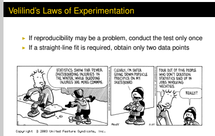
CS 147:
Computer Systems Performance Analysis
Examples Using a Distributed File System

CS 147:
Computer Systems Performance Analysis
Examples Using a Distributed File System

Velilind’s Laws of Experimentation

- If reproducibility may be a problem, conduct the test only once
- If a straight-line fit is required, obtain only two data points

└Velilind’s Laws of Experimentation



Overview

Overview
Overview of the Ficus File System
Characteristics
Performance Issues
Measured Data
Measurement Methodology
Raw Results
Data Analysis
What Can Be Analyzed?
Sample Analysis
Quality of the Analysis
Visual Tests
A Bad Example

What is Ficus?

- Distributed, replicated file system
- Individual computers store replicas of shared files
  - Fast local access
  - Shared data
- Designed for robustness in face of network disconnections
  - Anyone can write any file, any time

What is Ficus?
- Distributed, replicated file system
- Individual computers store replicas of shared files
  - Fast local access
  - Shared data
- Designed for robustness in face of network disconnections
  - Anyone can write any file, any time

Propagation

- Any update generates a “best-effort” propagation message
  - Generated on every write system call
  - Broadcast to all known replicas
  - Notifies of change, not contents
- Receiving site can ignore or can request latest version of file from generating site
  - Only when no conflict

Propagation

- Any update generates a “best-effort” propagation message
  - Generated on every write system call
  - Broadcast to all known replicas
  - Notifies of change, not contents
- Receiving site can ignore or can request latest version of file from generating site
  - Only when no conflict

Reconciliation

- Correctness guarantees provided by *reconciliation* process
- Runs periodically
- Operates between pair of replicas
  - Transfers data in one direction only
- Complex distributed algorithm
  - Proven to terminate correctly
  - Data is guaranteed to eventually get everywhere

Reconciliation

- Correctness guarantees provided by *reconciliation* process
- Runs periodically
- Operates between pair of replicas
  - Transfers data in one direction only
- Complex distributed algorithm
  - Proven to terminate correctly
  - Data is guaranteed to eventually get everywhere

Garbage Collection

- Tricky to get deletion right
- Example: Joe deletes `foo` while Mary renames it to `bar`
- Need to globally agree that all names are gone
- Requires complex two-phase distributed algorithm

Garbage Collection

- Tricky to get deletion right
- Example: Joe deletes `foo` while Mary renames it to `bar`
- Need to globally agree that all names are gone
- Requires complex two-phase distributed algorithm

Ficus Performance

- File access (open) performance
- Read/write performance
- Aspects of deletion
- Reconciliation
- Cross-machine interference

Ficus Performance

- File access (open) performance
- Read/write performance
- Aspects of deletion
- Reconciliation
- Cross-machine interference

Open Performance

- Opening file requires:
  - Finding file
  - Checking for conflicts
  - Local or remote (NFS-like) open
- Finding file requires
  - Local or remote root access
  - Tracing path, changing machines as needed
- Other steps are basically one remote procedure call (RPC—one message exchange) each

Open Performance

- Opening file requires:
  - Finding file
  - Checking for conflicts
  - Local or remote (NFS-like) open
- Finding file requires
  - Local or remote root access
  - Tracing path, changing machines as needed
- Other steps are basically one remote procedure call (RPC—one message exchange) each

Read/Write Performance

- Reading is same as local or NFS operation
- Write is like local or NFS, plus:
  - Propagation (small outgoing packet)
  - Attribute update (beyond i-node update)

Read/Write Performance

- Reading is same as local or NFS operation
- Write is like local or NFS, plus:
  - Propagation (small outgoing packet)
  - Attribute update (beyond i-node update)

Deletion

- Initially removing a file is reasonably cheap
  - Mark deleted
  - Remove from visible namespace
  - May actually be cheaper than UFS unlink
- True cost is garbage collection
  - How long is space consumed?
  - CPU cost?
  - Still have to do unlink equivalent someday

