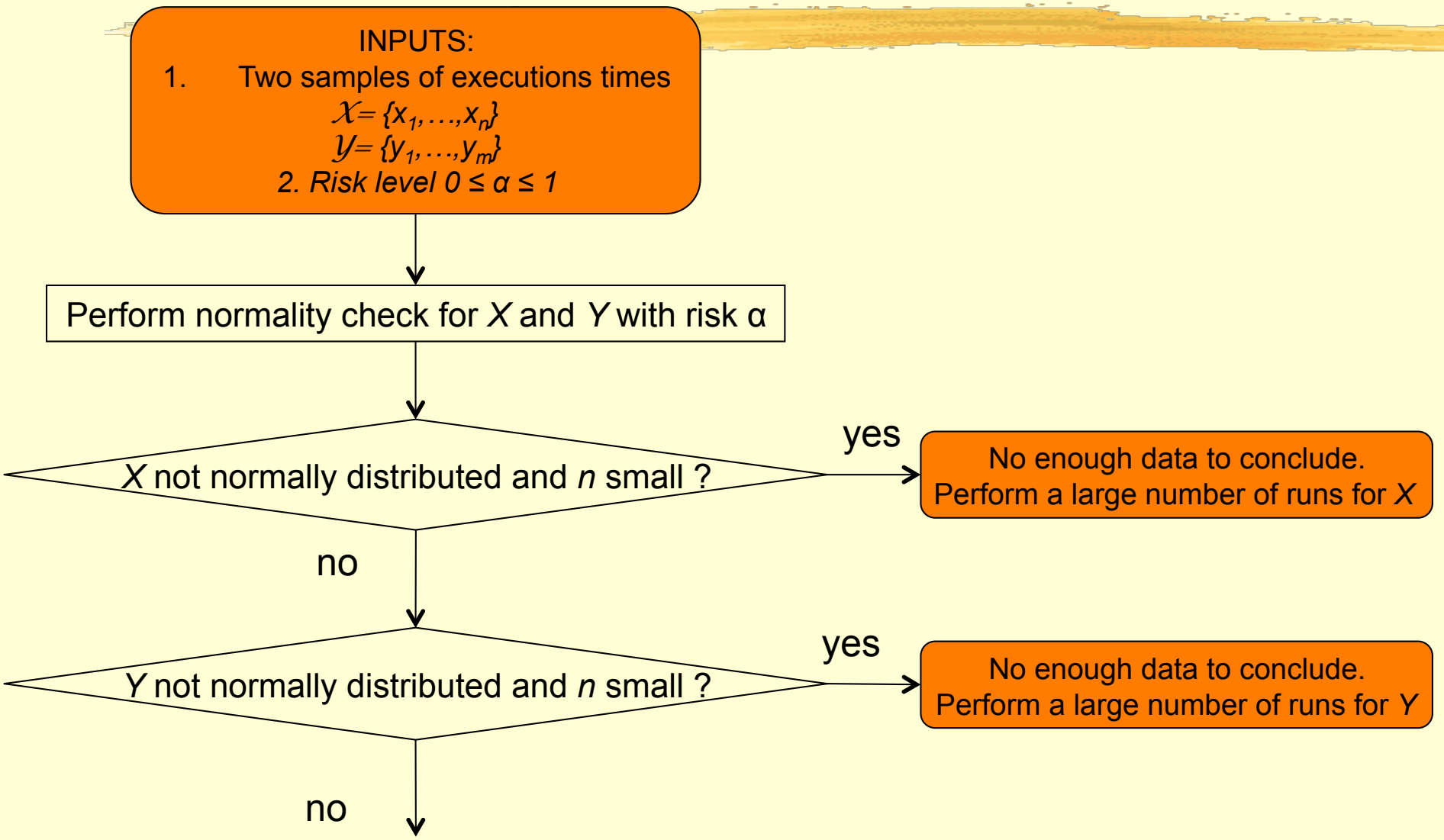
4.1 Statistical significance of the observed speedup of the mean execution time

- The null hypothesis that we want to reject
  - $H_0: \mu_X \leq \mu_Y$
  - This is called the one-sided Student's t-test
- The observation x_i is not associated to the observation y_i
  - The unpaired version of Student's t-test

