CS 147:
Computer Systems Performance Analysis
Ratio Games and Introduction to Experimental Design

CS 147:
Computer Systems Performance Analysis
Ratio Games and Introduction to Experimental Design

Overview

Overview

Ratio Games
How to Lie
Strategies for Winning
Fair Analysis

Experimental Design
Introduction
2^k Designs

Ratio Games

- Choosing a base system
- Using ratio metrics
- Relative performance enhancement
- Ratio games with percentages
- Strategies for winning a ratio game
- Correct analysis of ratios

Ratio Games
- Choosing a base system
- Using ratio metrics
- Relative performance enhancement
- Ratio games with percentages
- Strategies for winning a ratio game
- Correct analysis of ratios

Choosing a Base System

- Run workloads on two systems
- Normalize performance to chosen system
- Take average of ratios
- *Presto*: you control what's best

Choosing a Base System

- Run workloads on two systems
- Normalize performance to chosen system
- Take average of ratios
- *Presto*: you control what's best

Example of Choosing a Base System

- (Carefully) selected Ficus results:

	1	2	1/2	2/1
cp	231.8	168.6	1.37	0.73
rcp	260.6	338.3	0.77	1.30
Mean	246.2	253.45	1.07	1.02

Example of Choosing a Base System

- (Carefully) selected Ficus results:

	1	2	1/2	2/1
cp	231.8	168.6	1.37	0.73
rcp	260.6	338.3	0.77	1.30
Mean	246.2	253.45	1.07	1.02

Here, the mean running time on two replicas is worse. But by choosing the appropriate base, I can make a single replica 7% slower, or I can make two replicas 2% slower (i.e., a single replica is 2% faster).

Why Does This Work?

- Expand the arithmetic:

$$R_{a;b} = \frac{y_a}{y_b} \qquad R_{b;b} = 1.0$$

$$\begin{aligned} P_{a;b} &= \frac{1}{n}\sum R_{i;a;b} = \frac{1}{n}\left(\frac{y_{0;a}}{y_{0;b}} + \frac{y_{1;a}}{y_{1;b}} + \cdots\right) \\ &\neq \frac{\frac{1}{n}\sum y_{i;a}}{\frac{1}{n}\sum y_{1;b}} \neq \frac{1}{P_{b;a}} \end{aligned}$$

Why Does This Work?
Expand the arithmetic:
$$R_{a;b} = \frac{y_a}{y_b} \qquad R_{b;b} = 1.0$$
$$\begin{aligned} P_{a;b} &= \frac{1}{n}\sum R_{i;a;b} = \frac{1}{n}\left(\frac{y_{0;a}}{y_{0;b}} + \frac{y_{1;a}}{y_{1;b}} + \cdots\right) \\ &\neq \frac{\frac{1}{n}\sum y_{i;a}}{\frac{1}{n}\sum y_{1;b}} \neq \frac{1}{P_{b;a}} \end{aligned}$$

R is the performance ratio of the overall test, i.e., the total time of all tests (equivalently, their average, assuming paired tests). *P* is the average of ratios.

Using Ratio Metrics

- Pick a metric that is itself a ratio
  - E.g., power = throughput $\div$ response time
  - Or cost/performance
- Handy because division is “hidden”

Using Ratio Metrics
- Pick a metric that is itself a ratio
  - E.g., power = throughput ÷ response time
  - Or cost/performance
- Handy because division is “hidden”

This is subtler because of the hidden division.

Relative Performance Enhancement

- Compare systems with incomparable bases
- Turn into ratios
- Example: compare Ficus 1 vs. 2 replicas with UFS vs. NFS (1 run on chosen day):

	"cp" Time		Ratio
Ficus 1 vs. 2	197.4	246.6	1.25
UFS vs. NFS	178.7	238.3	1.33

- "Proves" adding Ficus replica costs less than going from UFS to NFS

Relative Performance Enhancement

- Compare systems with incomparable bases
- Turn into ratios
- Example: compare Ficus 1 vs. 2 replicas with UFS vs. NFS (1 run on chosen day):

	"cp" Time		Ratio
Ficus 1 vs. 2	197.4	246.6	1.25
UFS vs. NFS	178.7	238.3	1.33

- "Proves" adding Ficus replica costs less than going from UFS to NFS

Ratio Games with Percentages

- Percentages are inherently ratios
  - But disguised
  - So great for ratio games
- Example: Passing tests

Test	A Runs	A Passes	A %	B Runs	B Passes	B %
1	300	60	20	32	8	25
2	50	2	4	500	40	8
Total	350	62	18	532	48	9

- A is worse, but looks better in total line!

