Experimental Design

Samuel Kounev

"The fundamental principle of science, the definition almost, is this: the sole test of the validity of any idea is experiment."

-- Richard P. Feynman

References

- „Measuring Computer Performance – A Practitioner's Guide" by David J. Lilja, Cambridge University Press, New York, NY, 2000, ISBN 0-521-64105-5
- The supplemental teaching materials provided at http://www.arctic.umn.edu/perf-book/ by David J. Lilja

Roadmap

- Goals
- Terminology
- Two-factor full factorial designs
  - 2-factor ANOVA
- General *m*-factor full factorial designs
  - *m*-factor ANOVA
- n2^m factorial designs
- Fractional factorial designs
  - Plackett and Burman designs
- Case Studies

3

Recall: One-Factor ANOVA

- Separates total variation observed in a set of measurements into:
  1. Variation within individual systems
     - Due to random measurement errors
  2. Variation between systems
     - Due to real differences + random errors
- Is variation(2) statistically > variation(1)?
- *One-factor experimental design*

4

One-Factor ANOVA Summary

Variation	Alternatives	Error	Total
Sum of squares	SSA	SSE	SST
Deg freedom	$k-1$	$k(n-1)$	$kn-1$
Mean square	$s_a^2 = SSA/(k-1)$	$s_e^2 = SSE/[k(n-1)]$	
Computed F	s_a^2/s_e^2		
Tabulated F	$F_{[1-\alpha;(k-1),k(n-1)]}$		

5

Generalized Design of Experiments

- Goals
  - Isolate effects of each input variable
  - Determine effects of interactions
  - Determine magnitude of experimental error
  - Obtain maximum information for given effort
- Basic idea
  - Expand 1-factor ANOVA to *m* factors

6

Terminology

- **Response variable**
  - Measured output value, e.g. total execution time
- **Factors**
  - Input variables that can be changed, e.g. cache size, clock rate, bytes transmitted.
- **Levels**
  - Specific values of factors (inputs), continuous (e.g. ~bytes) or discrete (e.g. type of system)
- **Replication**
  - Completely re-run experiment with same input levels
  - Used to determine impact of measurement error
- **Interaction**
  - *Effect* of one input factor depends on *level* of another input factor

7

Roadmap

- Goals
- Terminology
- Two-factor full factorial designs
  - 2-factor ANOVA
- General *m*-factor full factorial designs
  - *m*-factor ANOVA
- $n2^m$ factorial designs
- Fractional factorial designs
  - Plackett and Burman designs
- Case Studies

8

Two-Factor Experiments

- Two factors (inputs)
  - A, B
- Separate total variation in output values into:
  - Effect due to A
  - Effect due to B
  - Effect due to interaction of A and B (AB)
  - Experimental error

9

Example – User Response Time

- A = degree of multiprogramming
- B = memory size
- AB = interaction of memory size and degree of multiprogramming

	B (Mbytes)		
A	32	64	128
1	0.25	0.21	0.15
2	0.52	0.45	0.36
3	0.81	0.66	0.50
4	1.50	1.45	0.70

10

Why not vary one factor at a time?

- E.g. fix B to 64 MB and vary A. Then fix A to 3 and vary B ➔ this would reduce the number of configurations to be considered from 12 to 6!
- Problem: Unable to determine if there is any interaction between the memory size (factor B) and the degree of multiprogramming (factor A).
- If A = 4, the response time decreases nonlinearly with B. When A < 4, however, the response time appears to be more directly correlated to B.

11

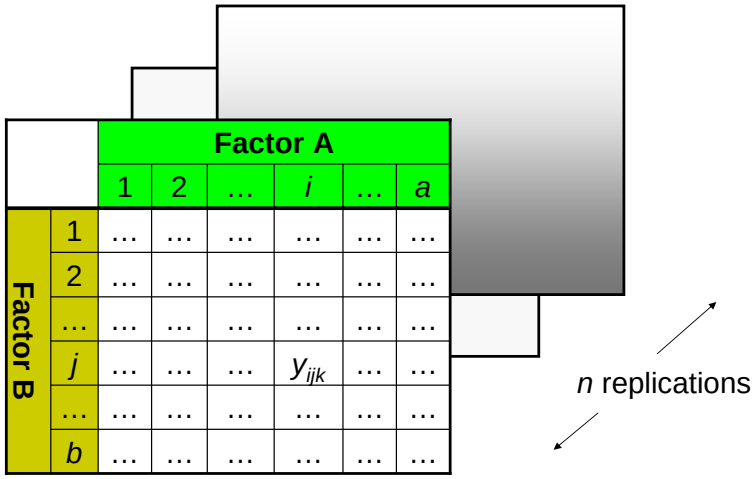
Two-Factor ANOVA

- Factor A – *a* input levels
- Factor B – *b* input levels
- *n* measurements for each input combination
- *abn* total measurements

12

Two Factors, *n* Replications

y_{ijk} is the k^{th} measurement with A set to its i^{th} level and B set to its j^{th} level