Deletion

- Initially removing a file is reasonably cheap
  - Mark deleted
  - Remove from visible namespace
  - May actually be cheaper than UFS unlink
- True cost is garbage collection
  - How long is space consumed?
  - CPU cost?
  - Still have to do unlink equivalent someday

Reconciliation

- Runs periodically
- Mechanism to suppress under high load
- Must check every file
  - If updated, exchange info with remote
  - May also transfer data
  - Special handling, but similar, for new/deleted files
- Primary cost is checking what's updated

Reconciliation
- Runs periodically
- Mechanism to suppress under high load
- Must check every file
  - If updated, exchange info with remote
  - May also transfer data
  - Special handling, but similar, for new/deleted files
- Primary cost is checking what's updated

Cross-Machine Interference

- If you store a replica, you pay some costs:
  - Receiving propagation requests
  - Running reconciliation as client and server
  - Servicing remote access requests

Cross-Machine Interference
- If you store a replica, you pay some costs:
  - Receiving propagation requests
  - Running reconciliation as client and server
  - Servicing remote access requests

Ficus Measurement Methodology

- Two classes of measurement
  - Local replica
  - Interference with remote replicas
- Set up test volume
- Populate with files
- Run several “standard” benchmarks
- Destroy volume after test

Ficus Measurement Methodology

- Two classes of measurement
  - Local replica
  - Interference with remote replicas
- Set up test volume
- Populate with files
- Run several “standard” benchmarks
- Destroy volume after test

Benchmarks Used

- Eight benchmarks: cp, find, findgrep, grep, ls, mab, rcp, rm
- Most did single operation implied by name
  - cp copied locally within volume
  - rcp copied from remote machine
  - findgrep essentially did recursive grep
  - mab, Modified Andrew Benchmark, did more complex compile-edit-debug simulation

Benchmarks Used

- Eight benchmarks: cp, find, findgrep, grep, ls, mab, rcp, rm
- Most did single operation implied by name
  - cp copied locally within volume
  - rcp copied from remote machine
  - findgrep essentially did recursive grep
  - mab, Modified Andrew Benchmark, did more complex compile-edit-debug simulation

Local-Replica Measurements

- Set up UFS, remotely-accessed NFS, or Ficus volume
  - Ficus volume varies from 1 to 8 replicas
- Run benchmarks on machine that stores local copy (except for NFS tests)
- Ignore effect on machines holding other replicas

Local-Replica Measurements

- Set up UFS, remotely-accessed NFS, or Ficus volume
  - Ficus volume varies from 1 to 8 replicas
- Run benchmarks on machine that stores local copy (except for NFS tests)
- Ignore effect on machines holding other replicas

Interference Measurements

- Set up UFS volume on “interfered” machine
- On 1 to 3 other machines, set up 2-replica Ficus volume
  - Unique volume for each machine
  - Second replica stored on “interfered” machine
- Run all 8 benchmarks simultaneously on all machines
  - Compare UFS time to uninterfered version

Interference Measurements

- Set up UFS volume on “interfered” machine
- On 1 to 3 other machines, set up 2-replica Ficus volume
  - Unique volume for each machine
  - Second replica stored on “interfered” machine
- Run all 8 benchmarks simultaneously on all machines
  - Compare UFS time to uninterfered version

Example of Raw Ficus Results

```
.../RESULTS/950531.211023/benchtimes:ficus mab 2
162.9 real 83.2 user 40.9 sys
```

- Test was run on May 31, 1995, at 21:10:23
- Ficus test with with MAB benchmark, 2 replicas
- 162.9 seconds for run; 83.2 user time, 40.9 charged to system

Example of Raw Ficus Results
.../RESULTS/950531.211023/benchtimes:ficus mab 2
162.9 real 83.2 user 40.9 sys
Test was run on May 31, 1995, at 21:10:23
Ficus test with with MAB benchmark, 2 replicas
162.9 seconds for run; 83.2 user time, 40.9 charged to system