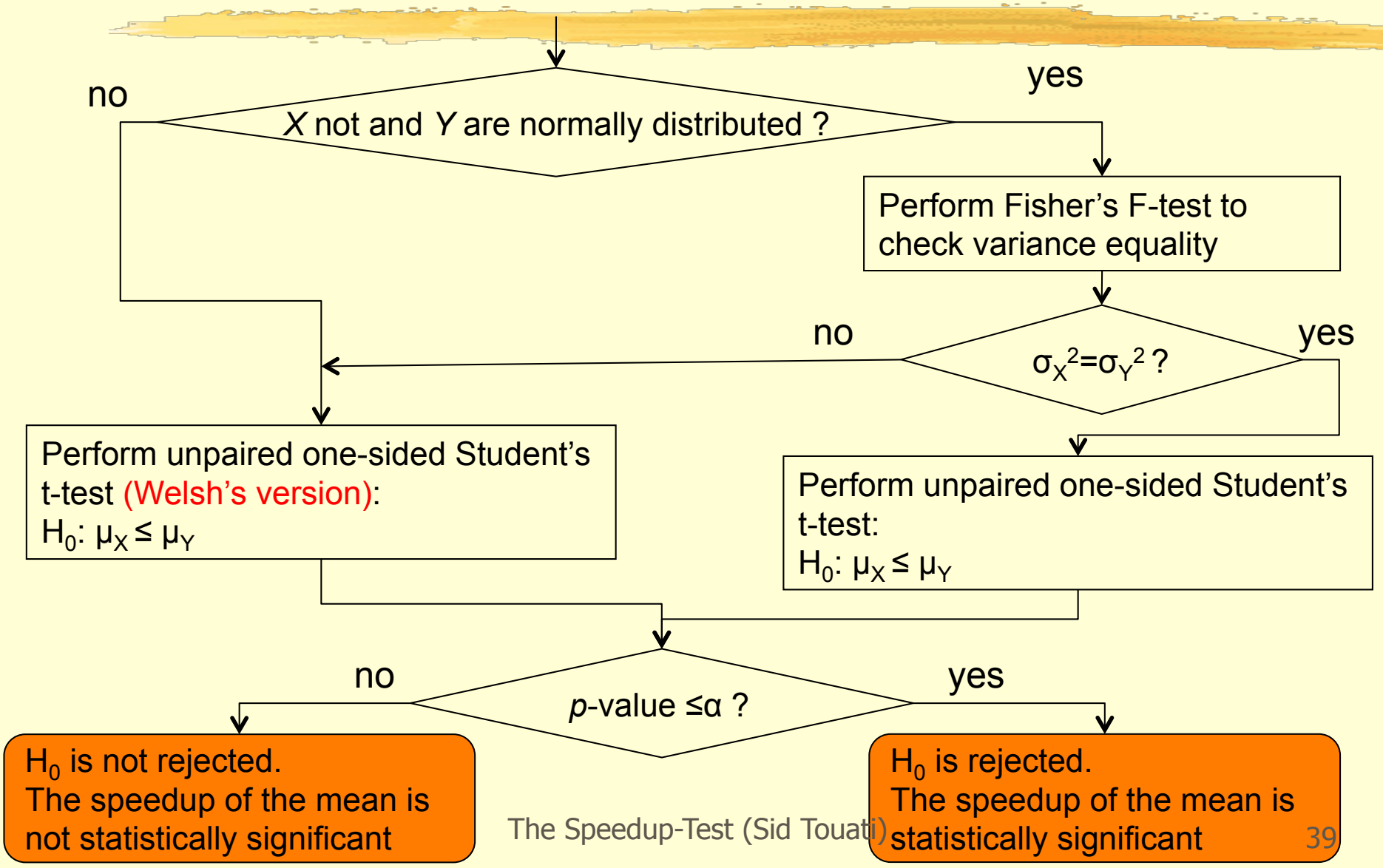
4.1 Statistical significance of the observed speedup of the mean execution time

- The test computes a *p*-value, which is the minimal probability of error if we reject H_0
  - If $\alpha \leq p$-value then H_0 is rejected and we declare that $\mu_X > \mu_Y$
    - Statistical significant speedup of the average execution time
  - Otherwise, H_0 is not rejected. By abuse, we accept H_a.
    - But remind that this does not mean that H_a is true with a probability of $1-\alpha$
    - Statistical insignificant speedup of the average execution time

4.1 Statistical significance of the observed speedup of the mean execution time



4.1 Statistical significance of the observed speedup of the mean execution time



Problem with the average execution time

- The average is sensitive to outliers
- The average may be far from the "feeling" of the observed performance
- The Student's t-test is proved for normally distributed data only
- The median execution time is a better metric for reporting program performance
  - SPEC advocates for it, we also do so

4.2 Statistical significance of the observed speedup of the median execution time

- The Wilcoxon-Mann-Whitney's test allows to compare between two medians
- It also allow to check if P[X>Y] > ½
- The Wilcoxon-Mann-Whitney's test does not require that *X* and *Y* are normally distributed

4.2 Statistical significance of the observed speedup of the median execution time

- The test Wilcoxon-Mann-Whitney's is proved if X and Y fit in the location shift model
  - $X=Y+\Delta$
- This can be checked thanks to the Kolmogov-Smirnov's test
- If X and Y do not fit in the location shift model, then we can still use the Wilcoxon-Mann-Whitney's but the risk level may not be preserved