13

Recall: One-Factor ANOVA

- Each individual measurement was composition of
  - Overall mean
  - Effect of alternative
  - Measurement errors

$$y_{ij} = \bar{y}_{..} + \alpha_i + e_{ij}$$

$$\bar{y}_{..} = \text{overall mean}$$

$$\alpha_i = \text{effect due to A}$$

$$e_{ij} = \text{measurement error}$$

14

Two-Factor ANOVA

- Each individual measurement is composition of
  - Overall mean
  - Effects
  - Interactions
  - Measurement errors

$$y_{ijk} = \bar{y}_{...} + \alpha_i + \beta_j + \gamma_{ij} + e_{ijk}$$

$\bar{y}_{...}$ = overall mean

α_i = effect due to A

β_j = effect due to B

γ_{ij} = effect due to interaction of A and B

e_{ijk} = measurement error

15

Effects of Factors/Interactions

- The effects of the individual factors and their interactions are defined as follows:

$$\alpha_i = \bar{y}_{i..} - \bar{y}_{...} \qquad \sum_{i=1}^{a} \alpha_i = 0$$

$$\beta_j = \bar{y}_{.j.} - \bar{y}_{...} \qquad \sum_{j=1}^{b} \beta_j = 0$$

$$\gamma_{ij} = \bar{y}_{ij.} - \bar{y}_{i..} - \bar{y}_{.j.} + \bar{y}_{...} \qquad \sum_{i=1}^{a} \gamma_{ij} = 0 \quad \sum_{j=1}^{b} \gamma_{ij} = 0$$

Note: $\gamma_{ij} = \bar{y}_{ij.} - \bar{y}_{...} - \alpha_i - \beta_j$

16

Sum-of-Squares Terms

$$SST = \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}\left(y_{ijk} - \bar{y}_{...}\right)^2$$

$$SSA = bn\sum_{i=1}^{a}\left(\bar{y}_{i..} - \bar{y}_{...}\right)^2$$

$$SSB = an\sum_{j=1}^{b}\left(\bar{y}_{.j.} - \bar{y}_{...}\right)^2$$

$$SSAB = n\sum_{i=1}^{a}\sum_{j=1}^{b}\left(\bar{y}_{ij.} - \bar{y}_{i..} - \bar{y}_{.j.} + \bar{y}_{...}\right)^2$$

$$SSE = \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}\left(y_{ijk} - \bar{y}_{ij.}\right)^2$$

17

Sum-of-Squares

- As before, the *sum-of-squares identity* holds

  SST = SSA + SSB + SSAB + SSE

- Degrees of freedom
  - *df*(SSA) = *a* – 1
  - *df*(SSB) = *b* – 1
  - *df*(SSAB) = (*a* – 1)(*b* – 1)
  - *df*(SSE) = *ab*(*n* – 1)
  - *df*(SST) = *abn* – 1

df(SST) = df(SSA) + df(SSB) + df(SSAB) + df(SSE)

18

Computing The Sum-of-Squares Terms

$$SST = \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n} y_{ijk}^2 - \frac{S_{...}^2}{abn} \qquad S_{...} = \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n} y_{ijk}$$

$$SSA = \frac{\sum_{i=1}^{a} S_{i..}^2}{bn} - \frac{S_{...}^2}{abn} \qquad S_{i..} = \sum_{j=1}^{b}\sum_{k=1}^{n} y_{ijk}$$

$$SSB = \frac{\sum_{j=1}^{b} S_{.j.}^2}{an} - \frac{S_{...}^2}{abn} \qquad S_{.j.} = \sum_{i=1}^{a}\sum_{k=1}^{n} y_{ijk}$$

$$SSAB = \frac{\sum_{i=1}^{a}\sum_{j=1}^{b} S_{ij.}^2}{n} - \frac{\sum_{i=1}^{a} S_{i..}^2}{bn} - \frac{\sum_{j=1}^{b} S_{.j.}^2}{an} + \frac{S_{...}^2}{abn}$$

$$SSE = SST - SSA - SSB - SSAB$$

19

Two-Factor ANOVA - Summary

	A	B	AB	Error
Sum of squares	SSA	SSB	$SSAB$	SSE
Deg freedom	$a-1$	$b-1$	$(a-1)(b-1)$	$ab(n-1)$
Mean square	$s_a^2 = SSA/(a-1)$	$s_b^2 = SSB/(b-1)$	$s_{ab}^2 = SSAB/[(a-1)(b-1)]$	$s_e^2 = SSE/[ab(n-1)]$
Computed F	$F_a = s_a^2/s_e^2$	$F_b = s_b^2/s_e^2$	$F_{ab} = s_{ab}^2/s_e^2$	
Tabulated F	$F_{[1-\alpha;(a-1),ab(n-1)]}$	$F_{[1-\alpha;(b-1),ab(n-1)]}$	$F_{[1-\alpha;(a-1)(b-1),ab(n-1)]}$	

If $F_a > F[1-\alpha; a-1, ab(n-1)]$ the effect of factor A is statistically significant.

If $F_b > F[1-\alpha; b-1, ab(n-1)]$ the effect of factor B is statistically significant.

If $F_{ab} > F[1-\alpha; (a-1)(b-1), ab(n-1)]$ the effect of interaction AB is statistically significant.

20

Need for Replications

- If n=1, i.e. only one measurement of each configuration
- Can then be shown that
  - SSAB = SST – SSA – SSB
- SSE = SST – SSA – SSB – SSAB → SSE = 0
- Thus, when n=1 → No information about measurement errors
- Cannot separate effect due to interactions from measurement noise
- Must *replicate* each experiment at least twice

21

Example

- Output = user response time (seconds)
- Want to separate effects due to
  - A = degree of multiprogramming
  - B = memory size
  - AB = interaction
  - Error
- Need replications to separate error

	B (Mbytes)		
A	32	64	128
1	0.25	0.21	0.15
2	0.52	0.45	0.36
3	0.81	0.66	0.50
4	1.50	1.45	0.70

22

Example (cont.)