The “Standard” Analysis

- Everybody publishes means, usually in nice tables or graphs
- Standard deviations are becoming fairly common
- Sometimes they even tell you how many runs they did
  - Allows you to generate confidence intervals

The “Standard” Analysis

- Everybody publishes means, usually in nice tables or graphs
- Standard deviations are becoming fairly common
- Sometimes they even tell you how many runs they did
  - Allows you to generate confidence intervals

Earning Some Self-Respect

- You should always provide the reader or listener with at least:
  - A mean of a specified number of runs
  - A confidence interval at 90% or higher
  - An analysis of whether the results are meaningful
- Standard deviations are nice, but not as important as confidence intervals

Earning Some Self-Respect

- You should always provide the reader or listener with at least:
  - A mean of a specified number of runs
  - A confidence interval at 90% or higher
  - An analysis of whether the results are meaningful
- Standard deviations are nice, but not as important as confidence intervals

Learning Something About the System

- Use confidence intervals to compare various parameters and results
- Consider whether regression is meaningful
- Can you do multivariate regression?
- What about ANOVA?

Learning Something About the System

- Use confidence intervals to compare various parameters and results
- Consider whether regression is meaningful
- Can you do multivariate regression?
- What about ANOVA?

Sample Analysis of Ficus Experiments

- We will consider only the “cp” benchmark
- Local-replica tests only
- Questions to ask:
  - Is Ficus significantly slower than UFS?
  - Is Ficus faster than NFS remote access?
  - What is the cost of adding a remote replica?

Sample Analysis of Ficus Experiments

- We will consider only the “cp” benchmark
- Local-replica tests only
- Questions to ask:
  - Is Ficus significantly slower than UFS?
  - Is Ficus faster than NFS remote access?
  - What is the cost of adding a remote replica?

Some Raw “cp” Benchmark Data

Repls	Real	Repls	Real	Repls	Real
1	179.5	2	189.0	3	178.0
1	193.2	2	246.6	3	207.9
1	197.4	2	227.7	3	202.0
1	231.8	2	275.2	3	213.9
1	202.4	2	203.8	3	218.2
1	180.3	2	235.3	3	249.0
1	222.1	2	199.9	3	207.6
1	186.2	2	168.6	3	213.2

Some Raw “cp” Benchmark Data
Repls Real Repls Real Repls Real
1 179.5 2 189.0 3 178.0
1 193.2 2 246.6 3 207.9
1 197.4 2 227.7 3 202.0
1 231.8 2 275.2 3 213.9
1 202.4 2 203.8 3 218.2
1 180.3 2 235.3 3 249.0
1 222.1 2 199.9 3 207.6
1 186.2 2 168.6 3 213.2

Is Ficus Slower Than UFS?

- UFS “cp” benchmark, 90% confidence interval is $(167.6, 186.6)$, mean 177.1
- Fastest Ficus (1-replica) “cp” 90% confidence is $(188.0, 210.2)$, mean 199.1
- Non-overlapping intervals $\Rightarrow$ meaningful difference, Ficus is indeed slower
- Results might differ at higher confidence

Is Ficus Slower Than UFS?
- UFS “cp” benchmark, 90% confidence interval is (167.6, 186.6), mean 177.1
- Fastest Ficus (1-replica) “cp” 90% confidence is (188.0, 210.2), mean 199.1
- Non-overlapping intervals ⇒ meaningful difference, Ficus is indeed slower
- Results might differ at higher confidence

Is Ficus Faster Than Remote NFS?

- NFS interval is $(231.0, 259.3)$, mean 245.2
- 3-replica Ficus is $(198.1, 224.4)$ at 90%
- So Ficus is definitely faster
- (Incidentally, result would have held even at higher confidence)

Is Ficus Faster Than Remote NFS?

- NFS interval is (231.0, 259.3), mean 245.2
- 3-replica Ficus is (198.1, 224.4) at 90%
- So Ficus is definitely faster
- (Incidentally, result would have held even at higher confidence)

What is the Cost of a New Replica?