Ratio Games with Percentages

- Percentages are inherently ratios
  - But disguised
  - So great for ratio games
- Example: Passing tests

Test	A Runs	A Passes	A %	B Runs	B Passes	B %
1	300	60	20	32	8	25
2	50	2	4	500	40	8
Total	350	62	18	532	48	9

- A is worse, but looks better in total line!

More on Percentages

- Psychological impact
  - 1000% sounds bigger than 10-fold (or 11-fold)
  - Great when both original and final performance are lousy
    - E.g., salary went from $40 to $80 per week
- Small sample sizes can generate big lies
  - “83% of dentists surveyed recommend Crest”
  - (We asked 6 dentists; 5 liked Crest)
- Base should be initial, not final value
  - E.g., price can't drop 400%

More on Percentages

- Psychological impact
  - 1000% sounds bigger than 10-fold (or 11-fold)
  - Great when both original and final performance are lousy
    - E.g., salary went from $40 to $80 per week
- Small sample sizes can generate big lies
  - “83% of dentists surveyed recommend Crest”
  - (We asked 6 dentists; 5 liked Crest)
- Base should be initial, not final value
  - E.g., price can't drop 400%

Can You Win the Ratio Game?

- If one system is better by all measures, a ratio game won't work
  - But recall percent-passes example
  - And selecting the base lets you change the magnitude of the difference
- If each system wins on some measures, ratio games might be possible (but no promises)
  - May have to try all bases

Can You Win the Ratio Game?
- If one system is better by all measures, a ratio game won't work
  - But recall percent-passes example
  - And selecting the base lets you change the magnitude of the difference
- If each system wins on some measures, ratio games might be possible (but no promises)
  - May have to try all bases

How to Win Your Ratio Game

- For LB metrics, use your system as the base
- For HB metrics, use the other as a base
- If possible, adjust lengths of benchmarks
  - Elongate when your system performs best
  - Short when your system is worst
  - This gives greater weight to your strengths

How to Win Your Ratio Game

- For LB metrics, use your system as the base
- For HB metrics, use the other as a base
- If possible, adjust lengths of benchmarks
  - Elongate when your system performs best
  - Short when your system is worst
  - This gives greater weight to your strengths

Correct Analysis of Ratios

- Previously covered in lecture #5
- Generally, harmonic or geometric means are appropriate
  - Or use only the raw data

Correct Analysis of Ratios

- Previously covered in lecture #5
- Generally, harmonic or geometric means are appropriate
  - Or use only the raw data

Introduction To Experimental Design

- You know your metrics
- You know your factors
- You know your levels
- You've got your instrumentation and test loads
- Now what?

Introduction To Experimental Design

- You know your metrics
- You know your factors
- You know your levels
- You've got your instrumentation and test loads
- Now what?

Goals in Experiment Design

- Obtain maximum information with minimum work
  - Typically meaning minimum number of experiments
- More experiments aren't better if you're the one who has to perform them
- Well-designed experiments are also easier to analyze

Goals in Experiment Design
- Obtain maximum information with minimum work
  - Typically meaning minimum number of experiments
- More experiments aren't better if you're the one who has to perform them
- Well-designed experiments are also easier to analyze

Experimental Replications

- System under study will be run with varying levels of different factors, potentially with differing workloads
- Run with particular set of levels and other inputs is a *replication*
- Often, need to do multiple replications with each set of levels and other inputs
  - Usually necessary for statistical validation

Experimental Replications

- System under study will be run with varying levels of different factors, potentially with differing workloads
- Run with particular set of levels and other inputs is a *replication*
- Often, need to do multiple replications with each set of levels and other inputs
  - Usually necessary for statistical validation

Interacting Factors

- Some factors have completely independent effects
  - Double the factor's level, halve the response, regardless of other factors
- But effects of some factors depends on values of others
  - Called *interacting* factors
- Presence of interacting factors complicates experimental design

Interacting Factors
- Some factors have completely independent effects
  - Double the factor's level, halve the response, regardless of other factors
- But effects of some factors depends on values of others
  - Called *interacting* factors
- Presence of interacting factors complicates experimental design

The Basic Problem in Designing Experiments

- You've chosen some number of factors
  - May or may not interact
- How to design experiment that captures full range of levels?
  - Want minimum amount of work
- Which combination or combinations of levels (of factors) do you measure?