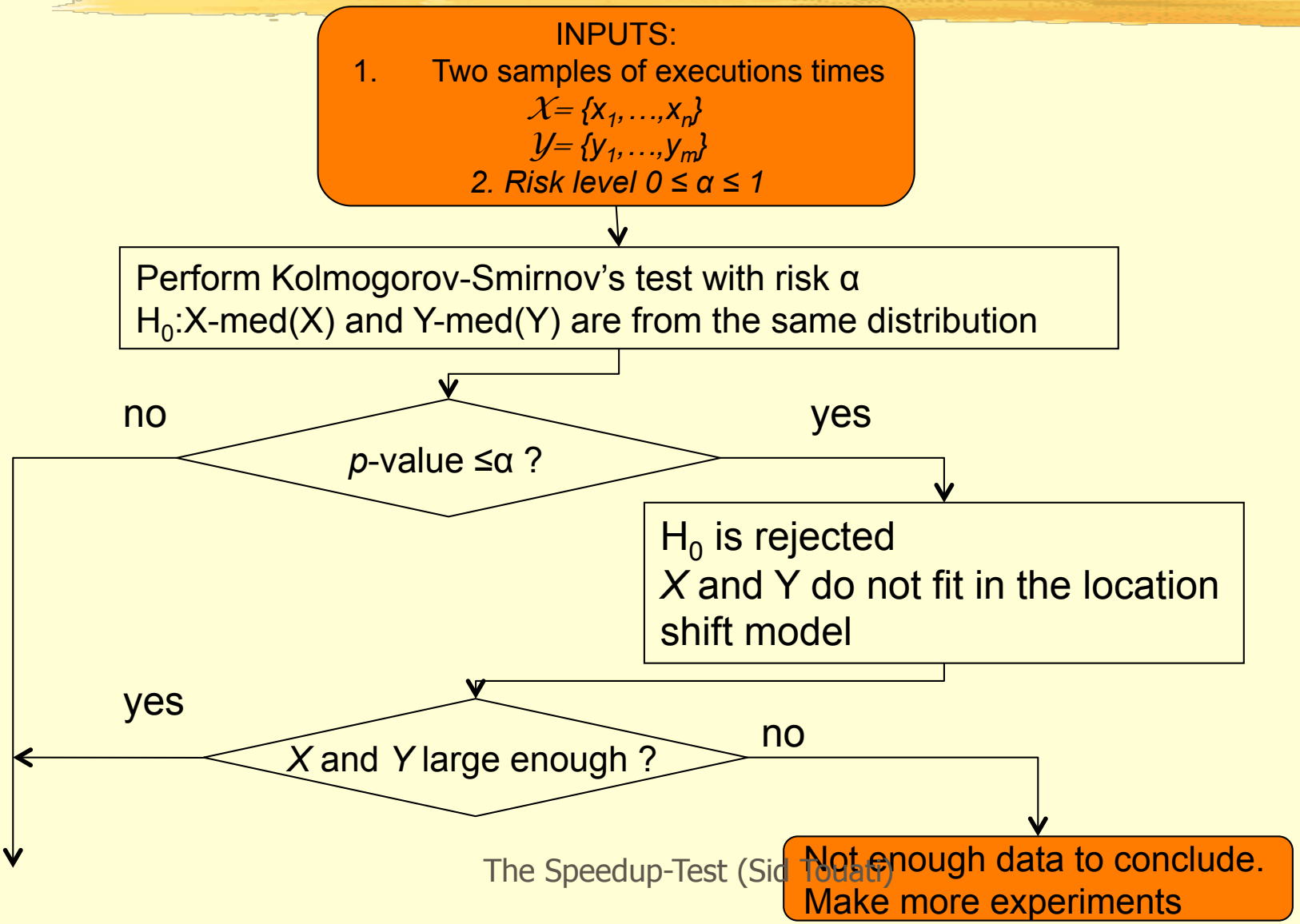
4.2 Statistical significance of the observed speedup of the median execution time

- The Kolmogorov-Smirnov's test checks if two samples are drawn from the same distribution
- Null hypothesis H_0: X-med(X) and Y-med(Y) are drawn from the same distribution
- If *p*-value$\leq \alpha$ then rejects H_0
  - *X* and *Y* do not fit in the location shift model
  - *X* and *Y* must be large enough to use the Wilcoxon-Mann-Whitney's test