- Do regression on data from 1 to 8 replicas
  - Note that some tests have different numbers of runs
  - Regression on means will give incorrect results
  - Proper method: regress on raw observations, with repeated x values
- Time $= 20.25 \times$ replicas $+ 168.12$
- So each replica slows “cp” by about 20 seconds

What is the Cost of a New Replica?

- Do regression on data from 1 to 8 replicas
  - Note that some tests have different numbers of runs
  - Regression on means will give incorrect results
  - Proper method: regress on raw observations, with repeated x values
- Time $= 20.25 \times$ replicas $+ 168.12$
- So each replica slows “cp” by about 20 seconds

Allocation of Variation for “cp” Benchmark

- SSR $= 76590.40$, SSE $= 61437.46$
- $R^2 = SSR/(SSR + SSE) = 0.55$
  - So regression explains only 50% of variation
- Standard deviation of errors

$$s_e^2 = \sqrt{\frac{SSE}{n-2}} = \sqrt{\frac{61437.46}{44-2}} = 38.25$$

  - Relatively high value compared to mean of 240.4

Allocation of Variation for “cp” Benchmark
- SSR $= 76590.40$, SSE $= 61437.46$
- $R^2 = SSR/(SSR + SSE) = 0.55$
  - So regression explains only 50% of variation
- Standard deviation of errors
$$s_e^2 = \sqrt{\frac{SSE}{n-2}} = \sqrt{\frac{61437.46}{44-2}} = 38.25$$
  - Relatively high value compared to mean of 240.4

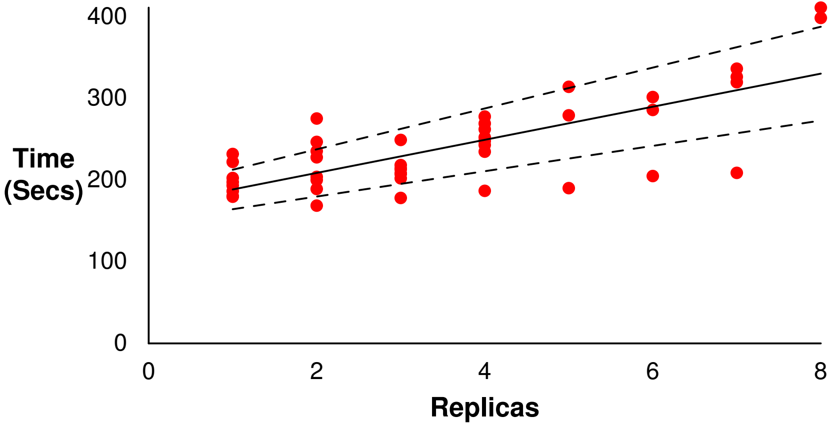
Confidence Intervals for "cp" Regression

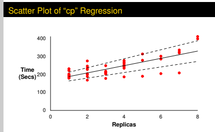
- $s_{b_0} = 11.53$, $s_{b_1} = 2.80$
- Using 90% confidence level, $t_{0.95;44} = 1.68$
- This gives $b_0 = (148.7, 187.5)$, while $b_1 = (15.5, 25.0)$
- Standard deviation for 9-replica prediction, single observation is $s_e(1.09) = 41.6$
- Using same t, interval is $(308.8, 391.9)$
  - Compare to narrower 1-replica interval