The Basic Problem in Designing Experiments

- You've chosen some number of factors
  - May or may not interact
- How to design experiment that captures full range of levels?
  - Want minimum amount of work
- Which combination or combinations of levels (of factors) do you measure?

Common Mistakes in Experimentation

- Ignoring experimental error
- Uncontrolled parameters
- Not isolating effects of different factors
- One-factor-at-a-time experimental designs
- Interactions ignored
- **Designs require too many experiments**

Common Mistakes in Experimentation

- Ignoring experimental error
- Uncontrolled parameters
- Not isolating effects of different factors
- One-factor-at-a-time experimental designs
- Interactions ignored
- **Designs require too many experiments**

Types of Experimental Designs

- Simple designs
- Full factorial design
- Fractional factorial design

Types of Experimental Designs

- Simple designs
- Full factorial design
- Fractional factorial design

This is all we'll cover, but there are other possibilities.

Simple Designs

- Vary one factor at a time
- For k factors with i^{th} factor having n_i levels, number of experiments needed is:

$$n = 1 + \sum_{i=1}^{k} (n_i - 1)$$

- Assumes factors don't interact
  - Even then, more effort than required
- Don't use it, usually

Simple Designs
- Vary one factor at a time
- For k factors with i^{th} factor having n_i levels, number of experiments needed is:
$$n = 1 + \sum_{i=1}^{k} (n_i - 1)$$
- Assumes factors don't interact
  - Even then, more effort than required
- Don't use it, usually

Full Factorial Designs

- Test every possible combination of factors' levels
- For k factors with i^{th} factor having n_i levels:

$$n = \prod_{i=1}^{k} n_i$$

- Captures full information about interaction
- But a huge amount of work

Full Factorial Designs

- Test every possible combination of factors' levels
- For k factors with i^{th} factor having n_i levels:

$$n = \prod_{i=1}^{k} n_i$$

- Captures full information about interaction
- But a huge amount of work

Reducing the Work in Full Factorial Designs

- Reduce number of levels per factor
  - Generally good choice
  - Especially if you know which factors are most important
    - Use more levels for those
- Reduce number of factors
  - But don't drop important ones!
- Use fractional factorial designs

Reducing the Work in Full Factorial Designs
- Reduce number of levels per factor
  - Generally good choice
  - Especially if you know which factors are most important
    - Use more levels for those
- Reduce number of factors
  - But don't drop important ones!
- Use fractional factorial designs

Fractional Factorial Designs

- Only measure some combination of levels of the factors
- Must design carefully to best capture any possible interactions
- Less work, but more chance of inaccuracy
- Especially useful if some factors are known to not interact
- Covered later

Fractional Factorial Designs

- Only measure some combination of levels of the factors
- Must design carefully to best capture any possible interactions
- Less work, but more chance of inaccuracy
- Especially useful if some factors are known to not interact
- Covered later

2^k Factorial Designs

- Used to determine effect of k factors
  - Each with two alternatives or levels
- Often used as preliminary to larger performance study
  - Each factor measured at its maximum and minimum level
  - Might offer insight on importance and interaction of various factors

2^k Factorial Designs

- Used to determine effect of k factors
  - Each with two alternatives or levels
- Often used as preliminary to larger performance study
  - Each factor measured at its maximum and minimum level
  - Might offer insight on importance and interaction of various factors

Unidirectional Effects

- Effects that only increase as level of a factor increases
  - Or vice versa
- If system known to have unidirectional effects, 2^k factorial design at minimum and maximum levels is useful
- Shows whether factor has significant effect

Unidirectional Effects

- Effects that only increase as level of a factor increases
  - Or vice versa
- If system known to have unidirectional effects, 2^k factorial design at minimum and maximum levels is useful
- Shows whether factor has significant effect

2^2 Factorial Designs

- Two factors with two levels each
- Simplest kind of factorial experiment design
- Concepts developed here generalize
- Regression can easily be used

2^2 Factorial Designs

- Two factors with two levels each
- Simplest kind of factorial experiment design
- Concepts developed here generalize
- Regression can easily be used