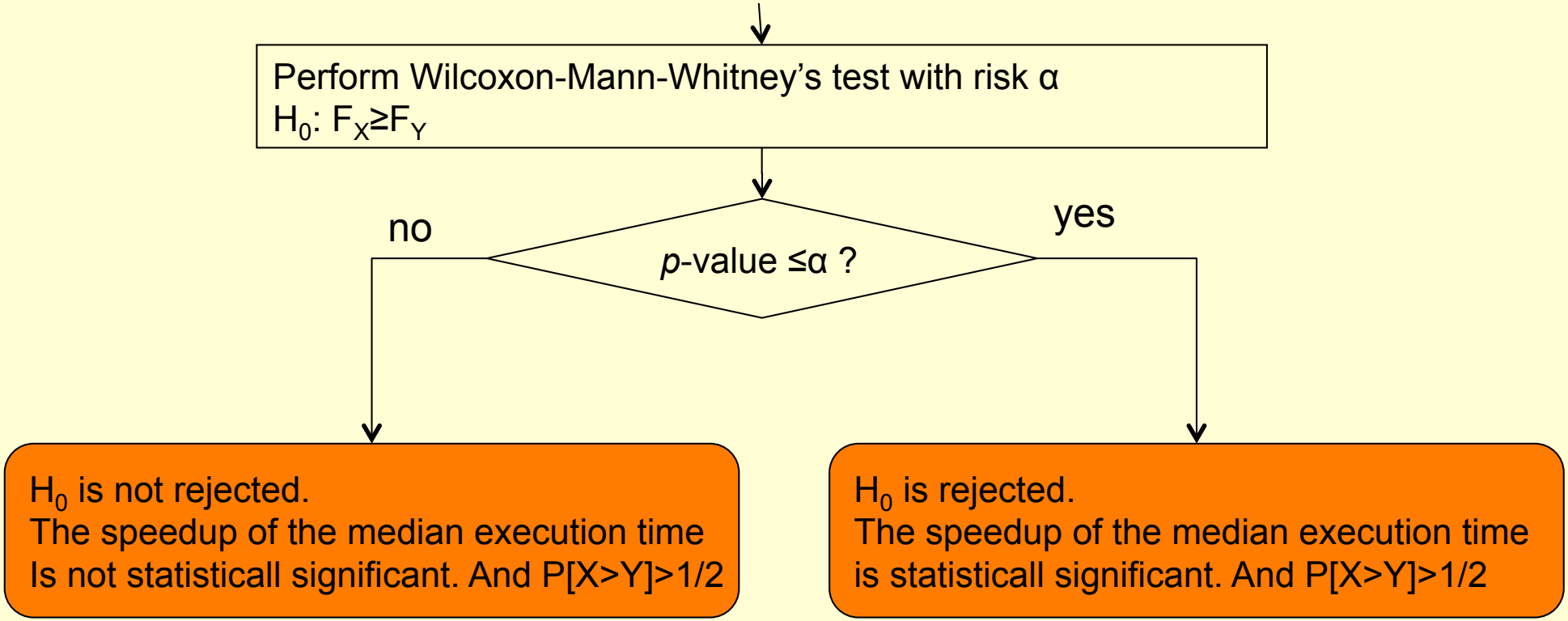
4.2 Statistical significance of the observed speedup of the median execution time

- The Wilcoxon-Mann-Whitney's test compares between F_X and F_Y
- If $F_X \leq F_Y$ then $med(X) \geq med(Y)$
- The null hypothesis that we want to reject
  - $H_0: F_X \geq F_Y$
  - Unpaired and one sided version of the Wilcoxon-Mann-Whitney's test
- If p-value$\leq \alpha$ then rejects H_0
  - Declare that the speedup of the median is statistically significant

4.2 Statistical significance of the observed speedup of the median execution time



4.2 Statistical significance of the observed speedup of the median execution time



Tutorial Outline

1. General introduction
2. Common observed non rigorous experimental methodology
3. Different kinds of observed speedups
4. Statistical significance of the observed speedups
5. Proportion of accelerated programs
6. Conclusion on the Speedup-Test methodology
7. The Speedup-Test Software: tool demo

Proportion of accelerated programs

UNIVERSITÉ DE VERSAILLES SAINT-QUENTIN-EN-YVELINES

- How can we measure the quality of a code optimisation technique ?
- If you consider a set of *b* benchmarks, you may get a speedup in a fraction only, say in *a* benchmarks among *b*
- Usually, we publish only positive results
  - Select the set of benchmarks that exhibit a speedup

Proportion of accelerated programs

- Studying the confidence interval of the fraction *a/b* is a metric for evaluating the quality of a code optimisation technique
- Let consider the random event "The code optimisation has produced a speedup on a benchmark".
  - This event is binomial (Bernoulli) since it has only two possible values: TRUE and FALSE

Proportion of accelerated programs

- In order to be valid, the initial set of *b* benchmarks must be *randomly* selected among a huge number of representative benchmarks
  - Do not consider a manually selected benchmarks (SPEC, and so on)
  - Do not select by yourself the set of benchmmarks
  - You need really to have a random selection of *b* benchmarks

Proportion of accelerated programs

- Having *b* benchmarks, apply the code optimisation method and check with the Speedup-Test how much benchmarks have got a speedup (mean of median).
- The observed fraction of accelerated benchmarks is then *a/b*
- We can compute the confidence interval for this fraction

Proportion of accelerated programs

- By fixing α a risk level, the confidence interval of $C=a/b$ is

$$\left[C - z_{1-\alpha/2} \times \sqrt{\frac{C(1-C)}{b}},\ C + z_{1-\alpha/2} \times \sqrt{\frac{C(1-C)}{b}}\right]$$

- Here $z_{1-\alpha/2}$ represents the value of the quartile of order $1-\alpha/2$ of the standard normal distribution

$$P[N(0,1) > z] = \frac{\alpha}{2}$$

Proportion of accelerated programs

- We can also compute an estimation of the number of *b* benchmarks needed to have an accuracy of *r*% of the confidence interval

$$b \geq (z_{1-\alpha/2})^2 \times \frac{C(1-C)}{r^2}$$

- The confidence interval of the proportion *a/b* isn't valid if *C* is too close from 0 or 1
- A frequently cited rule is that $a - \frac{a^2}{b} > 5$

What is a *reasonable* large sample ?

- In some books devoted to practice, you can read that *n>30* is the size of large samples
- This size limit is arbitrary, it has no mathematical justification
- Indeed, in books devoted to statistics, you would not find an answer of the sample sizes unless you know the distribution function of the data

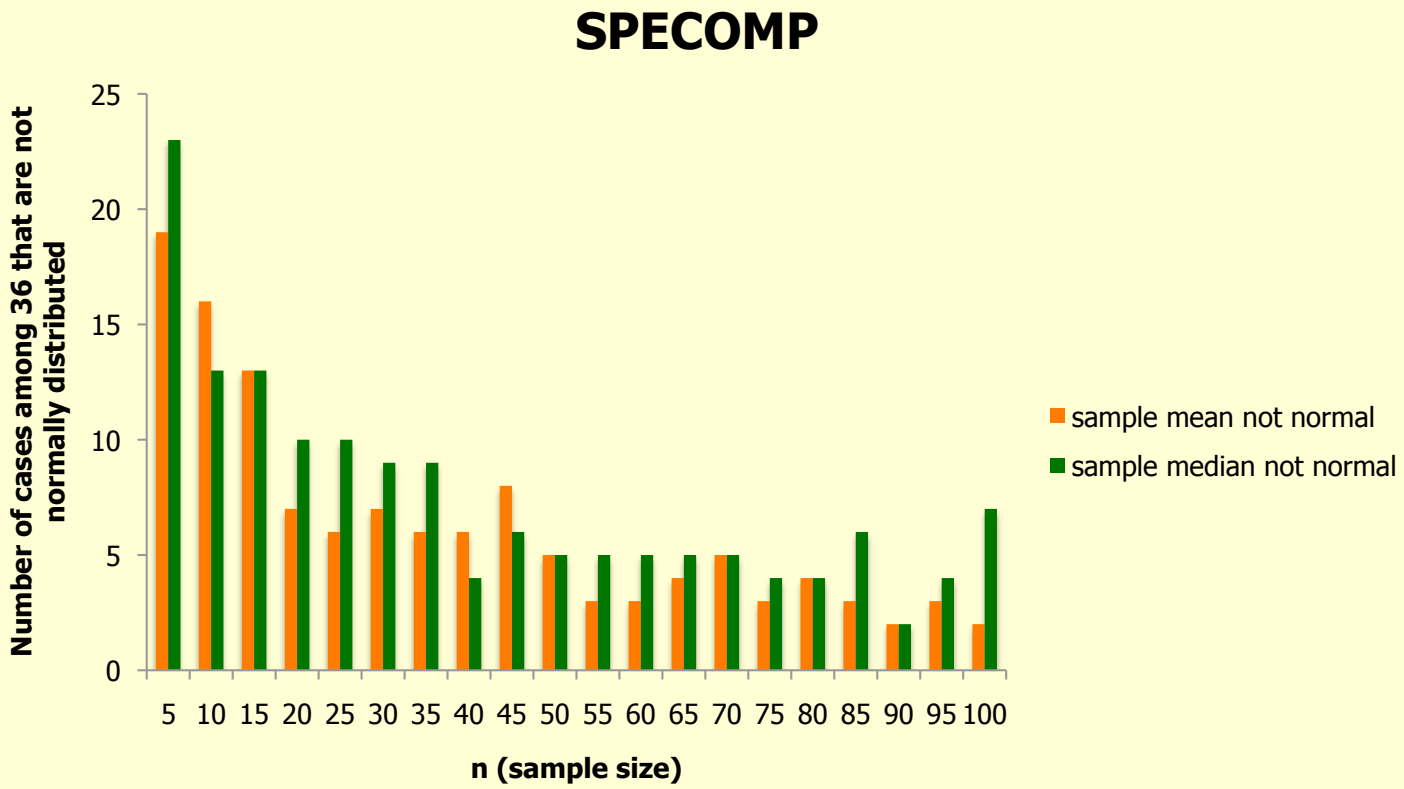
What is a *reasonable* large sample ?

- So from where do we get the constant *n>30 ?*
- From the Student's t-test
  - If *n>30*, the values of the Student's distribution become very close to normal distributions
  - In the old days, manual computation used pre-computed tables printed in documents
  - Nowadays computers are used. So considering *z* values instead of *t* values is out of date

What is a *reasonable* large sample ?

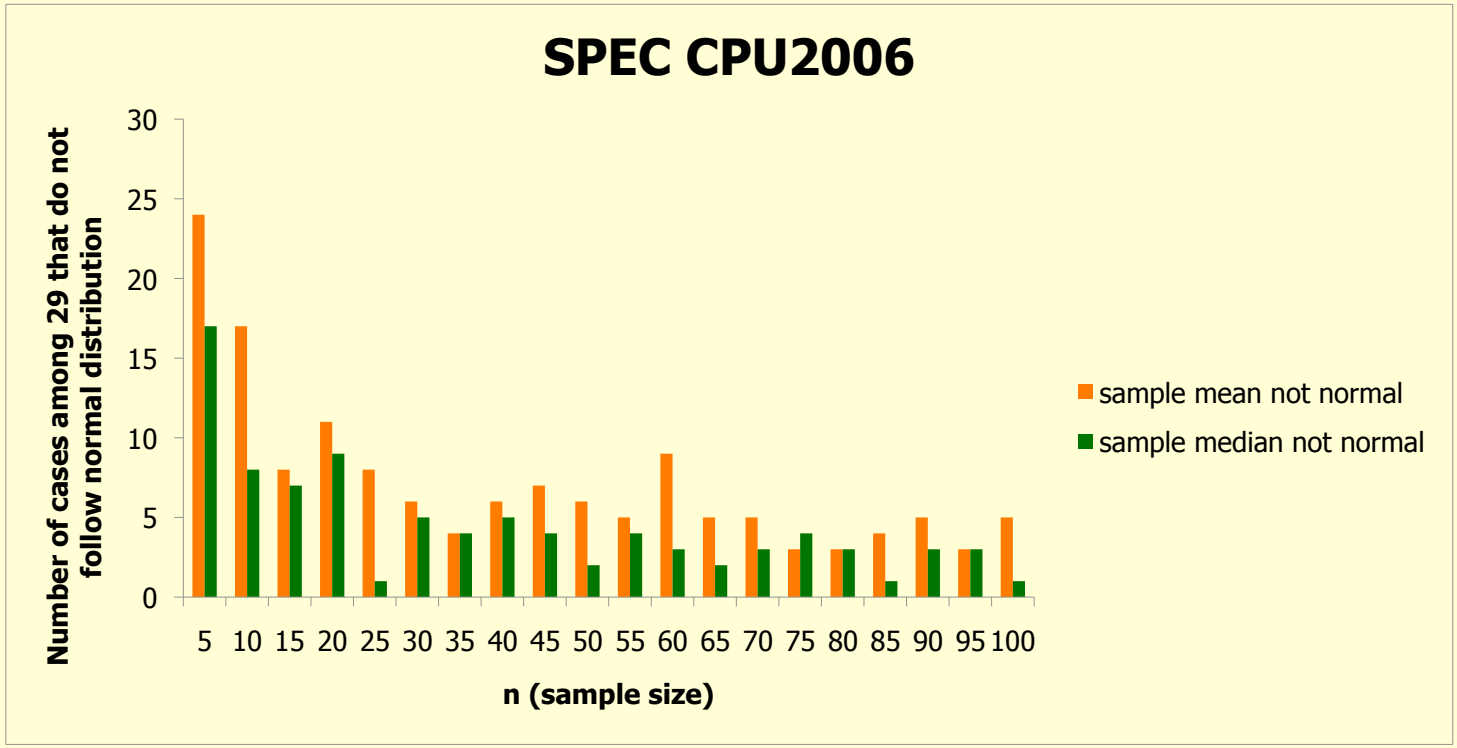
- Is *n>30* relevant in the context of the Speedup-Test ?
- We have made extensive experiments
- SPEC2006, SPEC OMP run thousands of times.
- Goal : compute a value for *n* that makes us observe the central limit theorem.
  - According to CLT, the sample mean and the sample median are normally distributed
  - So we generated multiple samples of different sizes, and we check if this is observable in practice

What is a *reasonable* large sample ?



Experiments conducted on a low overhead machine

What is a *reasonable* large sample ?

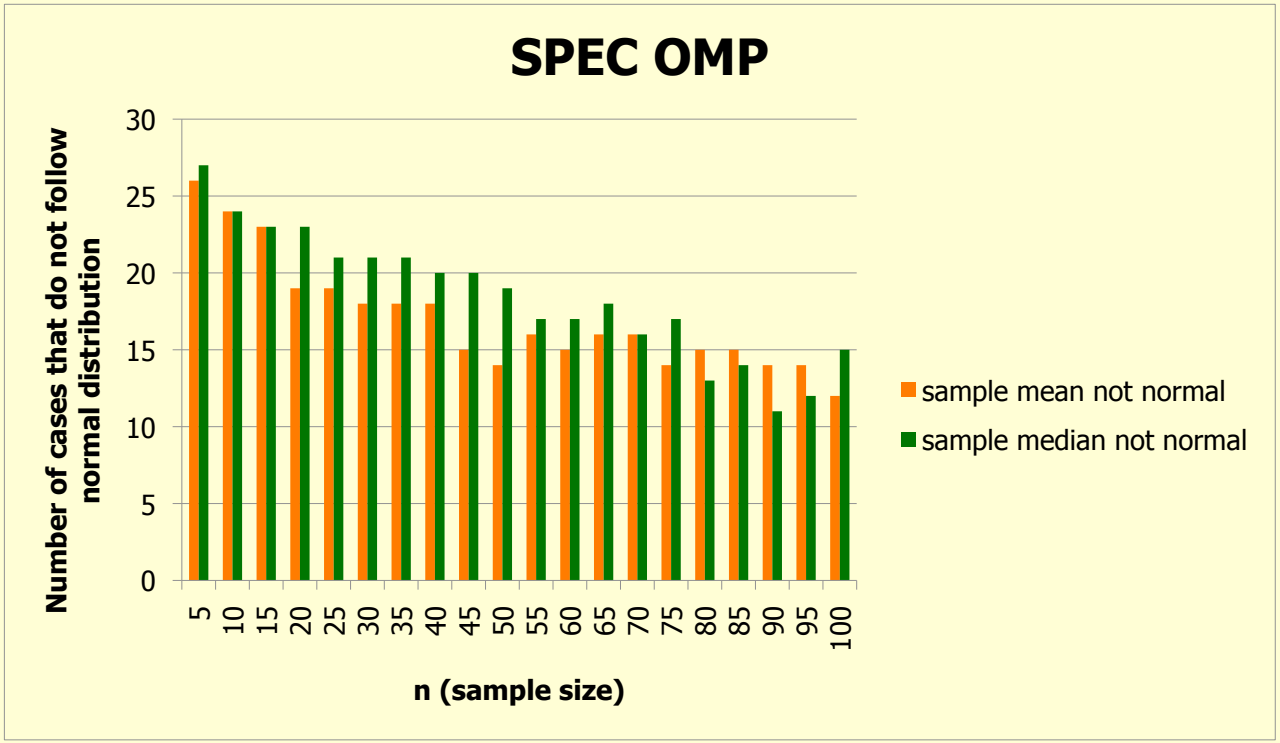


Experiments conducted on a low overhead machine

What is a *reasonable* large sample ?

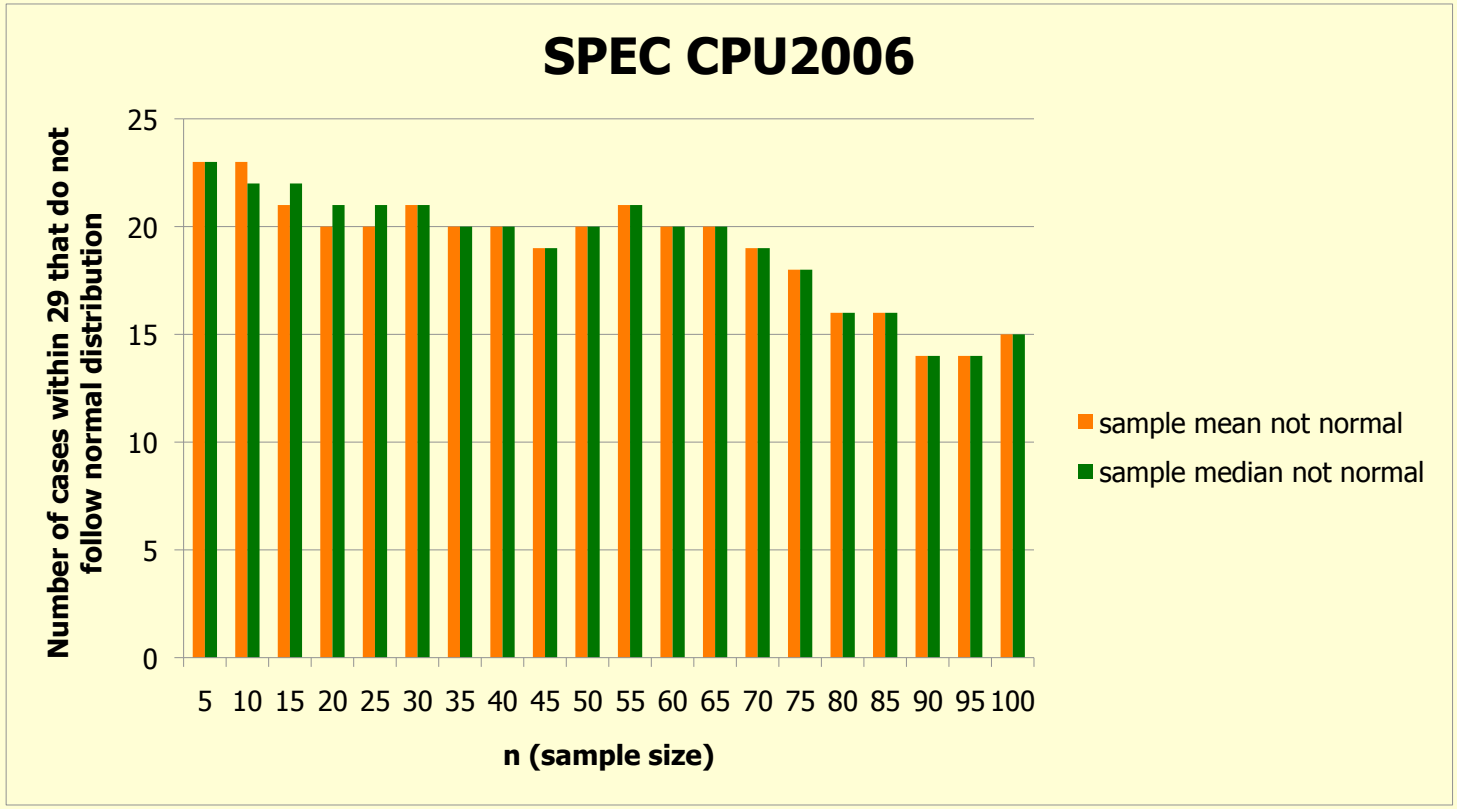
- While we see that n>30 does not guarantee the observation of the CLT, we must be aware that
  - Normality check has a probability of error
  - The observations of the executions times may not be independent
  - The sample size is not large enough
- However, we advocate to use the size limit n>30, n=35 for instance

What is a *reasonable* large sample ?



Experiments conducted on a high overhead machine

What is a *reasonable* large sample ?



Experiments conducted on a high overhead machine

Tutorial Outline

1. General introduction
2. Common observed non rigorous experimental methodology
3. Different kinds of observed speedups
4. Statistical significance of the observed speedups
5. Proportion of accelerated programs
6. Conclusion on the Speedup-Test methodology
7. The Speedup-Test Software: tool demo

Conclusion on the Speedup-Test

- Observe the speedup of the median or the speedup of the average execution time
- Do not remove outliers from the sample
- To declare a speedup of the mean
  - Normality check for small samples
  - Unpaired one sided Student's t-test
- To declare a speedup of the median (preferable)
  - Location shift model for small samples
  - Unpaired one sided Wilcoxon-Mann-Whitney's test

Conclusion on the Speedup-Test

- Overall speedup of the set of benchmarks
  - Do not compute the mean of speedups
  - Better to use the given equations for reporting overall gains or overall speedups
- Confidence interval of proportions
  - Randomly select *b* benchmarks among a huge number of representative benchmarks
  - Calculate how many benchmarks have got a speedup (mean or median)
  - Compute the confidence interval of *a/b*

Conclusion on the Speedup-Test

- The preferable size of the sample cannot be determined precisely if the data distribution is not known
- In practice, *n=35* defines the size of large samples
  - But we have seen that this is not necessarily sufficient (SPEC2006, SPEC OMP)
  - Independence of the observations

Tutorial Outline

1. General introduction
2. Common observed non rigorous experimental methodology
3. Different kinds of observed speedups
4. Statistical significance of the observed speedups
5. Proportion of accelerated programs
6. Conclusion on the Speedup-Test methodology
7. The Speedup-Test Software: tool demo

The Speedup-Test software

- Implements all the protocols to analyse the statistical significance of the observed speedups
- Input: collected observations of executions times
- Output: analysis report, confidence levels, warnings

Installation

- Install the R software
  - Open source, available under many linux distributions, also under MacOS
- Download Speedup-Test from

  http://hal.inria.fr/inria-00443839
  - Two scripts (bash and R scripts)
  - Must be copied in your PATH

Input configuration file (CSV)

```
Name           ,   Sample1,          Sample2,          ConfLevel, Coef
"First bench",   "bench1.data.1", "bench1.data.2",NA,          NA
"Second bench",  "bench2.data.1", "bench2.data.2",NA,          NA
"Third bench",   "bench3.data.1", "bench3.data.2",NA,          NA
"Fourth bench",  "bench4.data.1", "bench4.data.2",NA,          NA
```

Command line

```
SpeedUpTest.sh config.csv
```

Command line options

- `--conf-level #value`
- `--weight custom|equal|fraction`
- `--precision #value`
- `-o outputprefix`

Outputs

- `outputfile.report`
  - Overall speedup and gain
  - Proportion confidence interval
- `outputfile.out`
  - For each benchmark, report the observed speedups (min, mean and median), its statistical significance (TRUE, FALSE), confidence levels, coefficients (weights)

Outputs

- `outputfile.warning`
  - Warnings on the rigor of the statistics declared in outputfile.out
- `outputfile.error`
  - Bad behaviour of the Speedup-Test software