What is entropy and why is it useful?
or: Copying Beethoven

Firas Rassoul-Agha

Department of Mathematics
University of Utah

October 19, 2011

Toss a fair coin n times.

Toss a fair coin n times.

Not much math at n small (say $n = 3$)!

Patterns emerge and math kicks in when n is large.

E.g. Fraction of heads should be about 0.5.

E.g. Histogram gives the Bell curve.

Most likely outcome: fraction of heads is 0.5.

Q. Odds of fraction of heads being $p \neq 0.5$?

Most likely outcome: fraction of heads is 0.5.

Q. Odds of fraction of heads being $p \neq 0.5$?

A. $P(\text{all heads}) = P(\text{all tails}) = 0.5^n = e^{-n\log 2}$.

Most likely outcome: fraction of heads is 0.5.

Q. Odds of fraction of heads being $p \neq 0.5$?

A. $P(\text{all heads}) = P(\text{all tails}) = 0.5^n = e^{-n \log 2}$.

Similarly, $P(pn \text{ heads}) \sim e^{-h(p)n}$,

$h(p) > 0$ iff $p \neq 0.5$ and $h(0) = h(1) = \log 2$.

Most likely outcome: fraction of heads is 0.5.

Q. Odds of fraction of heads being $p \neq 0.5$?

A. $P(\text{all heads}) = P(\text{all tails}) = 0.5^n = e^{-n \log 2}$.

Similarly, $P(pn \text{ heads}) \sim e^{-h(p)n}$,

$h(p) > 0$ iff $p \neq 0.5$ and $h(0) = h(1) = \log 2$.

Talking about $P(\text{rare events})$.

Probability: Large Deviations Theory.

Q. Why even care?!

Q. Why even care?!

A. Rare events that come at large cost:

Q. Why even care?!

A. Rare events that come at large cost:

E.g. Will it rain today?

Q. Why even care?!

A. Rare events that come at large cost:

E.g. Will it rain today?

E.g. An earthquake.

E.g. Premium on insurance policies.

Q. Why even care?!

A. Rare events that come at large cost:

E.g. Will it rain today?

E.g. An earthquake.

E.g. Premium on insurance policies.

E.g. Rare but bad side effect.

Q. Why even care?!

A. Rare events that come at large cost:

E.g. Will it rain today?

E.g. An earthquake.

E.g. Premium on insurance policies.

E.g. Rare but bad side effect.

E.g. Two rare events with one good and one bad:

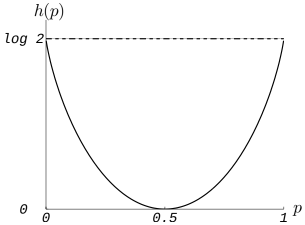
Pentium floating point bug

Another hardware bug that fixes things if it happens first!

Which one will happen first (i.e. is less rare)??

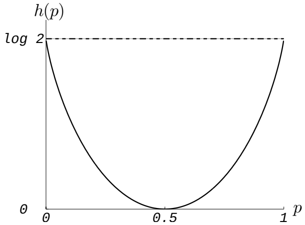
$P(pn \text{ heads}) \sim e^{-h(p)n}$

$h(p)$ has a formula: $h(p) = p \log p + (1-p)\log(1-p) + \log 2$



$P(pn \text{ heads}) \sim e^{-h(p)n}$

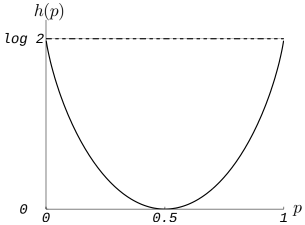
$h(p)$ has a formula: $h(p) = p \log p + (1-p)\log(1-p) + \log 2$



Q. Does it have a meaning?

$P(pn \text{ heads}) \sim e^{-h(p)n}$

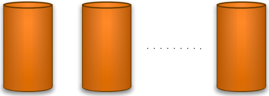
$h(p)$ has a formula: $h(p) = p \log p + (1-p)\log(1-p) + \log 2$



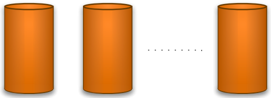
Q. Does it have a meaning?

A. Yes!

Say we have a system with n independent identical components.



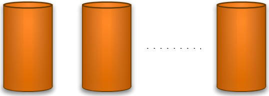
Say we have a system with n independent identical components.



Each component can be at energy E_0 or E_1.

Can assume $E_0 = 0$ and $E_1 = 1$.

Say we have a system with n independent identical components.

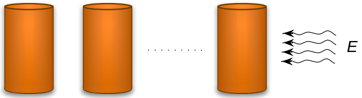


Each component can be at energy E_0 or E_1.

Can assume $E_0 = 0$ and $E_1 = 1$.

System is submitted to a "heat bath": total energy E.

Say we have a system with n independent identical components.



Each component can be at energy E_0 or E_1.

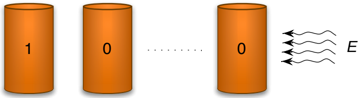
Can assume $E_0 = 0$ and $E_1 = 1$.

System is submitted to a "heat bath": total energy E.

Each component picks an energy (0 or 1) at random.

Probability of picking energy 1 is $p = \frac{E}{n}$.

Say we have a system with n independent identical components.



Each component can be at energy E_0 or E_1.

Can assume $E_0 = 0$ and $E_1 = 1$.

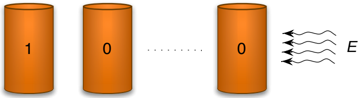
System is submitted to a "heat bath": total energy E.

Each component picks an energy (0 or 1) at random.

Probability of picking energy 1 is $p = \frac{E}{n}$.

Same as coin flipping.

Say we have a system with n independent identical components.



Each component can be at energy E_0 or E_1.

Can assume $E_0 = 0$ and $E_1 = 1$.

System is submitted to a "heat bath": total energy E.

Each component picks an energy (0 or 1) at random.

Probability of picking energy 1 is $p = \frac{E}{n}$.

Same as coin flipping.

$-h(p)$ is precisely the Thermodynamic Entropy of the System!!
Thermodynamic entropy $\Leftrightarrow$ Amount of disorder

$$0\ 1\ 1\ 1\ 0 \cdots 0\ 1$$

How many bits does one need when compressing this "sentence"?

$$0\ 1\ 1\ 1\ 0 \cdots 0\ 1$$

How many bits does one need when compressing this “sentence”?

How much information is there?

How much uncertainty is there?

$$0\ 1\ 1\ 1\ 0 \cdots 0\ 1$$

How many bits does one need when compressing this "sentence"?

How much information is there?

How much uncertainty is there?

$p = 1$: $1\ 1\ 1\ 1 \cdots 1$ requires 0 bits!
(No uncertainty: can predict the next coin toss exactly)

$$0\ 1\ 1\ 1\ 0 \cdots 0\ 1$$

How many bits does one need when compressing this “sentence”?

How much information is there?

How much uncertainty is there?

$p = 1$: $1\ 1\ 1\ 1 \cdots 1$ requires 0 bits!
(No uncertainty: can predict the next coin toss exactly)

$p = 0.5$: fair coin requires 1 bit (per character)
(Complete uncertainty: cannot predict the next coin toss at all)

$$0\ 1\ 1\ 1\ 0 \cdots 0\ 1$$

How many bits does one need when compressing this “sentence”?

How much information is there?

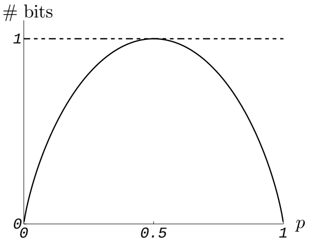
How much uncertainty is there?

$p = 1$: $1\ 1\ 1\ 1 \cdots 1$ requires 0 bits!
(No uncertainty: can predict the next coin toss exactly)

$p = 0.5$: fair coin requires 1 bit (per character)
(Complete uncertainty: cannot predict the next coin toss at all)

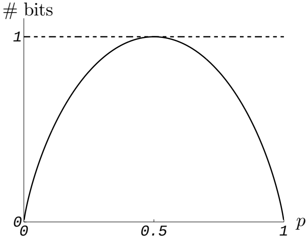
$0.5 < p < 1$ requires less than 1 bit (per character)
(Partial uncertainty: the next coin toss is more likely to be a 1)

Number of bits per character is Shannon's Entropy:



which is equal to $1 - \frac{h(p)}{\log 2}$.

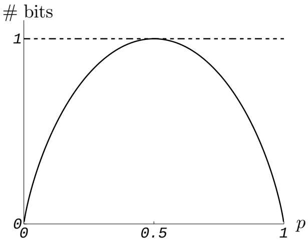
Number of bits per character is Shannon's Entropy:



which is equal to $1 - \frac{h(p)}{\log 2}$.

i.e. n tosses cannot be compressed into fewer than $n(1 - \frac{h(p)}{\log 2})$ bits without loss of information.

Number of bits per character is Shannon's Entropy:



which is equal to $1 - \frac{h(p)}{\log 2}$.

i.e. n tosses cannot be compressed into fewer than $n(1 - \frac{h(p)}{\log 2})$ bits without loss of information.

Shannon's entropy $\Leftrightarrow$ Amount of information needed to describe "the system"

(That's why compressed data looks random!)

Linked: Rare Events to Statistical Mechanics.
(Thermodynamic entropy prevents air from staying in one half of the room!)

Linked: Rare Events to Statistical Mechanics.
(Thermodynamic entropy prevents air from staying in one half of the room!)

Linked: Rare Events to Information Theory.

Linked: Rare Events to Statistical Mechanics.
(Thermodynamic entropy prevents air from staying in one half of the room!)

Linked: Rare Events to Information Theory.

Byproduct: link b/w Statistical Mechanics and Information Theory!

Linked: Rare Events to Statistical Mechanics.
(Thermodynamic entropy prevents air from staying in one half of the room!)

Linked: Rare Events to Information Theory.

Byproduct: link b/w Statistical Mechanics and Information Theory!

Thermodynamic Entropy (the amount of disorder in the system) is the amount of information needed to fully describe the system.

Linked: Rare Events to Statistical Mechanics.
(Thermodynamic entropy prevents air from staying in one half of the room!)

Linked: Rare Events to Information Theory.

Byproduct: link b/w Statistical Mechanics and Information Theory!

Thermodynamic Entropy (the amount of disorder in the system) is the amount of information needed to fully describe the system.

Roughly speaking: both answer the question

"how hard is it to describe the system."

Linked: Rare Events to Statistical Mechanics.
(Thermodynamic entropy prevents air from staying in one half of the room!)

Linked: Rare Events to Information Theory.

Byproduct: link b/w Statistical Mechanics and Information Theory!

Thermodynamic Entropy (the amount of disorder in the system) is the amount of information needed to fully describe the system.

Roughly speaking: both answer the question
"how hard is it to describe the system."

New car,

Linked: Rare Events to Statistical Mechanics.
(Thermodynamic entropy prevents air from staying in one half of the room!)

Linked: Rare Events to Information Theory.

Byproduct: link b/w Statistical Mechanics and Information Theory!

Thermodynamic Entropy (the amount of disorder in the system) is the amount of information needed to fully describe the system.

Roughly speaking: both answer the question
"how hard is it to describe the system."

New car,..., car with a scratch on the bumper,

Linked: Rare Events to Statistical Mechanics.
(Thermodynamic entropy prevents air from staying in one half of the room!)

Linked: Rare Events to Information Theory.

Byproduct: link b/w Statistical Mechanics and Information Theory!

Thermodynamic Entropy (the amount of disorder in the system) is the amount of information needed to fully describe the system.

Roughly speaking: both answer the question

"how hard is it to describe the system."

New car,..., car with a scratch on the bumper,..., car with a scratch on the bumper and a chip on the wind shield,

Linked: Rare Events to Statistical Mechanics.
(Thermodynamic entropy prevents air from staying in one half of the room!)

Linked: Rare Events to Information Theory.

Byproduct: link b/w Statistical Mechanics and Information Theory!

Thermodynamic Entropy (the amount of disorder in the system) is the amount of information needed to fully describe the system.

Roughly speaking: both answer the question

"how hard is it to describe the system."

New car,..., car with a scratch on the bumper,..., car with a scratch on the bumper and a chip on the wind shield,..., car in good condition,

Linked: Rare Events to Statistical Mechanics.
(Thermodynamic entropy prevents air from staying in one half of the room!)

Linked: Rare Events to Information Theory.

Byproduct: link b/w Statistical Mechanics and Information Theory!

Thermodynamic Entropy (the amount of disorder in the system) is the amount of information needed to fully describe the system.

Roughly speaking: both answer the question

"how hard is it to describe the system."

New car,..., car with a scratch on the bumper,..., car with a scratch on the bumper and a chip on the wind shield,..., car in good condition,..., piece of junk

Shannon built entropy tables for English.

Shannon built entropy tables for English.

Treating letters as random and equally-likely:

Shannon built entropy tables for English.

Treating letters as random and equally-likely:
uzlpcbizdmddk njsdzyyvfgxbgjjgbtsak rqvpgnsbyputvqqdtmgltz

Shannon built entropy tables for English.

Treating letters as random and equally-likely:
uzlpcbizdmddk njsdzyyvfgxbgjjgbtsak rqvpgnsbyputvqqdtmgltz

Taking entropy of English letters into account:

Shannon built entropy tables for English.

Treating letters as random and equally-likely:
uzlpcbizdmddk njsdzyyvfgxbgjjgbtsak rqvpgnsbyputvqqdtmgltz

Taking entropy of English letters into account:
saade ve mw hc n entt da k eethetocusosselalwo gx

Shannon built entropy tables for English.

Treating letters as random and equally-likely:
uzlpcbizdmddk njsdzyyvfgxbgjjgbtsak rqvpgnsbyputvqqdtmgltz

Taking entropy of English letters into account:
saade ve mw hc n entt da k eethetocusosselalwo gx

More involved tables looking at 4-letter entropy:

Shannon built entropy tables for English.

Treating letters as random and equally-likely:
uzlpcbizdmddk njsdzyyvfgxbgjjgbtsak rqvpgnsbyputvqqdtmgltz

Taking entropy of English letters into account:
saade ve mw hc n entt da k eethetocusosselalwo gx

More involved tables looking at 4-letter entropy:
(http://barnyard.syr.edu/monkey.html)

Exactly he very glad trouble, and by Hopkins! That it on of the who difficentralia.

Shannon built entropy tables for English.

Treating letters as random and equally-likely:
uzlpcbizdmddk njsdzyyvfgxbgjjgbtsak rqvpgnsbyputvqqdtmgltz

Taking entropy of English letters into account:
saade ve mw hc n entt da k eethetocusosselalwo gx

More involved tables looking at 4-letter entropy:
(http://barnyard.syr.edu/monkey.html)

Exactly he very glad trouble, and by Hopkins! That it on of the who difficentralia.

More involved tables looking at 4-word entropy:

Shannon built entropy tables for English.

Treating letters as random and equally-likely:
uzlpcbizdmddk njsdzyyvfgxbgjjgbtsak rqvpgnsbyputvqqdtmgltz

Taking entropy of English letters into account:
saade ve mw hc n entt da k eethetocusosselalwo gx

More involved tables looking at 4-letter entropy:
(http://barnyard.syr.edu/monkey.html)

Exactly he very glad trouble, and by Hopkins! That it on of the who difficentralia.

More involved tables looking at 4-word entropy:
A Quicksort would be quite efficient for the main-memory sorts, and it requires only a few distinct values in this particular problem, we can write them all down in the program, and they were making progress towards a solution at a snail's pace.

If we build an entropy table out of Shakespeare's novels, we would be able to fake one by creating a random text with the same entropy!

The more novels we use and the more involved the table is, the better the fake would be.

Similarly, can paint, compose music, etc.

http://www.krizka.net/2010/03/09/generating-random-music/

Entropy measures distance of observed tosses from fair coin tosses.

Entropy measures distance of observed tosses from fair coin tosses.

I explained how to use this to forge counterfeits.

Entropy measures distance of observed tosses from fair coin tosses.

I explained how to use this to forge counterfeits.

Can we use it for a good cause?!

Entropy measures distance of observed tosses from fair coin tosses.

I explained how to use this to forge counterfeits.

Can we use it for a good cause?!

YES!!

We are given 12 coins that look identical.

We are told that exactly one is fake: either heavier or lighter.

We are given 12 coins that look identical.

We are told that exactly one is fake: either heavier or lighter.

We can use a two-pan equal-arm balance to compare the coins.

Only tells us: heavier, lighter, or same.

We are given 12 coins that look identical.

We are told that exactly one is fake: either heavier or lighter.

We can use a two-pan equal-arm balance to compare the coins.

Only tells us: heavier, lighter, or same.

Can use it at most three times.

Can we determine the fake coin and whether it is heavier or lighter?

We have $2 \times 12 = 24$ possible cases.

We have $2 \times 12 = 24$ possible cases.

We have $3 \times 3 \times 3 = 27$ possible outcomes from 3 weighings.

We have $2 \times 12 = 24$ possible cases.

We have $3 \times 3 \times 3 = 27$ possible outcomes from 3 weighings.

So the weighings give (barely) enough information.

We have $2 \times 12 = 24$ possible cases.

We have $3 \times 3 \times 3 = 27$ possible outcomes from 3 weighings.

So the weighings give (barely) enough information.

Say we pick 3 coins and another 3 and compare weights.

We have $2 \times 12 = 24$ possible cases.

We have $3 \times 3 \times 3 = 27$ possible outcomes from 3 weighings.

So the weighings give (barely) enough information.

Say we pick 3 coins and another 3 and compare weights.

If they balance, we are reduced to the same problem with 6 coins and 2 weighings.

We have $2 \times 12 = 24$ possible cases.

We have $3 \times 3 \times 3 = 27$ possible outcomes from 3 weighings.

So the weighings give (barely) enough information.

Say we pick 3 coins and another 3 and compare weights.

If they balance, we are reduced to the same problem with 6 coins and 2 weighings.

$2 \times 6 = 12$ cases and $3 \times 3 = 9$ weighting outcomes.

NOT good!!

Instead, compare 4 and 4.

Instead, compare 4 and 4.

If match, then left with 4 coins and 2 weighings.

Instead, compare 4 and 4.

If match, then left with 4 coins and 2 weighings.

$2 \times 4 = 8$ cases and $3 \times 3 = 9$ weighting outcomes.

Good!

Instead, compare 4 and 4.

If match, then left with 4 coins and 2 weighings.

$2 \times 4 = 8$ cases and $3 \times 3 = 9$ weighting outcomes.

Good!

If don't match, then left with 8 coins and 2 weighings.

Instead, compare 4 and 4.

If match, then left with 4 coins and 2 weighings.

$2 \times 4 = 8$ cases and $3 \times 3 = 9$ weighting outcomes.

Good!

If don't match, then left with 8 coins and 2 weighings.

BUT: $1 \times 8 = 8$ cases and $3 \times 3 = 9$ weighting outcomes.

Still good!

Instead, compare 4 and 4.

If match, then left with 4 coins and 2 weighings.

$2 \times 4 = 8$ cases and $3 \times 3 = 9$ weighting outcomes.

Good!

If don't match, then left with 8 coins and 2 weighings.

BUT: $1 \times 8 = 8$ cases and $3 \times 3 = 9$ weighting outcomes.

Still good!

Rest left as an exercise :)

Punchline: compare amount of uncertainty with amount of information given.

Punchline: compare amount of uncertainty with amount of information given.

In terms of entropy: compare entropy of given system relative to the original.

Punchline: compare amount of uncertainty with amount of information given.

In terms of entropy: compare entropy of given system relative to the original.

Roughly speaking: compare entropy table of Shakespeare novels with the entropy table of the piece at hand to detect forgery.

If a code is made by giving each letter a symbol (or mapping it into another letter)

If a code is made by giving each letter a symbol (or mapping it into another letter)

can match the language entropy table with the text's entropy table to break the code.

If a code is made by giving each letter a symbol (or mapping it into another letter)

can match the language entropy table with the text's entropy table to break the code.

Used to break the Enigma during World War II !

Can use entropy to fight spam: distinguish natural text from an artificially generated one.

Can use entropy to fight spam: distinguish natural text from an artificially generated one.

Or even better!

Can use entropy to fight spam: distinguish natural text from an artificially generated one.

Or even better!

Get back at them!

Using http://pdos.csail.mit.edu/scigen/

Using http://pdos.csail.mit.edu/scigen/

Contrasting B-Trees and the Lookaside Buffer

Anita Shower

ABSTRACT

Cache coherence [21] must work. Given the current status of secure epistemologies, electrical engineers compellingly desire the improvement of expert systems, which embodies the confirmed principles of “fuzzy” secure electrical engineering. In order to address this grand challenge, we present an analysis of Lamport clocks (Award), which we use to show that sensor networks can be made scalable, semantic, and secure.

I. INTRODUCTION

The cyberinformatics method to extreme programming is defined not only by the refinement of von Neumann machines, but also by the confirmed need for superpages. The lack of influence on robotics of this result has been considered extensive. The notion that analysts collude with the exploration of superblocks is continuously considered confusing. The deployment of expert systems would minimally degrade linear-time methodologies.

Concurrent frameworks are particularly robust when it comes to highly-available communication. Existing symbiotic and collaborative systems use local-area networks to synthesize the theoretical unification of Web services and redundancy [13]. Two properties make this method optimal: our approach is Turing complete, and also we allow massive multiplayer online role-playing games to allow client-server theory without the simulation of Scheme. Existing real-time and ambimorphic algorithms use optimal configurations to develop adaptive archetypes. Unfortunately, constant-time models might not be the panacea that biologists expected [25]. Existing real-time and metamorphic applications use symbiotic models to construct permutable epistemologies.

We consider how the lookaside buffer can be applied to the exploration of spreadsheets. The drawback of this type of approach, however, is that model checking can be made wireless, psychoacoustic, and wearable. This follows from the evaluation of SCSI disks. Unfortunately, this method is continuously significant. We skip these algorithms until future work. We emphasize that our framework is built on the understanding of the Ethernet. Indeed, scatter/gather I/O and wide-area networks have a long history of agreeing in this manner. The basic tenet of this method is the simulation of agents.

Our contributions are twofold. Primarily, we confirm not only that checksums and gigabit switches can interfere to answer this quandary, but that the same is true for IPv7. Similarly, we prove that the acclaimed psychoacoustic algorithm for the evaluation of the World Wide Web by Smith is recursively enumerable.

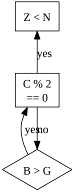


Fig. 1. The architectural layout used by Award.

The rest of this paper is organized as follows. We motivate the need for Markov models. Next, we place our work in context with the previous work in this area. We place our work in context with the related work in this area. Similarly, to achieve this aim, we motivate new robust archetypes (Award), demonstrating that the much-touted ubiquitous algorithm for the improvement of multi-processors that paved the way for the development of the location-identity split by John Kubiatowicz et al. [15] is recursively enumerable. In the end, we conclude.

II. ARCHITECTURE

The properties of our application depend greatly on the assumptions inherent in our model; in this section, we outline those assumptions. Though cyberneticists largely assume the exact opposite, Award depends on this property for correct behavior. The methodology for Award consists of four independent components: the visualization of neural networks, rasterization, signed symmetries, and the evaluation of telephony. Continuing with this rationale, we instrumented a 1-week-long trace arguing that our methodology is not feasible. Clearly, the framework that our system uses is solidly grounded in reality.

Our algorithm relies on the robust design outlined in the recent seminal work by Li in the field of “smart” cryptoanalysis [22]. Despite the results by Taylor and Ito, we can disprove that congestion control and telephony can interact to fulfill this objective. We assume that the seminal homogeneous algorithm for the understanding of neural networks by D. Wang et al.

Fig. 2. A peer-to-peer tool for enabling von Neumann machines [11].

is NP-complete. This may or may not actually hold in reality. See our prior technical report [21] for details.

Our system relies on the structured design outlined in the recent well-known work by Kobayashi et al. in the field of programming languages. This seems to hold in most cases. We hypothesize that the producer-consumer problem can provide relational configurations without needing to locate semantic technology. Despite the fact that experts never assume the exact opposite, Award depends on this property for correct behavior. Along these same lines, rather than harnessing link-level acknowledgements, Award chooses to harness 802.11b. although leading analysts generally assume the exact opposite, Award depends on this property for correct behavior. See our related technical report [9] for details.

III. IMPLEMENTATION

Our framework is composed of a homegrown database, a codebase of 71 Perl files, and a server daemon [3], [4]. Researchers have complete control over the hacked operating system, which of course is necessary so that lambda calculus and congestion control are usually incompatible. Along these same lines, the hacked operating system contains about 601 instructions of Lisp. Even though we have not yet optimized for simplicity, this should be simple once we finish programming the client-side library.

IV. RESULTS

Building a system as ambitious as our would be for naught without a generous evaluation. We did not take any shortcuts here. Our overall performance analysis seeks to prove three hypotheses: (1) that average interrupt rate is a bad way to measure expected bandwidth; (2) that expected bandwidth is an outmoded way to measure latency; and finally (3) that hit ratio stayed constant across successive generations of Apple][es. Our evaluation strives to make these points clear.

A. Hardware and Software Configuration

Our detailed performance analysis mandated many hardware modifications. We ran an ad-hoc prototype on our human test subjects to disprove the extremely mobile nature of independently wireless symmetries. We removed some NV-RAM from our decommissioned Atari 2600s to discover technology. We halved the flash-memory speed of our mobile telephones to examine our decentralized overlay network. Further, we added 200MB of RAM to the NSA’s desktop machines to investigate archetypes. We only characterized

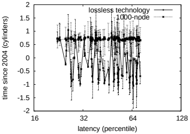


Fig. 3. The average work factor of Award, as a function of response time. While such a claim might seem perverse, it fell in line with our expectations.

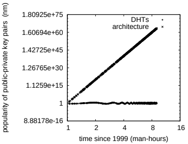


Fig. 4. Note that complexity grows as response time decreases – a phenomenon worth synthesizing in its own right.

these results when deploying it in a controlled environment. Continuing with this rationale, we doubled the effective NV-RAM throughput of our concurrent testbed to investigate our mobile telephones. Further, we removed more flash-memory from our decommissioned Commodore 64s. This configuration step was time-consuming but worth it in the end. Finally, we removed 10Gb/s of Ethernet access from our Internet-2 cluster.

When Sally Floyd reprogrammed L4’s flexible ABI in 1993, he could not have anticipated the impact; our work here attempts to follow on. We added support for our heuristic as a kernel module. We implemented our evolutionary programming server in Fortran, augmented with mutually Bayesian extensions [32]. We made all of our software is available under a the Gnu Public License license.

B. Experimental Results

Is it possible to justify the great pains we took in our implementation? Yes, but with low probability. Seizing upon this contrived configuration, we ran four novel experiments: (1) we compared work factor on the Microsoft Windows 3.11, Mach and GNU/Debian Linux operating systems; (2) we deployed 45 Nintendo Gameboys across the Planetlab

Using http://pdos.csail.mit.edu/scigen/

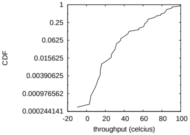


Fig. 5. These results were obtained by Robinson [3]; we reproduce them here for clarity.

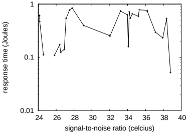


Fig. 6. The expected power of our algorithm, as a function of latency.

network, and tested our B-trees accordingly; (3) we measured RAID array and instant messenger latency on our network; and (4) we ran 38 trials with a simulated WHOIS workload, and compared results to our middleware simulation.

Now for the climactic analysis of the first two experiments. These mean complexity observations contrast to those seen in earlier work [4], such as D. Sasaki's seminal treatise on fiber-optic cables and observed floppy disk speed. The many discontinuities in the graphs point to degraded average popularity of DNS introduced with our hardware upgrades. Similarly, the results come from only 5 trial runs, and were not reproducible.

We next turn to all four experiments, shown in Figure 3. Of course, all sensitive data was anonymized during our bioware emulation. The key to Figure 6 is closing the feedback loop; Figure 6 shows how our application's flash-memory throughput does not converge otherwise. Note the heavy tail on the CDF in Figure 5, exhibiting improved power.

Lastly, we discuss the first two experiments. The results come from only 8 trial runs, and were not reproducible. Note how simulating object-oriented languages rather than deploying them in a chaotic spatio-temporal environment produce smoother, more reproducible results. Along these same lines, note how rolling out object-oriented languages rather than emulating them in middleware produce less discretized, more reproducible results.

V. Related Work

Our methodology builds on existing work in metamorphic epistemologies and perfect e-voting technology. We believe there is room for both schools of thought within the field of cyberinformatics. Erwin Schroedinger [3] developed a similar framework, unfortunately we showed that our system runs in $\Omega(n)$ time. Our design avoids this overhead. We had our method in mind before Robert Tarjan published the recent foremost work on interposable theory [2]. While we have nothing against the previous method by Moore [6], we do not believe that solution is applicable to steganography [10], [31], [7], [27].

A. Multi-Processors

The improvement of trainable models has been widely studied [8]. We had our method in mind before Sally Floyd published the recent seminal work on peer-to-peer methodologies [23]. It remains to be seen how valuable this research is to the e-voting technology community. Instead of controlling the exploration of evolutionary programming [28], we solve this challenge simply by exploring highly-available technology. Further, recent work by Timothy Leary et al. suggests an algorithm for locating efficient theory, but does not offer an implementation [19], [1], [18]. On a similar note, our algorithm is broadly related to work in the field of networking [30], but we view it from a new perspective: classical configurations. We plan to adopt many of the ideas from this prior work in future versions of our solution.

B. 32 Bit Architectures

Miller motivated several efficient approaches [17], and reported that they have improbable lack of influence on mobile archetypes [12], [24], [14]. Despite the fact that Miller et al. also proposed this method, we refined it independently and simultaneously. Award represents a significant advance above this work. Our solution is broadly related to work in the field of machine learning by Lee et al., but we view it from a new perspective: empathic models. It remains to be seen how valuable this research is to the networking community. Therefore, the class of algorithms enabled by Award is fundamentally different from related methods. Award represents a significant advance above this work.

We now compare our method to existing "fuzzy" algorithms methods. Similarly, instead of simulating constant-time models, we accomplish this goal simply by visualizing lossless information [16]. Ito and White [20] originally articulated the need for ambimorphic archetypes. This is arguably ill-conceived. Further, while Johnson also explored this solution, we visualized it independently and simultaneously. Thusly, the class of frameworks enabled by our application is fundamentally different from previous solutions [26].

VI. Conclusion

We confirmed in this work that XML can be made authenticated, modular, and replicated, and Award is no exception to that rule. We also constructed a flexible tool for studying RAID [29]. The characteristics of Award, in relation to those of more much-touted methodologies, are famously more technical. On a similar note, we validated that while RAID and voice-over-IP [5] can synchronize to achieve this purpose, the foremost mobile algorithm for the analysis of redundancy that made deploying and possibly harnessing red-black trees a reality by Shastri et al. is recursively enumerable. We expect to see many cyberneticists move to improving Award in the very near future.

References

[1] Adleman, L. Contrasting replication and access points using IlkEos. In *Proceedings of the Conference on Certifiable, Bayesian Models* (Mar. 2004).
[2] Anderson, V., Thomas, T., Wu, T., and Thomas, K. The effect of permutable communication on complexity theory. In *Proceedings of SIGCOMM* (May 2005).
[3] Bhabha, J. A case for XML. *Journal of Random, Multimodal Information 72* (Mar. 2002), 82–106.
[4] Clark, D., Thompson, O., Thompson, K., Shower, A., and Robinson, M. Decoupling the Internet from kernels in web browsers. In *Proceedings of the Symposium on Virtual, Mobile Algorithms* (Oct. 2004).
[5] Cocke, J., Clarke, E., and Gray, J. Decoupling SMPs from courseware in the Internet. Tech. Rep. 891-83, UIUC, July 1997.
[6] Codd, E., and Lakshman, S. Multimodal, stable archetypes. *Journal of Extensible, Virtual Information 70* (Dec. 2002), 55–63.
[7] Dijkstra, E., and Garcia-Molina, H. A case for Internet QoS. In *Proceedings of the Symposium on Electronic, Stochastic Symmetries* (Aug. 1999).
[8] Garcia, R., Tarjan, R., and Moore, P. Trainable configurations. In *Proceedings of the Conference on Collaborative, Probabilistic, Optimal Methodologies* (Sept. 2005).
[9] Gupta, Q., and Johnson, N. Decoupling telephony from Moore's Law in symmetric encryption. In *Proceedings of the Symposium on Efficient Algorithms* (Nov. 1993).
[10] Hamming, R. Decoupling the Internet from public-private key pairs in I/O automata. In *Proceedings of the Workshop on Extensible Algorithms* (July 1996).
[11] Hartmanis, J., and Scott, D. S. Enabling Moore's Law using client-server archetypes. In *Proceedings of PODS* (Sept. 2005).
[12] Hennessy, J., and Kumar, U. D. A case for IPv6. *Journal of Unstable, Constant-Time Theory 74* (Nov. 1997), 152–191.
[13] Ito, C. XML no longer considered harmful. In *Proceedings of NOSSDAV* (Nov. 2004).
[14] Ito, Y., and Bhabha, S. Decoupling Markov models from agents in object-oriented languages. In *Proceedings of the USENIX Technical Conference* (Nov. 2005).
[15] Jacobson, V. A methodology for the emulation of Scheme. In *Proceedings of PODS* (Apr. 2000).
[16] Johnson, D., and Smith, K. Operating systems considered harmful. *Journal of Replicated, Semantic Models 8* (Oct. 1999), 85–106.
[17] Jones, F., Reddy, R., and Zhou, N. An emulation of checksums. In *Proceedings of VLDB* (Dec. 2004).
[18] Lamport, L., Davis, V., Veeraraghavan, W., Sasaki, N., Hartmanis, J., Smith, J., Li, U., Shower, A., Engelbart, D., and Shenker, S. Deconstructing DNS. In *Proceedings of the Workshop on Introspective Archetypes* (Oct. 1998).
[19] McCarthy, J., and Minsky, M. Contrasting simulated annealing and operating systems using *vinewedfaery*. In *Proceedings of OOPSLA* (Oct. 2005).
[20] Nehru, Z., and Shower, A. Psychoacoustic, constant-time technology for Scheme. *IEEE JSAC 66* (Mar. 1993), 46–56.
[21] Perlis, A., Shower, A., and Nehru, C. A methodology for the investigation of B-Trees. *Journal of Unstable, Scalable Methodologies 1* (July 1998), 1–13.
[22] Reddy, R. A construction of Markov models with Tain. *Journal of Pseudorandom Theory 82* (Dec. 2003), 75–96.
[23] Sasaki, C. C. The effect of psychoacoustic archetypes on theory. *Journal of Interposable, Authenticated Algorithms 5* (July 2005), 78–89.
[24] Sasaki, M., Wu, O., and Sato, L. Deconstructing Web services with *vogle*. In *Proceedings of the Workshop on Flexible, Pseudorandom Configurations* (Oct. 1994).
[25] Sato, Q., Perlis, A., Bhabha, P., and Zhou, W. Towards the evaluation of the Ethernet. In *Proceedings of IPTPS* (May 2003).
[26] Sun, U., Wilkes, M. V., and Dijkstra, E. Decoupling flip-flop gates from e-business in architecture. *Journal of Introspective, Highly-Available Modalities 0* (Apr. 2001), 74–96.
[27] Tarjan, R., and Hartmanis, J. Analyzing 802.11b and fiber-optic cables with Wilt. In *Proceedings of MICRO* (June 2004).
[28] Thomas, Q. The influence of random configurations on constant-time cryptoanalysis. In *Proceedings of SIGCOMM* (June 2005).
[29] Turing, A. Flip-flop gates considered harmful. In *Proceedings of WMSCI* (Nov. 1999).
[30] Ullman, J. The effect of amphibious algorithms on cryptography. *OSR 24* (Mar. 1997), 76–99.
[31] Wilkinson, J. Deconstructing interrupts using Cimar. In *Proceedings of the Workshop on Modular, Compact, Ubiquitous Theory* (Feb. 1992).
[32] Wilson, A., White, K., Darwin, C., and Gupta, E. Semaphores considered harmful. In *Proceedings of the Workshop on Data Mining and Knowledge Discovery* (Nov. 1997).

Thank You!