2^2 Factorial Design Example

- Consider parallel operating system
- Goal is fastest possible completion of a given program
- Quality usually expressed as *speedup*
- We'll use *runtime* as metric (simpler but equivalent)

2^2 Factorial Design Example

- Consider parallel operating system
- Goal is fastest possible completion of a given program
- Quality usually expressed as *speedup*
- We'll use *runtime* as metric (simpler but equivalent)

Factors and Levels for Parallel OS

- First factor: number of CPUs
  - Vary between 8 and 64
- Second factor: use of dynamic load management
  - Migrates work between nodes as load changes
- Other factors possible, but ignore them for now

Factors and Levels for Parallel OS
First factor: number of CPUs
Vary between 8 and 64
Second factor: use of dynamic load management
Migrates work between nodes as load changes
Other factors possible, but ignore them for now

Defining Variables for 2^2 Factorial OS Example

$$x_A = \begin{cases} \text{-1 if 8 nodes} \\ \text{+1 if 64 nodes} \end{cases}$$

$$x_B = \begin{cases} \text{-1 if no dynamic load management} \\ \text{+1 if dynamic load management} \end{cases}$$

Defining Variables for 2^2 Factorial OS Example

$$x_A = \begin{cases} \text{-1 if 8 nodes} \\ \text{+1 if 64 nodes} \end{cases}$$

$$x_B = \begin{cases} \text{-1 if no dynamic load management} \\ \text{+1 if dynamic load management} \end{cases}$$

Sample Data for Parallel OS

Single runs of one benchmark (in seconds):

	8 Nodes	64 Nodes
NO DLM	820	217
DLM	776	197

Sample Data for Parallel OS

Single runs of one benchmark (in seconds):

	8 Nodes	64 Nodes
NO DLM	820	217
DLM	776	197

Regression Model for Example

- $y = q_0 + q_A x_A + q_B x_B + q_{AB} x_A x_B$
- Note that model is nonlinear!

$$\begin{aligned}
820 &= q_0 - q_A - q_B + q_{AB} \\
217 &= q_0 + q_A - q_B - q_{AB} \\
776 &= q_0 - q_A + q_B - q_{AB} \\
197 &= q_0 + q_A + q_B + q_{AB}
\end{aligned}$$

Regression Model for Example

- $y = q_0 + q_A x_A + q_B x_B + q_{AB} x_A x_B$
- Note that model is nonlinear!

$$\begin{aligned}
820 &= q_0 - q_A - q_B + q_{AB} \\
217 &= q_0 + q_A - q_B - q_{AB} \\
776 &= q_0 - q_A + q_B - q_{AB} \\
197 &= q_0 + q_A + q_B + q_{AB}
\end{aligned}$$

Solving the Equations

- 4 equations in 4 unknowns
- $q_0 = 502.5$
- $q_A = -295.5$
- $q_B = -16$
- $q_{AB} = 6$
- So $y = 502.5 - 295.5x_A - 16x_B + 6x_Ax_B$

Solving the Equations

- 4 equations in 4 unknowns
- $q_0 = 502.5$
- $q_A = -295.5$
- $q_B = -16$
- $q_{AB} = 6$
- So $y = 502.5 - 295.5x_A - 16x_B + 6x_Ax_B$

The Sign Table Method

- Write problem in tabular form:

I	A	B	AB	y
1	-1	-1	1	820
1	1	-1	-1	217
1	-1	1	-1	776
1	1	1	1	197
2010	-1182	-64	24	Total
502.5	-295.5	-16	6	Total/4

The Sign Table Method
Write problem in tabular form:
I A B AB y
1 -1 -1 1 820
1 1 -1 -1 217
1 -1 1 -1 776
1 1 1 1 197
2010 -1182 -64 24 Total
502.5 -295.5 -16 6 Total/4

Allocation of Variation for 2^2 Model

- Calculate the sample variance of y:

$$s_y^2 = \frac{\sum_{i=1}^{2^2}(y_i - \overline{y})^2}{2^2 - 1}$$

- Numerator is SST: total variation

$$\text{SST} = 2^2 q_A^2 + 2^2 q_B^2 + 2^2 q_{AB}^2$$

- SST explains causes of variation in y

Allocation of Variation for 2^2 Model
- Calculate the sample variance of y:
$$s_y^2 = \frac{\sum_{i=1}^{2^2}(y_i - \overline{y})^2}{2^2 - 1}$$
- Numerator is SST: total variation
$$\text{SST} = 2^2 q_A^2 + 2^2 q_B^2 + 2^2 q_{AB}^2$$
- SST explains causes of variation in y

Derivation of SST is in book, pp. 287–288. Note that q_0 is exactly the sample mean $\overline{y}$. Thus, $y_i - \overline{y} = q_A x_{Ai} + q_B x_{Bi} + q_{AB} x_{Ai} x_{Bi}$. Squaring the latter gives the squares of the individual terms, plus product terms—but the product terms sum to zero because the columns in the sign matrix are orthogonal.

Terms in the SST

- $2^2 q_A^2$ is variation explained by effect of A: SSA
- $2^2 q_B^2$ is variation explained by effect of B: SSB
- $2^2 q_{AB}^2$ is variation explained by interaction between A and B: SSAB
- SST = SSA + SSB + SSAB
- In each case, divide SSx by SST to get percent of variation explained by that factor
  - Useful for deciding which factors are important

Terms in the SST
- $2^2 q_A^2$ is variation explained by effect of A: SSA
- $2^2 q_B^2$ is variation explained by effect of B: SSB
- $2^2 q_{AB}^2$ is variation explained by interaction between A and B: SSAB
- SST = SSA + SSB + SSAB
- In each case, divide SSx by SST to get percent of variation explained by that factor
  - Useful for deciding which factors are important

Note that *variation* is not *variance*; computing contribution of each factor to variance is hard.

Variations in Our Example

- SST = 350449
- SSA = 349281
- SSB = 1024
- SSAB = 144
- Now easy to calculate fraction of total variation caused by each effect:
  - Fraction explained by A is 99.67%
  - Fraction explained by B is 0.29%
  - Fraction explained by interaction of A and B is 0.04%
- So almost all variation comes from number of nodes
- If you want to run faster, apply more nodes, don't turn on dynamic load management

Variations in Our Example
- SST = 350449
- SSA = 349281
- SSB = 1024
- SSAB = 144
- Now easy to calculate fraction of total variation caused by each effect:
  - Fraction explained by A is 99.67%
  - Fraction explained by B is 0.29%
  - Fraction explained by interaction of A and B is 0.04%
- So almost all variation comes from number of nodes
- If you want to run faster, apply more nodes, don't turn on dynamic load management

In this simple example, the same conclusion could have been drawn simply by observing the numbers. But that's not always the case.

General 2^k Factorial Designs

- Used to explain effects of k factors, each with two alternatives or levels
- 2^2 factorial designs are a special case
- Same methods extend to more general case
- Many more interactions between pairs (and trios, etc.) of factors

General 2^k Factorial Designs

- Used to explain effects of k factors, each with two alternatives or levels
- 2^2 factorial designs are a special case
- Same methods extend to more general case
- Many more interactions between pairs (and trios, etc.) of factors

Sample 2^3 Experiment

- Sign table columns A, B, C are binary count; interactions are products of appropriate columns:

y	I	A	B	C	AB	AC	BC	ABC
14	1	-1	-1	-1	1	1	1	-1
22	1	1	-1	-1	-1	-1	1	1
10	1	-1	1	-1	-1	1	-1	1
34	1	1	1	-1	1	-1	-1	-1
46	1	-1	-1	1	1	-1	-1	1
58	1	1	-1	1	-1	1	-1	-1
50	1	-1	1	1	-1	-1	1	-1
86	1	1	1	1	1	1	1	1
T/8	40	10	5	20	5	2	3	1
%		18	4.4	71	4.4	0.7	1.6	0.2

- SST = 564

Sample 2^3 Experiment

- Sign table columns A, B, C are binary count; interactions are products of appropriate columns:

y	I	A	B	C	AB	AC	BC	ABC
14	1	-1	-1	-1	1	1	1	-1
22	1	1	-1	-1	-1	-1	1	1
10	1	-1	1	-1	-1	1	-1	1
34	1	1	1	-1	1	-1	-1	-1
46	1	-1	-1	1	1	-1	-1	1
58	1	1	-1	1	-1	1	-1	-1
50	1	-1	1	1	-1	-1	1	-1
86	1	1	1	1	1	1	1	1
T/8	40	10	5	20	5	2	3	1
%		18	4.4	71	4.4	0.7	1.6	0.2

- SST = 564