Confidence Intervals for "cp" Regression

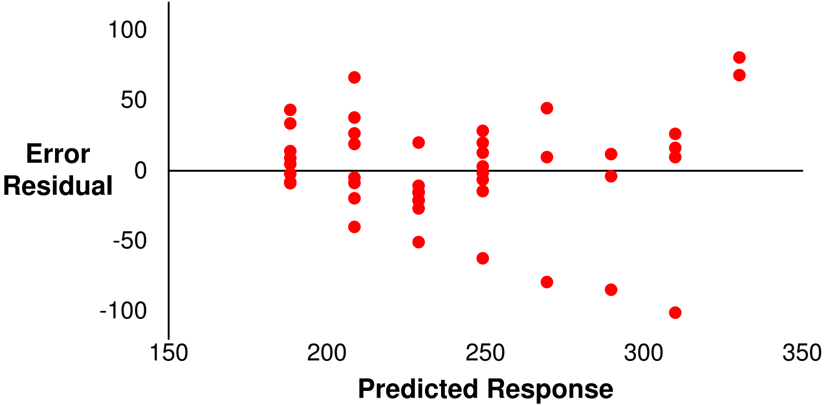
- $s_{b_0} = 11.53$, $s_{b_1} = 2.80$
- Using 90% confidence level, $t_{0.95;44} = 1.68$
- This gives $b_0 = (148.7, 187.5)$, while $b_1 = (15.5, 25.0)$
- Standard deviation for 9-replica prediction, single observation is $s_e(1.09) = 41.6$
- Using same t, interval is $(308.8, 391.9)$
  - Compare to narrower 1-replica interval

Scatter Plot of “cp” Regression



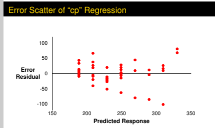


Error Scatter of “cp” Regression

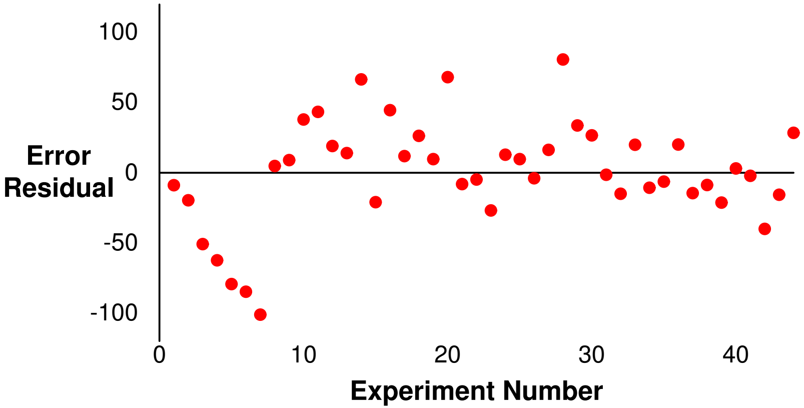


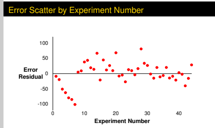
2015-06-15

CS147
└Data Analysis
└Visual Tests
└Error Scatter of “cp” Regression

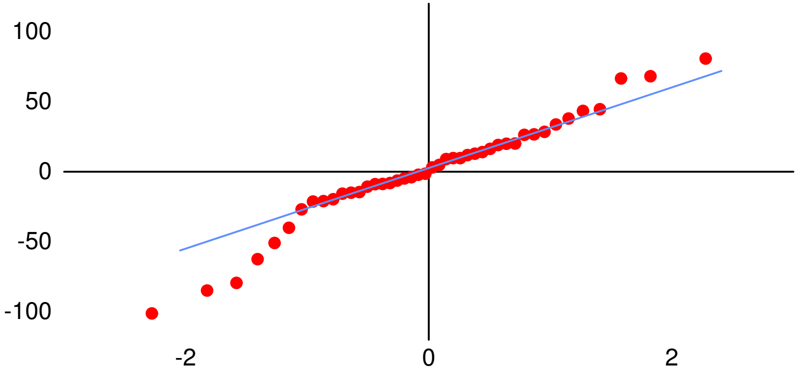


Error Scatter by Experiment Number





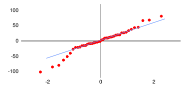
Quantile-Quantile Plot of Errors



2015-06-15

CS147
└Data Analysis
 └Visual Tests
  └Quantile-Quantile Plot of Errors

Quantile-Quantile Plot of Errors



F-test for Ficus “cp” Regression

- SSR = 76590.40, SSE = 61437.46
- MSR $=$ SSR$/k =$ SSR
- MSE $=$ SSE$/(n-k-1) = textSSE/42 = 1462.80$
- Computed F value = MSR$/$MSE $= 52.36$
- Table $F_{0.9;1;42} = 2.83$
  - Regression explains significant part of variation
  - Would have passed F-test at 99% level as well