A	B (Mbytes) 32	64	128
1	0.25	0.21	0.15
	0.28	0.19	0.11
2	0.52	0.45	0.36
	0.48	0.49	0.30
3	0.81	0.66	0.50
	0.76	0.59	0.61
4	1.50	1.45	0.70
	1.61	1.32	0.68

23

Example (cont.)

	A	B	AB	Error
Sum of squares	3.3714	0.5152	0.4317	0.0293
Deg freedom	3	2	6	12
Mean square	1.1238	0.2576	0.0720	0.0024
Computed F	460.2	105.5	29.5	
Tabulated F	$F_{[0.95;3,12]} = 3.49$	$F_{[0.95;2,12]} = 3.89$	$F_{[0.95;6,12]} = 3.00$	

- 77.6% (SSA/SST) of all variation in response time due to degree of multiprogramming
- 11.8% (SSB/SST) due to memory size
- 9.9% (SSAB/SST) due to interaction of the two factors
- 0.7% due to measurement error
- 95% confident that all effects and interactions are statistically significant

24

Roadmap

- Goals
- Terminology
- Two-factor full factorial designs
  - 2-factor ANOVA
- General *m*-factor full factorial designs
  - *m*-factor ANOVA
- n2^m factorial designs
- Fractional factorial designs
  - Plackett and Burman designs
- Case Studies

25

Generalized *m*-Factor Experiments

m factors $\Rightarrow$ m main effects

$\binom{m}{2}$ two-factor interactions

$\binom{m}{3}$ three-factor interactions

$\vdots$

$\binom{m}{m} = 1$ m-factor interactions

$2^m - 1$ total effects

Effects for 3 factors:

A
B
C
AB
AC
BC
ABC

26

Degrees of Freedom for *m*-Factor Experiments

- df(SSA) = (a-1)
- df(SSB) = (b-1)
- df(SSC) = (c-1)
- df(SSAB) = (a-1)(b-1)
- df(SSAC) = (a-1)(c-1)
- …
- df(SSE) = abc(n-1)
- df(SST) = abcn-1

27

Procedure for Generalized *m*-Factor Experiments

1. Calculate (2^m-1) sum of squares terms (SSx) and SSE
2. Determine degrees of freedom for each SSx
3. Calculate mean squares (variances)
4. Calculate F statistics
5. Find critical F values from table
6. If F(computed) > F(table), (1-α) confidence that effect is statistically significant

28

Problem With Full-Factorial Designs

- *Full factorial design with replication*
  - Measure system response with all possible input combinations
  - Replicate each measurement *n* times to determine effect of measurement error
- *m* factors, *v* levels, *n* replications
  → $n\,v^m$ experiments
- *m* = 5 input factors, *v* = 4 levels, *n* = 3
  - → $3(4^5)$ = 3072 experiments!

29

Roadmap

- Goals
- Terminology
- Two-factor full factorial designs
  - 2-factor ANOVA
- General *m*-factor full factorial designs
  - *m*-factor ANOVA
- $n2^m$ factorial designs
- Fractional factorial designs
  - Plackett and Burman designs
- Case Studies

30

$n2^m$ Experiments

- Special case of generalized *m*-factor experiments
- Restrict each factor to two possible levels (values)
  - High, low
  - On, off
- Find factors that have largest impact
- Full factorial design with only those factors

31

Finding Sum of Squares Terms

For simplicity, assume that there are only two factors A and B

Sum of *n* measurements with (A,B) = (High, Low)	Factor A	Factor B
y_{AB}	High	High
y_{Ab}	High	Low
y_{aB}	Low	High
y_{ab}	Low	Low

32

$n2^m$ Contrasts and Sum of Squares

Contrasts are defined as follows:

$$w_A = y_{AB} + y_{Ab} - y_{aB} - y_{ab}$$
$$w_B = y_{AB} - y_{Ab} + y_{aB} - y_{ab}$$
$$w_{AB} = y_{AB} - y_{Ab} - y_{aB} + y_{ab}$$

And can be used to derive the sum of squares terms:

$$SSA = \frac{w_A^2}{n2^m} \qquad SSB = \frac{w_B^2}{n2^m} \qquad SSAB = \frac{w_{AB}^2}{n2^m}$$
$$SSE = SST - SSA - SSB - SSAB$$

33

$n2^m$ Experiments - Summary

	A	B	AB	Error
Sum of squares	SSA	SSB	$SSAB$	SSE
Deg freedom	1	1	1	$2^m(n-1)$
Mean square	$s_a^2 = SSA/1$	$s_b^2 = SSB/1$	$s_{ab}^2 = SSAB/1$	$s_e^2 = SSE/[2^m(n-1)]$
Computed F	$F_a = s_a^2/s_e^2$	$F_b = s_b^2/s_e^2$	$F_{ab} = s_{ab}^2/s_e^2$	
Tabulated F	$F_{[1-\alpha;1,2^m(n-1)]}$	$F_{[1-\alpha;1,2^m(n-1)]}$	$F_{[1-\alpha;1,2^m(n-1)]}$	

34

Contrasts for $n2^m$ with m = 2 factors revisited

Measurements	Contrast		
	w_A	w_B	w_{AB}
y_{AB}	+	+	+
y_{Ab}	+	-	-
y_{aB}	-	+	-
y_{ab}	-	-	+

$$w_A = y_{AB} + y_{Ab} - y_{aB} - y_{ab}$$
$$w_B = y_{AB} - y_{Ab} + y_{aB} - y_{ab}$$
$$w_{AB} = y_{AB} - y_{Ab} - y_{aB} + y_{ab}$$

35

Contrasts for $n2^m$ with m = 3 factors

Measure-ments	Contrast						
	w_A	w_B	w_C	w_{AB}	w_{AC}	w_{BC}	w_{ABC}
y_{abc}	-	-	-	+	+	+	-
y_{Abc}	+	-	-	-	-	+	+
y_{aBc}	-	+	-	-	+	-	+
…	…	…	…	…	…	…	…

$$w_{AC} = y_{abc} - y_{Abc} + y_{aBc} - y_{abC} - y_{ABc} + y_{AbC} - y_{aBC} + y_{ABC}$$

36

$n2^m$ with m = 3 factors

$$SSAC = \frac{w_{AC}^2}{2^3 n}$$

- df(each effect) = 1, since only two levels measured
- SST = SSA + SSB + SSC + SSAB + SSAC + SSBC + SSABC + SSE
- df(SSE) = $(n-1)2^3$
- Then perform ANOVA as before
- Easily generalizes to m > 3 factors

37

Important Points

- Experimental design is used to
  - Isolate the effects of each input variable.
  - Determine the effects of interactions.
  - Determine the magnitude of the error
  - Obtain maximum information for given effort
- Expanded 1-factor ANOVA to *m* factors
- Used $n2^m$ design to reduce the number of experiments needed
  - But loses some information

38

Roadmap

- Goals
- Terminology
- Two-factor full factorial designs
  - 2-factor ANOVA
- General *m*-factor full factorial designs
  - *m*-factor ANOVA
- n2^m factorial designs
- Fractional factorial designs
  - Plackett and Burman designs
- Case Studies

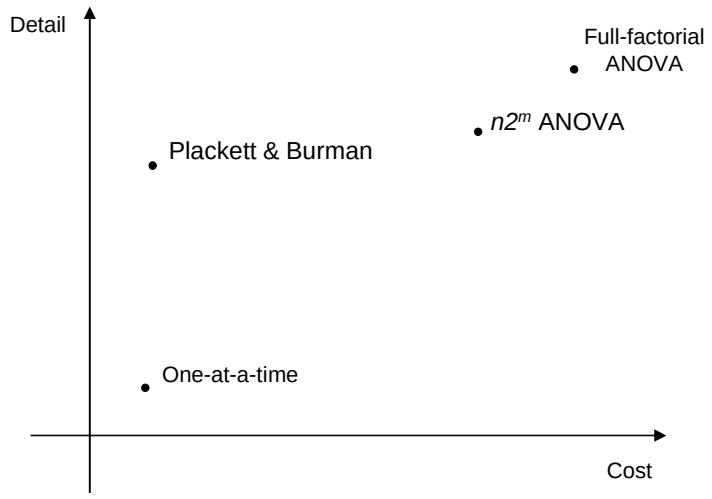
39

Still Too Many Experiments with *n2^m*!

- Plackett and Burman (PB) designs (1946)
  - Fractional multi-factorial designs
- Bridges the gap between:
  - Low-cost/low-detail approaches such as one-at-a-time
  - High-cost/high-detail approaches such as ANOVA
- Requires O(*m*) experiments for *m* factors
  - Instead of O(2^m) or O(v^m)
- Base PB designs ignore interactions
- PB designs with foldover
  - Quantify the effect of two-factor interactions

40

Trade-off Between Cost and Detail



41

Plackett and Burman (PB) Designs

- PB designs exist only in sizes that are multiples of 4
- Requires *X* experiments for *m* parameters
  - *X* = next multiple of 4 greater than *m*
- PB design matrix
  - X rows and X-1 columns
  - Rows = configurations
  - Columns = parameters' values in each configuration
    - High/low = +1/-1
    - If (m < X-1) use *dummy parameters*
  - First row initialized from P&B paper (see below)
  - Subsequent rows = circular right shift of preceding row
  - Last row = all (-1)
- Plackett, R. and Burman, J., *"The design of optimum multifactorial experiments",* Biometrika, 33, 4, 1946, 305-325.

42

PB Design Matrix

Config	Input Parameters (factors)							Response
	A	B	C	D	E	F	G	
1	+1	+1	+1	-1	+1	-1	-1	
2	-1	+1	+1	+1	-1	+1	-1	
3	-1	-1	+1	+1	+1	-1	+1	
4	+1	-1	-1	+1	+1	+1	-1	
5	-1	+1	-1	-1	+1	+1	+1	
6	+1	-1	+1	-1	-1	+1	+1	
7	+1	+1	-1	+1	-1	-1	+1	
8	-1	-1	-1	-1	-1	-1	-1	
Effect								

43

Choice of High/Low Values

- High/Low values need to be chosen for each parameter (factor)
- Selecting high and low values that span a range of values that is too small may understimate the effect of the parameter.
- Too large a range may overestimate the effect.
- Ideally, the high and low values should be just outside of the normal (or expected) range of values.

44

PB Design Matrix

Config	Input Parameters (Factors)							Response
	A	B	C	D	E	F	G	
1	+1	+1	+1	-1	+1	-1	-1	9
2	-1	+1	+1	+1	-1	+1	-1	11
3	-1	-1	+1	+1	+1	-1	+1	2
4	+1	-1	-1	+1	+1	+1	-1	1
5	-1	+1	-1	-1	+1	+1	+1	9
6	+1	-1	+1	-1	-1	+1	+1	74
7	+1	+1	-1	+1	-1	-1	+1	7
8	-1	-1	-1	-1	-1	-1	-1	4
Effect								

45

Computing Effects

$$Effect_A = (+1\times 9) + (-1\times 11) + (-1\times 2) + \ldots + (-1\times 4) = 65$$

Config	Input Parameters (Factors)							Response
	A	B	C	D	E	F	G	
1	+1	+1	+1	-1	+1	-1	-1	9
2	-1	+1	+1	+1	-1	+1	-1	11
3	-1	-1	+1	+1	+1	-1	+1	2
4	+1	-1	-1	+1	+1	+1	-1	1
5	-1	+1	-1	-1	+1	+1	+1	9
6	+1	-1	+1	-1	-1	+1	+1	74
7	+1	+1	-1	+1	-1	-1	+1	7
8	-1	-1	-1	-1	-1	-1	-1	4
Effect	65							

46

Computing Effects (cont.)

Config	Input Parameters (factors)							Response
	A	B	C	D	E	F	G	
1	+1	+1	+1	-1	+1	-1	-1	9
2	-1	+1	+1	+1	-1	+1	-1	11
3	-1	-1	+1	+1	+1	-1	+1	2
4	+1	-1	-1	+1	+1	+1	-1	1
5	-1	+1	-1	-1	+1	+1	+1	9
6	+1	-1	+1	-1	-1	+1	+1	74
7	+1	+1	-1	+1	-1	-1	+1	7
8	-1	-1	-1	-1	-1	-1	-1	4
Effect	65	-45						

47

Computing Effects (cont.)

Config	Input Parameters (Factors)							Response
	A	B	C	D	E	F	G	
1	+1	+1	+1	-1	+1	-1	-1	9
2	-1	+1	+1	+1	-1	+1	-1	11
3	-1	-1	+1	+1	+1	-1	+1	2
4	+1	-1	-1	+1	+1	+1	-1	1
5	-1	+1	-1	-1	+1	+1	+1	9
6	+1	-1	+1	-1	-1	+1	+1	74
7	+1	+1	-1	+1	-1	-1	+1	7
8	-1	-1	-1	-1	-1	-1	-1	4
Effect	65	-45	75	-75	-75	73	67	

48

Parameter Ranking

- Effects determine the relative impacts of parameters on the variation observed in the output.
- Only magnitude of effect is important
  - Sign is meaningless
- In example, most → least important parameters:
  - [C, D, E] → F → G → A → B
- Parameter with highest rank is considered a *performance bottleneck*, since a poor choice of its value will impact performance significantly.

49

PB Design with Foldover

- Provides some additional information
  - Quantifies effects of two-factor interactions
- Add *X* additional rows to matrix
  - Signs of additional rows are opposite original rows

50

PB Design Matrix with Foldover

A	B	C	D	E	F	G	Exec. Time
+1	+1	+1	-1	+1	-1	-1	9
-1	+1	+1	+1	-1	+1	-1	11
-1	-1	+1	+1	+1	-1	+1	2
+1	-1	-1	+1	+1	+1	-1	1
-1	+1	-1	-1	+1	+1	+1	9
+1	-1	+1	-1	-1	+1	+1	74
+1	+1	-1	+1	-1	-1	+1	7
-1	-1	-1	-1	-1	-1	-1	4
-1	-1	-1	+1	-1	+1	+1	17
+1	-1	-1	-1	+1	-1	+1	76
+1	+1	-1	-1	-1	+1	-1	6
-1	+1	+1	-1	-1	-1	+1	31
+1	-1	+1	+1	-1	-1	-1	19
-1	+1	-1	+1	+1	-1	-1	33
-1	-1	+1	-1	+1	+1	-1	6
+1	+1	+1	+1	+1	+1	+1	112
191	19	111	-13	79	55	239	

51

PB Design with Foldover

- Requires 2X experiments

$$Effect_{AB} = ((1\times1)\times9) + ((-1\times1)\times11) + ((-1\times-1)\times2) + \ldots + ((1\times1)\times112) = \ldots$$

52

Design Space Exploration

- Common activity in simulation-based computer architecture research and design
- Find optimal configuration
  - Step 1: Use PB design to find the most significant parameters (reduces # parameters from m→n).
    - Cost = *2m* simulations.
  - Step 2: Reduced parameters can then be fully explored using full factorial ANOVA.
    - Cost = 2^n simulations.

53

Important Points

- Plackett and Burman design
  - Requires only O(*m*) experiments
  - Estimates effects of main factors
  - Plus effects of 2-factor interactions when w/ foldover
- Logically minimal number of experiments
- Powerful technique for obtaining a big-picture view of a lot of data

54

Roadmap

- Goals
- Terminology
- Two-factor full factorial designs
  - 2-factor ANOVA
- General *m*-factor full factorial designs
  - *m*-factor ANOVA
- n2^m factorial designs
- Fractional factorial designs
  - Plackett and Burman designs
- Case Studies

55

Case Study #1

- Determine the most significant parameters in a processor simulator.
- *"A Statistically Rigorous Approach for Improving Simulation Methodology"* by Joshua J. Yi, David J. Lilja, and Douglas M. Hawkins, International Symposium on High-Performance Computer Architecture (HPCA), February, 2003.

56

Determine the Most Significant Processor Parameters

- Problem
  - So many parameters in a processor simulator.
  - How to choose parameter values?
  - How to decide which parameters are most important?
- Approach
  - Use Plackett & Burman design.
  - Choose reasonable upper/lower bounds.
  - Rank parameters by impact on total execution time.

57

Simulation Environment

- Superscalar simulator
  - sim-outorder 3.0 from the SimpleScalar tool suite
- Selected SPECcpu2000 Benchmarks
  - *gzip, vpr, gcc, mesa, art, mcf, equake, parser, vortex, bzip2, twolf*

58

Functional Unit Values

Parameter	Low Value	High Value
Int ALUs	1	4
Int ALU Latency	2 Cycles	1 Cycle
Int ALU Throughput	1	
FP ALUs	1	4
FP ALU Latency	5 Cycles	1 Cycle
FP ALU Throughputs	1	
Int Mult/Div Units	1	4
Int Mult Latency	15 Cycles	2 Cycles
Int Div Latency	80 Cycles	10 Cycles
Int Mult Throughput	1	
Int Div Throughput	Equal to Int Div Latency	
FP Mult/Div Units	1	4
FP Mult Latency	5 Cycles	2 Cycles
FP Div Latency	35 Cycles	10 Cycles
FP Sqrt Latency	35 Cycles	15 Cycles
FP Mult Throughput	Equal to FP Mult Latency	
FP Div Throughput	Equal to FP Div Latency	
FP Sqrt Throughput	Equal to FP Sqrt Latency	

Memory System Values, Part I

Parameter	Low Value	High Value
L1 I-Cache Size	4 KB	128 KB
L1 I-Cache Assoc	1-Way	8-Way
L1 I-Cache Block Size	16 Bytes	64 Bytes
L1 I-Cache Repl Policy	Least Recently Used	
L1 I-Cache Latency	4 Cycles	1 Cycle
L1 D-Cache Size	4 KB	128 KB
L1 D-Cache Assoc	1-Way	8-Way
L1 D-Cache Block Size	16 Bytes	64 Bytes
L1 D-Cache Repl Policy	Least Recently Used	
L1 D-Cache Latency	4 Cycles	1 Cycle
L2 Cache Size	256 KB	8192 KB
L2 Cache Assoc	1-Way	8-Way
L2 Cache Block Size	64 Bytes	256 Bytes

60

Memory System Values, Part II

Parameter	Low Value	High Value
L2 Cache Repl Policy	Least Recently Used	
L2 Cache Latency	20 Cycles	5 Cycles
Mem Latency, First	200 Cycles	50 Cycles
Mem Latency, Next	0.02 * Mem Latency, First	
Mem Bandwidth	4 Bytes	32 Bytes
I-TLB Size	32 Entries	256 Entries
I-TLB Page Size	4 KB	4096 KB
I-TLB Assoc	2-Way	Fully Assoc
I-TLB Latency	80 Cycles	30 Cycles
D-TLB Size	32 Entries	256 Entries
D-TLB Page Size	Same as I-TLB Page Size	
D-TLB Assoc	2-Way	Fully-Assoc
D-TLB Latency	Same as I-TLB Latency	

61

Processor Core Values

Parameter	Low Value	High Value
Fetch Queue Entries	4	32
Branch Predictor	2-Level	Perfect
Branch MPred Penalty	10 Cycles	2 Cycles
RAS Entries	4	64
BTB Entries	16	512
BTB Assoc	2-Way	Fully-Assoc
Spec Branch Update	In Commit	In Decode
Decode/Issue Width	4-Way	
ROB Entries	8	64
LSQ Entries	0.25 * ROB	1.0 * ROB
Memory Ports	1	4

62

Determining the Most Significant Parameters

1. Run simulations to find **response**
   - With input parameters at high/low, on/off values

Config	Input Parameters (Factors)							Response
	A	B	C	D	E	F	G	
1	+1	+1	+1	-1	+1	-1	-1	9
2	-1	+1	+1	+1	-1	+1	-1	
3	-1	-1	+1	+1	+1	-1	+1	
…	…	…	…	…	…	…	…	
Effect								

63

Determining the Most Significant Parameters (2)

2. Calculate the **effect** of each parameter
   - Across configurations

Config	Input Parameters (factors)							Response
	A	B	C	D	E	F	G	
1	+1	+1	+1	-1	+1	-1	-1	9
2	-1	+1	+1	+1	-1	+1	-1	
3	-1	-1	+1	+1	+1	-1	+1	
…	…	…	…	…	…	…	…	
Effect	65							

64

Determining the Most Significant Parameters (3)

3. For each benchmark

   Rank the parameters in descending order of effect (1 = most important, …)

Parameter	Benchmark 1	Benchmark 2	Benchmark 3
A	3	12	8
B	29	4	22
C	2	6	7
…	…	…	…

65

Determining the Most Significant Parameters (4)

4. For each parameter

   Average the ranks

Parameter	Benchmark 1	Benchmark 2	Benchmark 3	Average
A	3	12	8	7.67
B	29	4	22	18.3
C	2	6	7	5
…	…	…	…	…

66

Most Significant Parameters

Number	Parameter	gcc	gzip	art	Average
1	**ROB Entries**	4	1	2	2.77
2	**L2 Cache Latency**	2	4	4	4.00
3	**Branch Predictor Accuracy**	5	2	27	7.69
4	**Number of Integer ALUs**	8	3	29	9.08
5	**L1 D-Cache Latency**	7	7	8	10.00
6	**L1 I-Cache Size**	1	6	12	10.23
7	**L2 Cache Size**	6	9	1	10.62
8	**L1 I-Cache Block Size**	3	16	10	11.77
9	**Memory Latency, First**	9	36	3	12.31
10	**LSQ Entries**	10	12	39	12.62
11	**Speculative Branch Update**	28	8	16	18.23

67

General Procedure

- Determine upper/lower *bounds* for parameters
- Simulate configurations to find *response*
- Compute *effects* parameters
- *Rank* the parameters for each benchmark based on effects
- *Average* the ranks across benchmarks
- Focus on *top-ranked* parameters for subsequent analysis

68

Summary - Case Study #1

- Started with 41 parameters (2^{41} = 2.2 trillion potential test cases!)
- Reduced to 88 Plackett & Burman test cases (X=44, 2x44) plus 1024 ANOVA test cases (2^{10}) for a total of 1112 test cases!
- Using PB design to first pare the design space reduced the number of test cases by over *nine orders of magnitude*!

69

Case Study #2

- Determine the “big picture” impact of a system enhancement.

70

Determining the Overall Effect of an Enhancement

- Find most important parameters without enhancement
  - Using Plackett and Burman
- Find most important parameters with enhancement
  - Again using Plackett and Burman
- Compare parameter ranks

71

Example: Instruction Precomputation

- Profile to find the most common operations
  - 0+1, 1+1, etc.
- Insert the results of common operations in a table when the program is loaded into memory
- Query the table when an instruction is issued
- Don't execute the instruction if it is already in the table
- Reduces contention for function units

72

The Effect of Instruction Precomputation

	Average Rank		
Parameter	**Before**	**After**	**Difference**
ROB Entries	2.77		
L2 Cache Latency	4.00		
Branch Predictor Accuracy	7.69		
Number of Integer ALUs	9.08		
L1 D-Cache Latency	10.00		
L1 I-Cache Size	10.23		
L2 Cache Size	10.62		
L1 I-Cache Block Size	11.77		
Memory Latency, First	12.31		
LSQ Entries	12.62		

73

The Effect of Instruction Precomputation (2)

	Average Rank		
Parameter	**Before**	**After**	**Difference**
ROB Entries	2.77	2.77	
L2 Cache Latency	4.00	4.00	
Branch Predictor Accuracy	7.69	7.92	
Number of Integer ALUs	9.08	10.54	
L1 D-Cache Latency	10.00	9.62	
L1 I-Cache Size	10.23	10.15	
L2 Cache Size	10.62	10.54	
L1 I-Cache Block Size	11.77	11.38	
Memory Latency, First	12.31	11.62	
LSQ Entries	12.62	13.00	

74

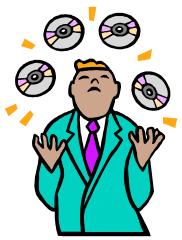
The Effect of Instruction Precomputation (3)

	Average Rank		
Parameter	**Before**	**After**	**Difference**
ROB Entries	2.77	2.77	0.00
L2 Cache Latency	4.00	4.00	0.00
Branch Predictor Accuracy	7.69	7.92	-0.23
Number of Integer ALUs	9.08	10.54	-1.46
L1 D-Cache Latency	10.00	9.62	0.38
L1 I-Cache Size	10.23	10.15	0.08
L2 Cache Size	10.62	10.54	0.08
L1 I-Cache Block Size	11.77	11.38	0.39
Memory Latency, First	12.31	11.62	0.69
LSQ Entries	12.62	13.00	-0.38

75

Case Study #3

- Benchmark program classification.



76

Benchmark Classification

- By application type
  - Scientific and engineering applications
  - Transaction processing applications
  - Multimedia applications
- By use of processor function units
  - Floating-point code
  - Integer code
  - Memory intensive code
- Etc., etc.

77

Another Point-of-View

- Classify by overall *impact* on processor
- Define:
  - Two benchmark programs are **similar** if
    - They stress the same components of a system to similar degrees
- How to measure this similarity?
  - Use Plackett and Burman design to find ranks
  - Then compare ranks

78

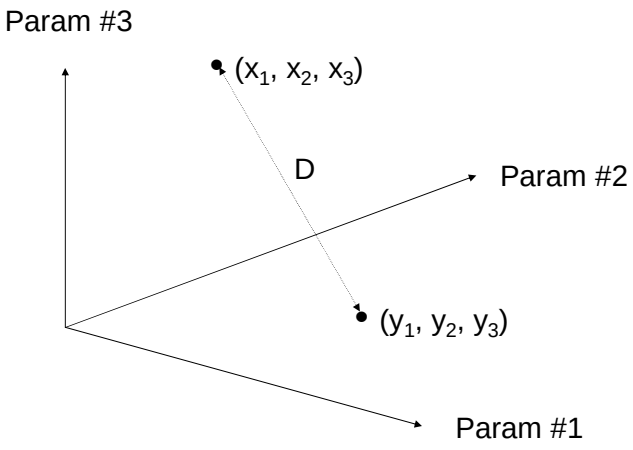
Similarity Metric

- Use rank of each parameter as elements of a vector
- For benchmark program X, let
  - **X** = (x_1, x_2,…, x_{n-1}, x_n)
  - x_1 = rank of parameter 1
  - x_2 = rank of parameter 2
  - …
- Use the Euclidean distance between points as similarity metric:

$$D = [(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_{n-1} - y_{n-1})^2 + (x_n - y_n)^2]^{1/2}$$

79

Vector Defines a Point in *n*-space



80

Most Significant Parameters

Number	Parameter	gcc	gzip	art
1	**ROB Entries**	4	1	2
2	**L2 Cache Latency**	2	4	4
3	**Branch Predictor Accuracy**	5	2	27
4	**Number of Integer ALUs**	8	3	29
5	**L1 D-Cache Latency**	7	7	8
6	**L1 I-Cache Size**	1	6	12
7	**L2 Cache Size**	6	9	1
8	**L1 I-Cache Block Size**	3	16	10
9	**Memory Latency, First**	9	36	3
10	**LSQ Entries**	10	12	39
11	**Speculative Branch Update**	28	8	16

81

Distance Computation

- Rank vectors
  - gcc = (4, 2, 5, 8, …)
  - gzip = (1, 4, 2, 3, …)
  - art = (2, 4, 27, 29, …)
- Euclidean distances
  - D(gcc - gzip) = $[(4-1)^2 + (2-4)^2 + (5-2)^2 + \ldots]^{1/2}$
  - D(gcc - art) = $[(4-2)^2 + (2-4)^2 + (5-27)^2 + \ldots]^{1/2}$
  - D(gzip - art) = $[(1-2)^2 + (4-4)^2 + (2-27)^2 + \ldots]^{1/2}$

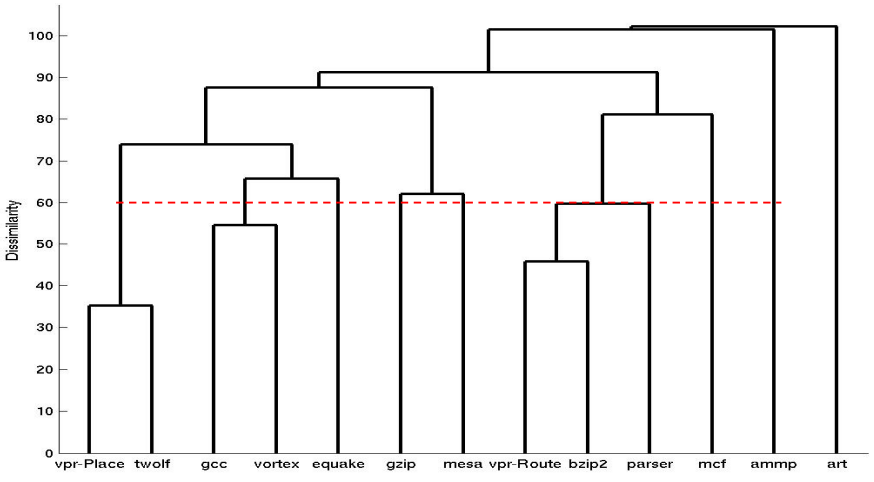
82

Euclidean Distances for Selected Benchmarks

	gcc	gzip	art	mcf
gcc	0	81.9	92.6	94.5
gzip		0	113.5	109.6
art			0	98.6
mcf				0

83

Dendogram of Distances Showing (Dis-)Similarity



Final Benchmark Groupings

Group	Benchmarks
I	Gzip, mesa
II	Vpr-Place, twolf
III	Vpr-Route, parser, bzip2
IV	Gcc, vortex
V	Art
VI	Mcf
VII	Equake
VIII	ammp

85

Summary

- Experimental Design (Design of Experiments)
  - Isolate effects of each input variable.
  - Determine effects of interactions.
  - Determine magnitude of experimental error.
- *m*-factor ANOVA (full factorial design)
  - All effects, interactions, and errors
- $n2^m$ designs
  - All effects, interactions, and errors
  - But for only 2 input values
    - high/low
    - on/off

86

Summary (cont.)

- Plackett and Burman (*fractional factorial design*)
- O(*m*) experiments
- Quantifies main effects and 2-factor interactions
- For only 2 input values (high/low, on/off)
- Applications – rank parameters, group benchmarks, overall impact of an enhancement

87

Further Reading

- *„Performance Evaluation and Benchmarking“* – Edited by Lizy Kurian John and Lieven Eeckhout, (c) 2006 CRC Press
- *“The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling”* by Raj Jain, (c) 1991 Wiley
- *“Analyzing the Processor Bottlenecks in SPEC CPU 2000”* by Joshua J. Yi, Ajay Joshi, Resit Sendag, Lieven Eeckhout and David J. Lilja, SPEC Benchmark Workshop 2006
- *“A Statistically Rigorous Approach for Improving Simulation Methodology”* by Joshua J. Yi, David J. Lilja, and Douglas M. Hawkins, International Symposium on High-Performance Computer Architecture (HPCA), February, 2003.
- Plackett, R. and Burman, J., *“The design of optimum multifactorial experiments”,* Biometrika, 33, 4, 1946, 305-325.

88