F-test for Ficus “cp” Regression
- SSR = 76590.40, SSE = 61437.46
- MSR = SSR/k = SSR
- MSE = SSE/(n − k − 1) = textSSE/42 = 1462.80
- Computed F value = MSR/MSE = 52.36
- Table F0.9;1;42 = 2.83
  - Regression explains significant part of variation
  - Would have passed F-test at 99% level as well

Summary of Ficus “cp” Analysis

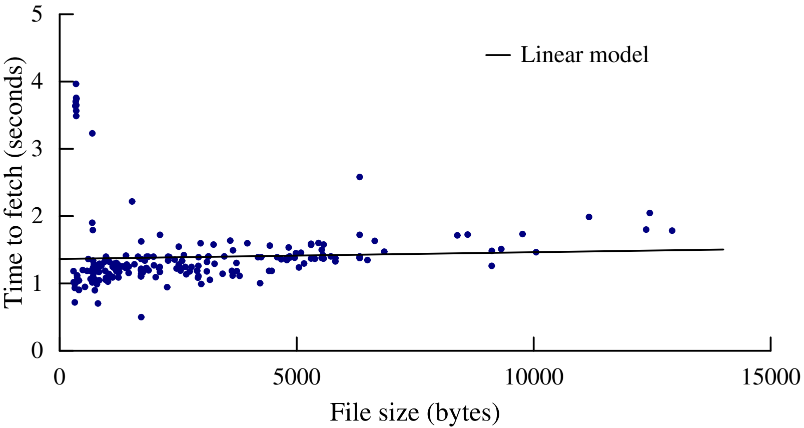
- Ficus costs something compared to UFS, but is much faster than remote access via NFS
- Adding one replica costs around 20 seconds
  - Wide confidence intervals make this uncertain
  - Removing outliers might greatly improve confidence
  - Regression quality questionable (with outliers)

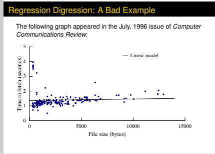
Summary of Ficus “cp” Analysis

- Ficus costs something compared to UFS, but is much faster than remote access via NFS
- Adding one replica costs around 20 seconds
  - Wide confidence intervals make this uncertain
  - Removing outliers might greatly improve confidence
  - Regression quality questionable (with outliers)

Regression Digression: A Bad Example

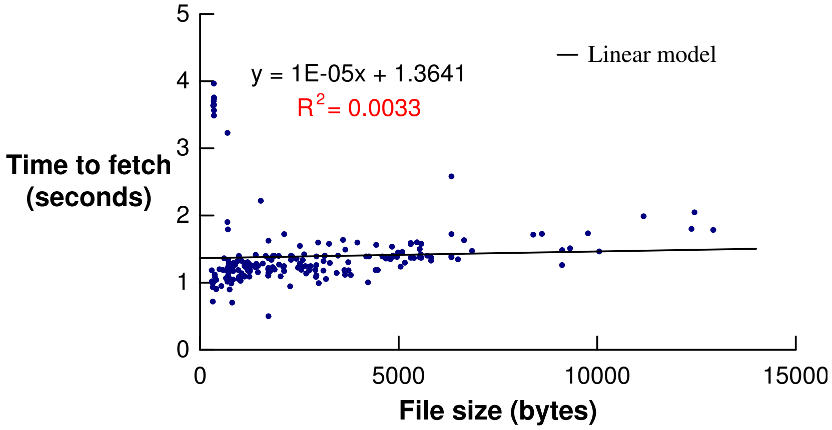
The following graph appeared in the July, 1996 issue of *Computer Communications Review*:

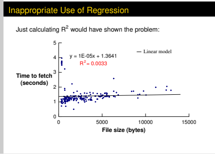




Inappropriate Use of Regression

Just calculating R^2 would have shown the problem:





The Tale of the Residuals

Plot of residuals also shows data isn't homoscedastic: