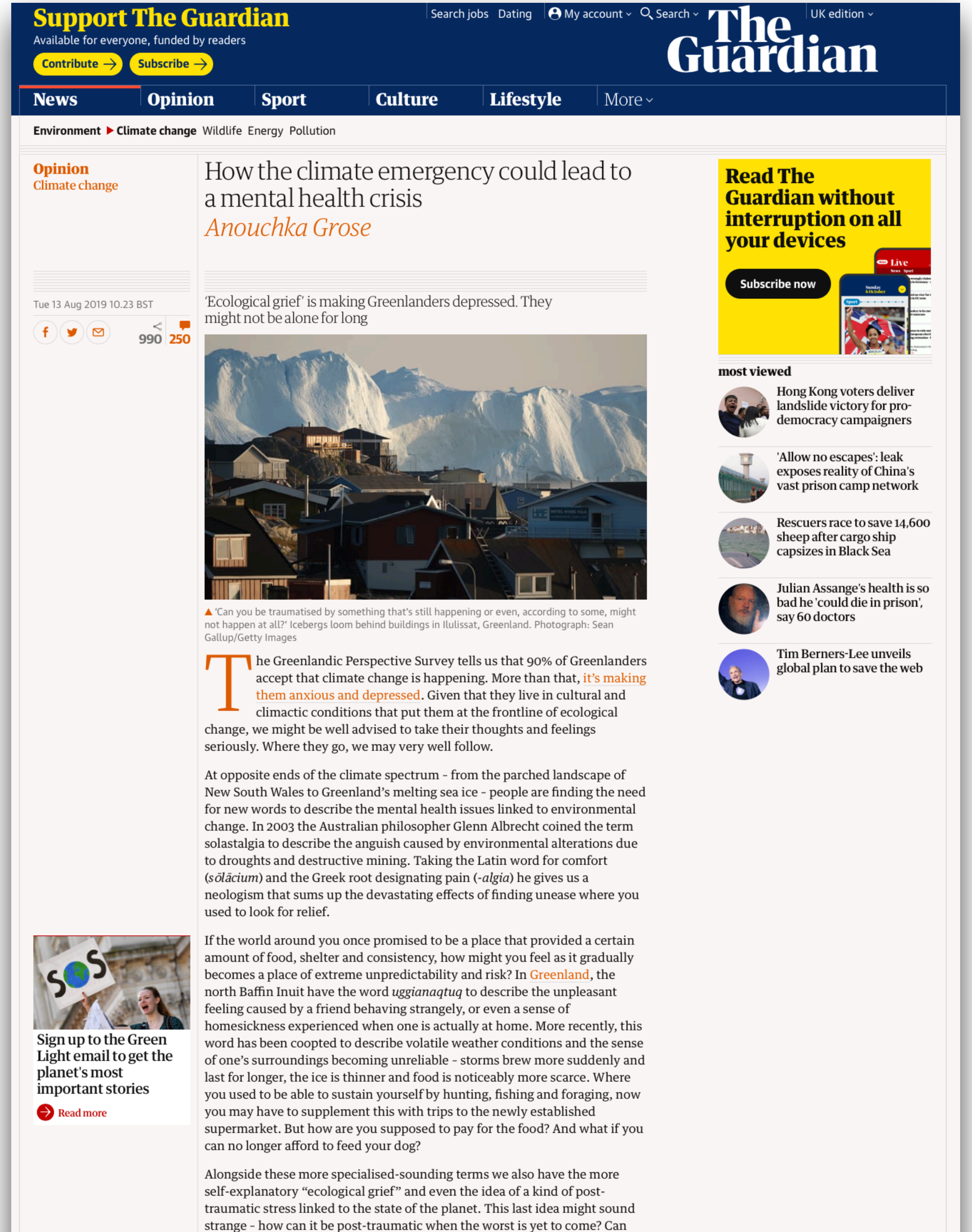
Environmental Impacts

CS489

Shin Yoo



Support The Guardian
Available for everyone, funded by readers
Contribute → Subscribe →

Search jobs Dating My account Search UK edition

The Guardian

News Opinion Sport Culture Lifestyle More

Environment ▸ Climate change Wildlife Energy Pollution

Opinion
Climate change

How the climate emergency could lead to a mental health crisis

Anouchka Grose

Tue 13 Aug 2019 10.23 BST

990 250

'Ecological grief' is making Greenlanders depressed. They might not be alone for long

▲ 'Can you be traumatised by something that's still happening or even, according to some, might not happen at all?' Icebergs loom behind buildings in Ilulissat, Greenland. Photograph: Sean Gallup/Getty Images

The Greenlandic Perspective Survey tells us that 90% of Greenlanders accept that climate change is happening. More than that, it's making them anxious and depressed. Given that they live in cultural and climactic conditions that put them at the frontline of ecological change, we might be well advised to take their thoughts and feelings seriously. Where they go, we may very well follow.

At opposite ends of the climate spectrum - from the parched landscape of New South Wales to Greenland's melting sea ice - people are finding the need for new words to describe the mental health issues linked to environmental change. In 2003 the Australian philosopher Glenn Albrecht coined the term solastalgia to describe the anguish caused by environmental alterations due to droughts and destructive mining. Taking the Latin word for comfort (*sōlācium*) and the Greek root designating pain (*-algia*) he gives us a neologism that sums up the devastating effects of finding unease where you used to look for relief.

If the world around you once promised to be a place that provided a certain amount of food, shelter and consistency, how might you feel as it gradually becomes a place of extreme unpredictability and risk? In Greenland, the north Baffin Inuit have the word *uggianaqtuq* to describe the unpleasant feeling caused by a friend behaving strangely, or even a sense of homesickness experienced when one is actually at home. More recently, this word has been coopted to describe volatile weather conditions and the sense of one's surroundings becoming unreliable - storms brew more suddenly and last for longer, the ice is thinner and food is noticeably more scarce. Where you used to be able to sustain yourself by hunting, fishing and foraging, now you may have to supplement this with trips to the newly established supermarket. But how are you supposed to pay for the food? And what if you can no longer afford to feed your dog?

Alongside these more specialised-sounding terms we also have the more self-explanatory "ecological grief" and even the idea of a kind of post-traumatic stress linked to the state of the planet. This last idea might sound strange - how can it be post-traumatic when the worst is yet to come? Can

Sign up to the Green Light email to get the planet's most important stories
Read more

Read The Guardian without interruption on all your devices
Subscribe now

most viewed

Hong Kong voters deliver landslide victory for pro-democracy campaigners

'Allow no escapes': leak exposes reality of China's vast prison camp network

Rescuers race to save 14,600 sheep after cargo ship capsizes in Black Sea

Julian Assange's health is so bad he 'could die in prison', say 60 doctors

Tim Berners-Lee unveils global plan to save the web

90% of Greenlanders feel anxiety and depression caused by climate change.

https://www.theguardian.com/commentisfree/2019/aug/13/climate-crisis-mental-health-environmental-anguish

What can we do as software engineers / computer scientists?

Develop energy efficient systems.

How?

Moore’s Law

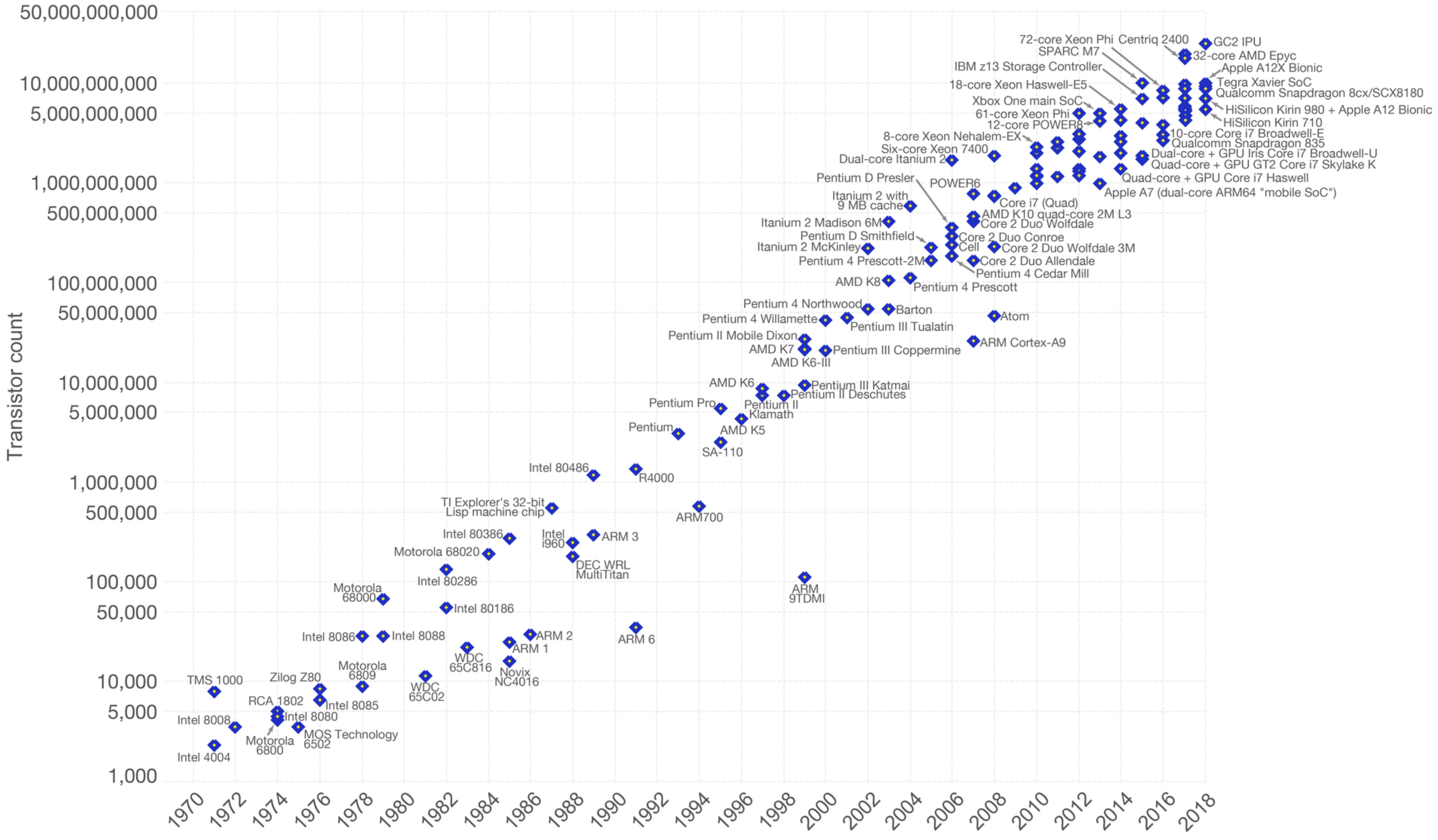
- An observation that the number of transistors in a dense integrated circuits doubles roughly every two years (named after Gordon Moore, ex-CEO of Intel)

- Moore initially claimed double every year in 1965.

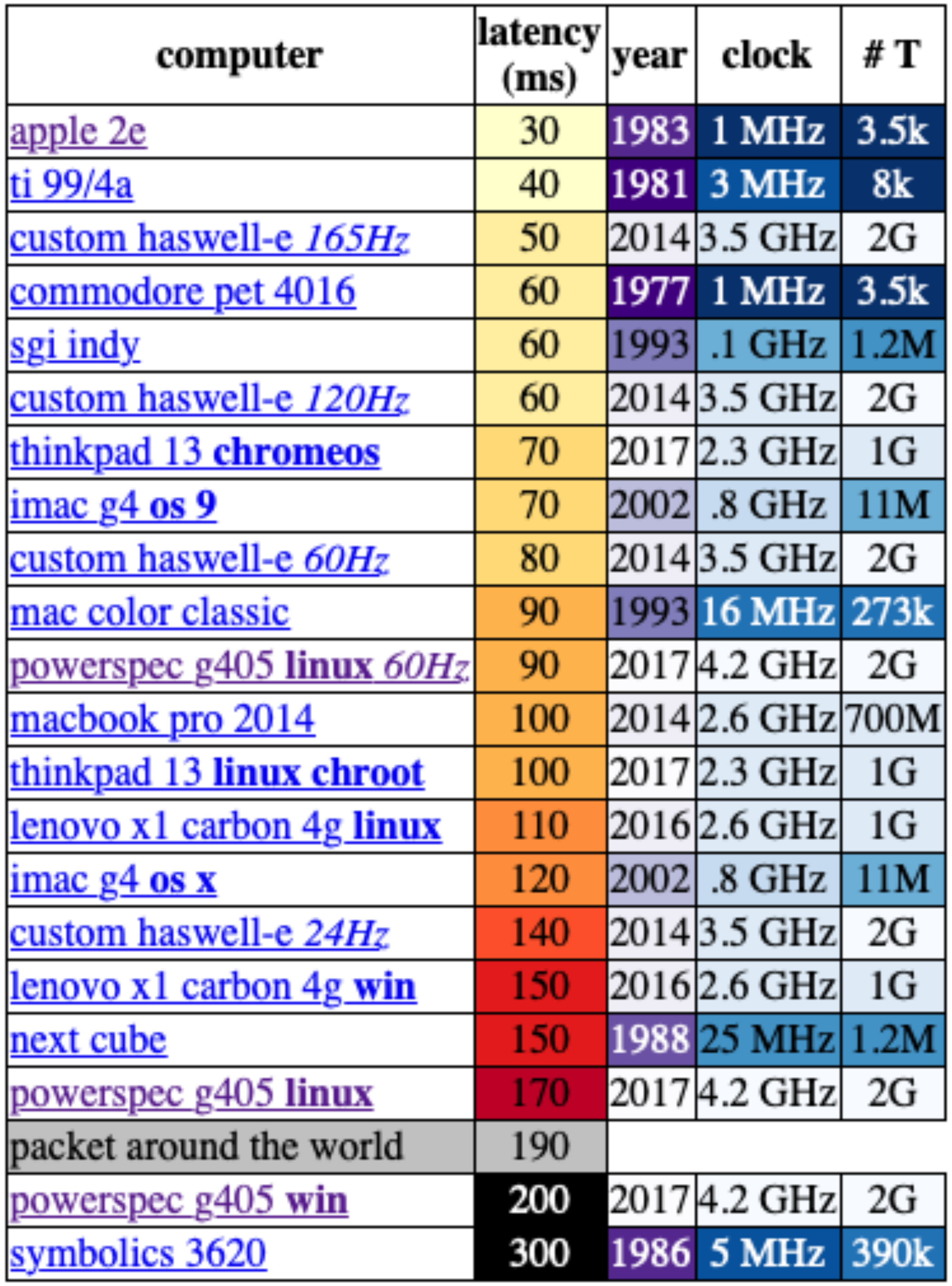
Moore's Law – The number of transistors on integrated circuit chips (1971-2018)

Our World in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore's law.



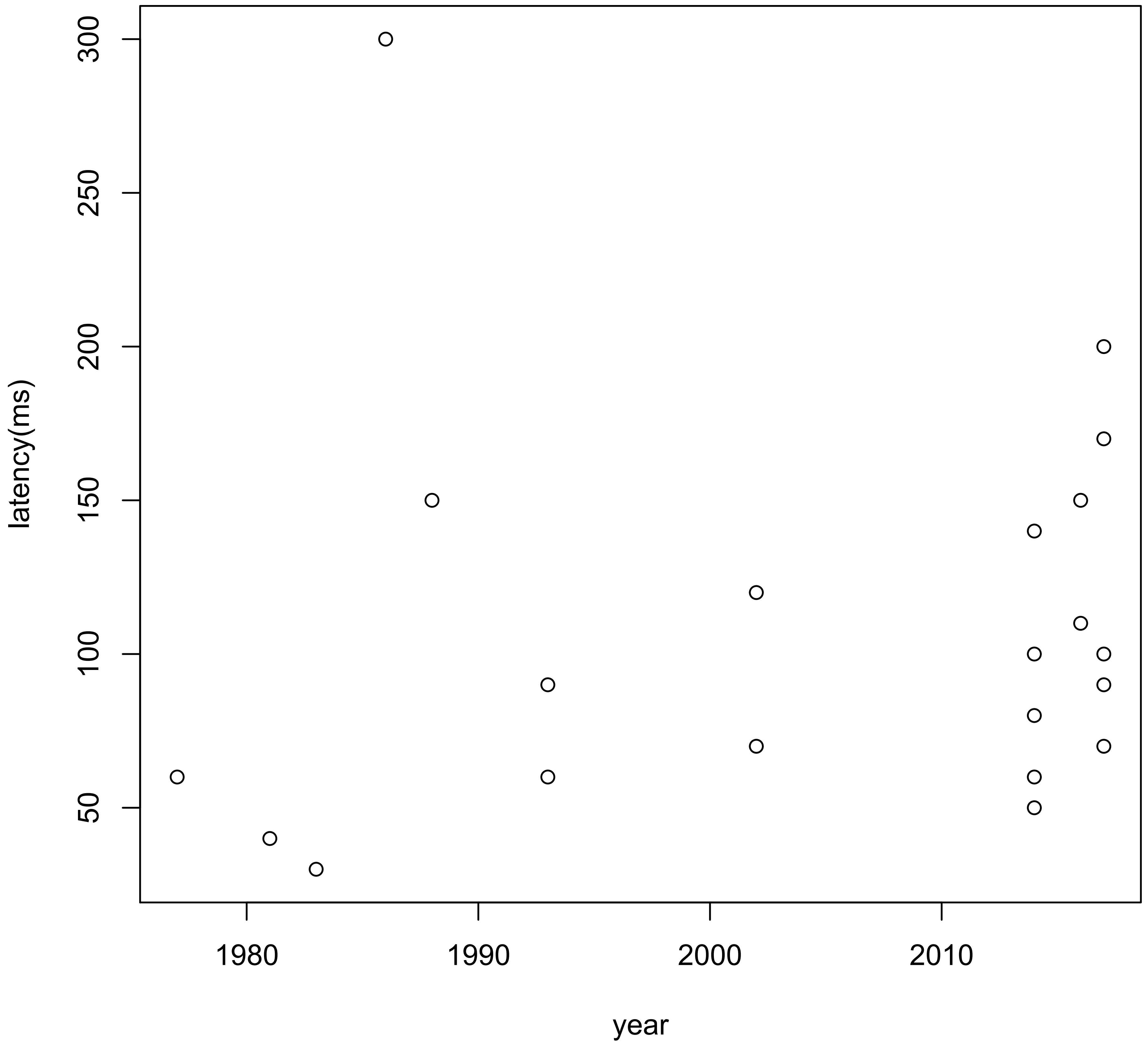
Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

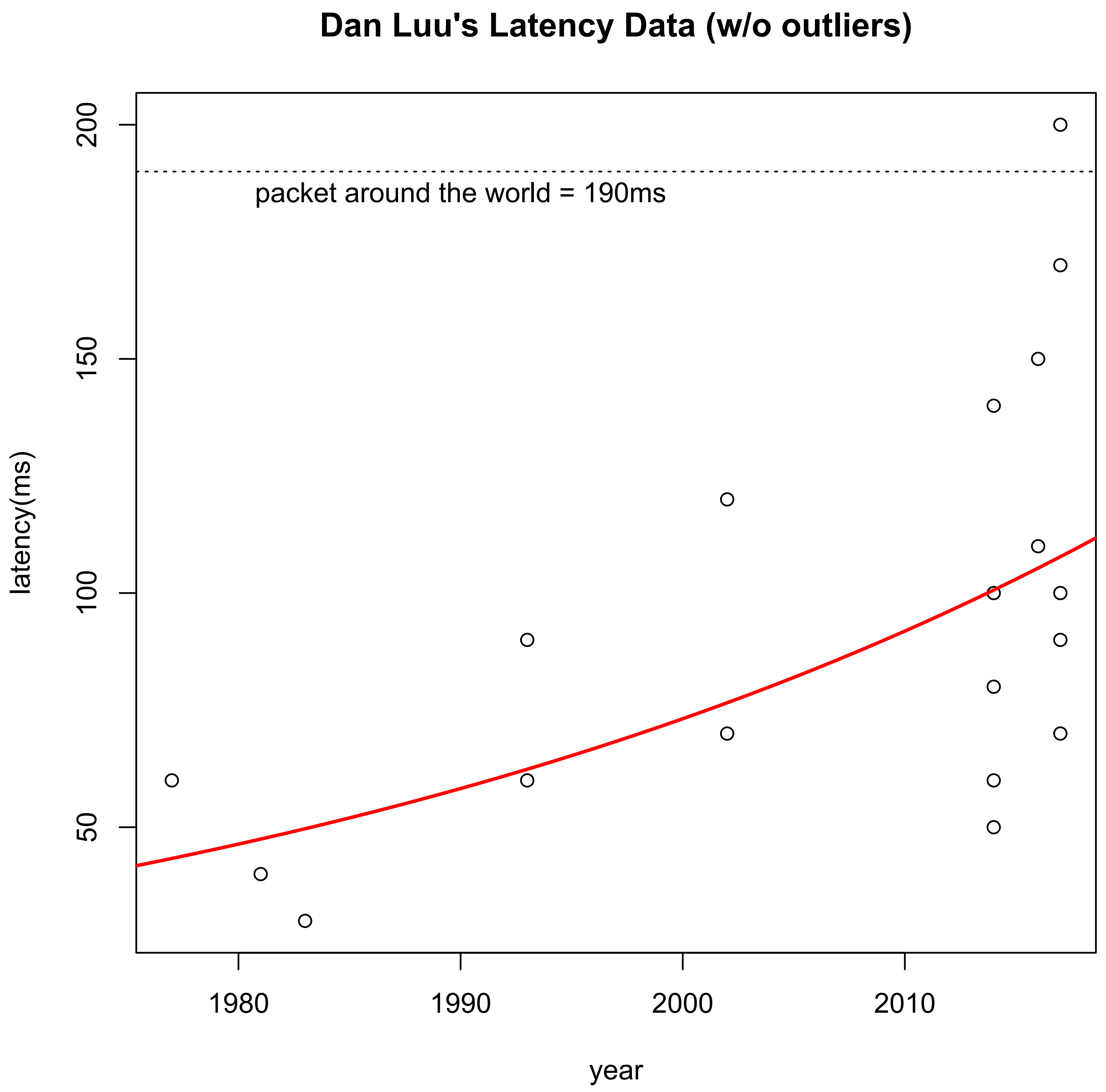


computer	latency (ms)	year	clock	# T
apple 2e	30	1983	1 MHz	3.5k
ti 99/4a	40	1981	3 MHz	8k
custom haswell-e *165Hz*	50	2014	3.5 GHz	2G
commodore pet 4016	60	1977	1 MHz	3.5k
sgi indy	60	1993	.1 GHz	1.2M
custom haswell-e *120Hz*	60	2014	3.5 GHz	2G
thinkpad 13 **chromeos**	70	2017	2.3 GHz	1G
imac g4 **os 9**	70	2002	.8 GHz	11M
custom haswell-e *60Hz*	80	2014	3.5 GHz	2G
mac color classic	90	1993	16 MHz	273k
powerspec g405 **linux** *60Hz*	90	2017	4.2 GHz	2G
macbook pro 2014	100	2014	2.6 GHz	700M
thinkpad 13 **linux chroot**	100	2017	2.3 GHz	1G
lenovo x1 carbon 4g **linux**	110	2016	2.6 GHz	1G
imac g4 **os x**	120	2002	.8 GHz	11M
custom haswell-e *24Hz*	140	2014	3.5 GHz	2G
lenovo x1 carbon 4g **win**	150	2016	2.6 GHz	1G
next cube	150	1988	25 MHz	1.2M
powerspec g405 **linux**	170	2017	4.2 GHz	2G
packet around the world	190			
powerspec g405 **win**	200	2017	4.2 GHz	2G
symbolics 3620	300	1986	5 MHz	390k

https://danluu.com/input-lag/

Dan Luu's Latency Data





Computers are exponentially getting slower (what?)

If we had to pick one root cause of latency bloat, we might say that it’s because of **“complexity”**. Of course, we all know that complexity is bad. If you’ve been to a non-academic non-enterprise tech conference in the past decade, there’s a good chance that there was at least one talk on how complexity is the root of all evil and we should aspire to reduce complexity.

Unfortunately, it's a lot harder to remove complexity than to give a talk saying that we should remove complexity. **A lot of the complexity buys us something, either directly or indirectly.** When we looked at the input of a fancy modern keyboard vs. the apple 2 keyboard, we saw that using a relatively powerful and expensive general purpose processor to handle keyboard inputs can be slower than dedicated logic for the keyboard, which would both be simpler and cheaper. However, using the processor gives people the ability to easily customize the keyboard, and also pushes the problem of “programming” the keyboard from hardware into software, which reduces the cost of making the keyboard. The more expensive chip increases the manufacturing cost, but considering how much of the cost of these small-batch artisanal keyboards is the design cost, it seems like a net win to trade manufacturing cost for ease of programming.

A lot of the complexity might be called accidental complexity, but most of that accidental complexity is there because it's so convenient. At every level from the hardware architecture to the syscall interface to the I/O framework we use, we take on complexity, **much of which could be eliminated if we could sit down and re-write all of the systems and their interfaces today, but it's too inconvenient to re-invent the universe to reduce complexity and we get benefits from economies of scale, so we live with what we have.**

For those reasons and more, in practice, **the solution to poor performance caused by "excess" complexity is often to add more complexity.** In particular, the gains we've seen that get us back to the quickness of the quickest machines from thirty to forty years ago have come not from listening to exhortations to reduce complexity, but from piling on more complexity.

Power Consumption

- To a software engineer, energy consumption is a lot like speed / latency
    - We normally do not really think about it until it becomes a problem
    - We are not trained properly to think about it
    - We expect better hardware to make the problem go away (and hardware does get better… but until when?)

What do programmers know about SW energy consumption?

- A survey of 122 developers
- Only 22 (18%) considers energy consumption when developing software
- Only 12 (10%) have measured power consumption
- Respondents agreed more about cause of energy consumption in mobile devices
  - But none identified advertisement as a power sink
  - Only 19 (16%) identified network as a cause

C. Pang, A. Hindle, B. Adams, and A. E. Hassan. What do programmers know about software energy consumption? IEEE Software, 33(3):83–89, May 2016.

How do you measure power consumption?

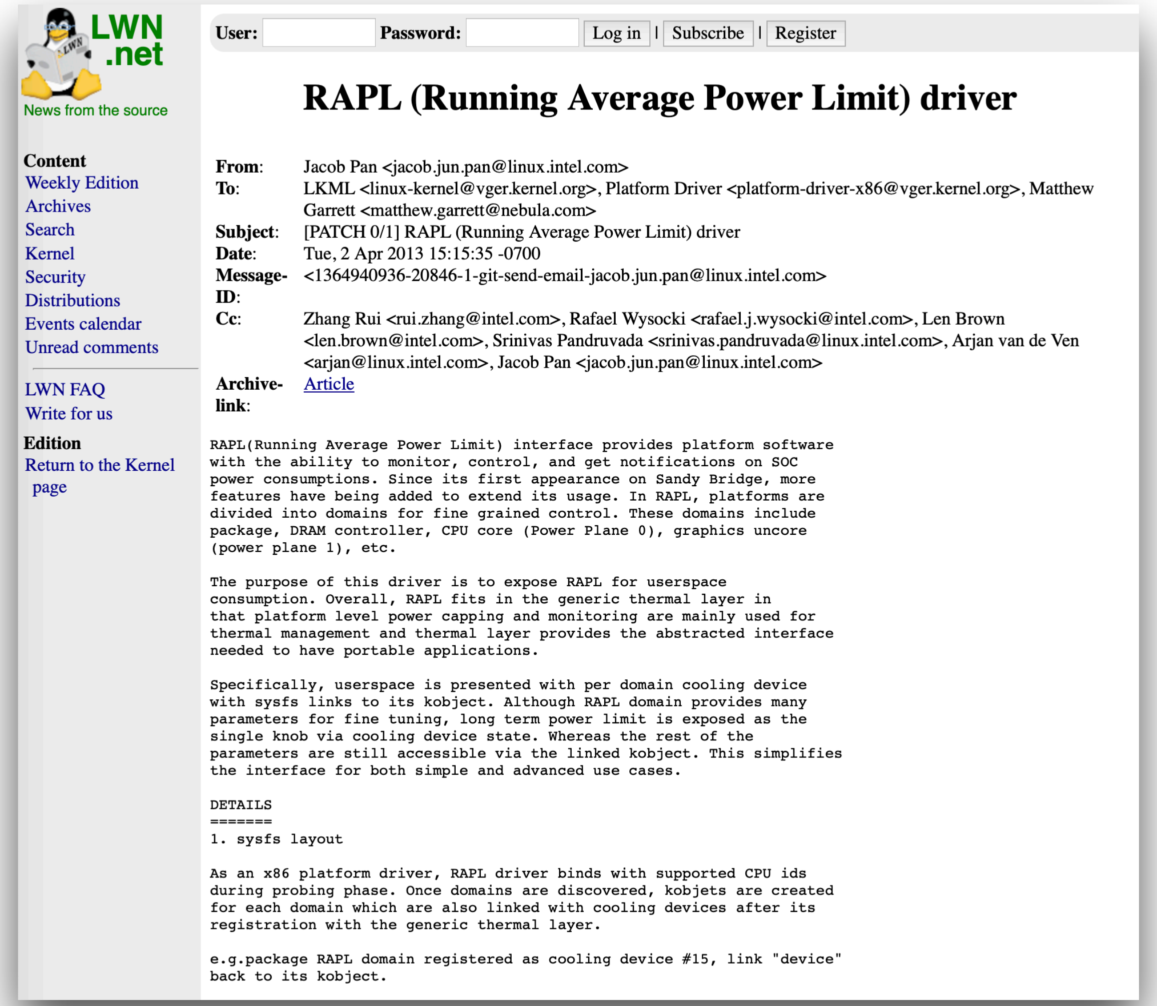
X: too noisy, hard to control programmatically, etc etc

In fact, measuring power consumption remained difficult for long time…

Intel RAPL (Running Average Power Limit)

- Starting from Sandy Bridge, Intel has implemented RAPL, a hardware-based SOC power consumption monitoring that can be read from a special register.

- Linux `perf` supports it.

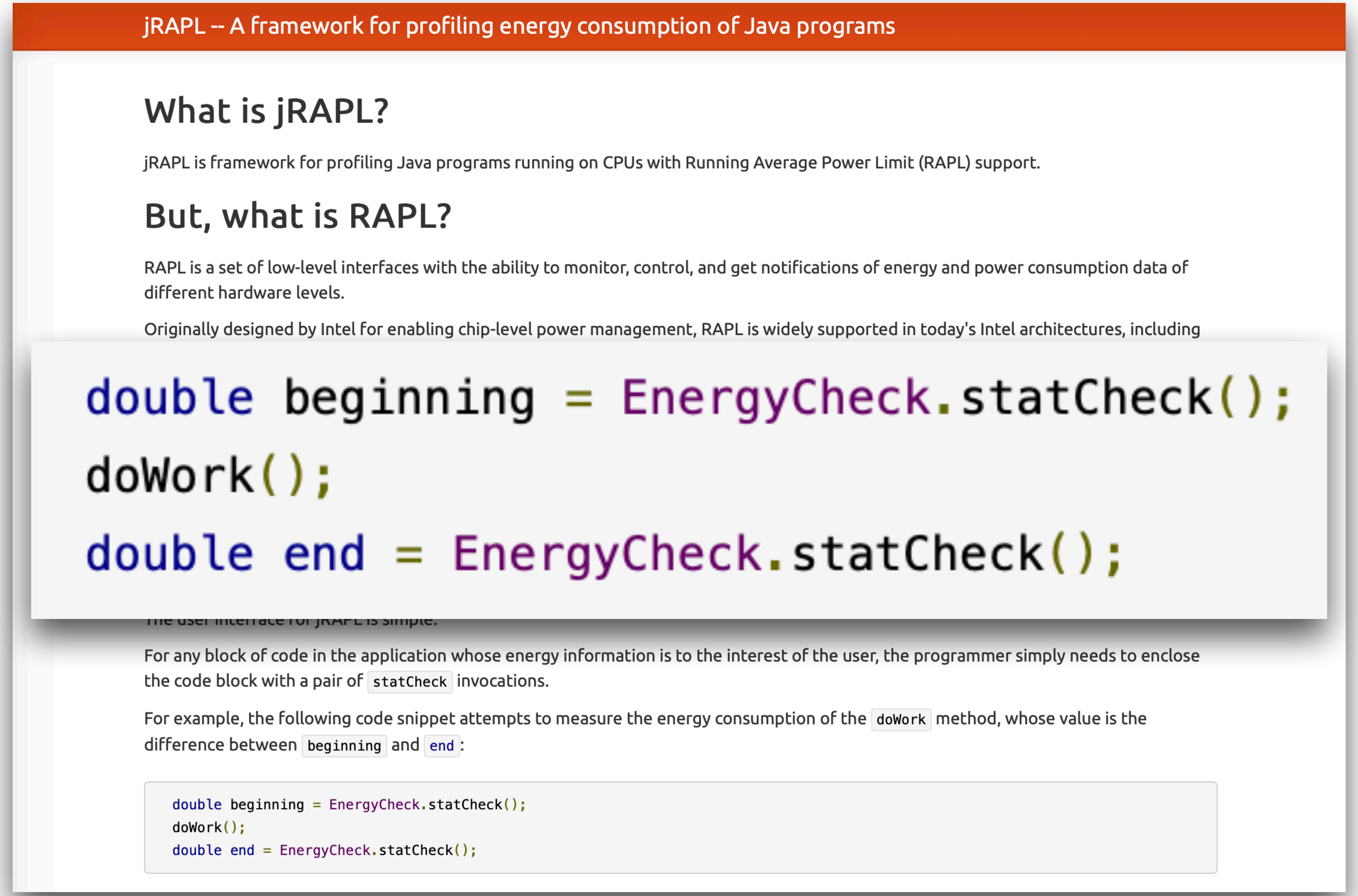


RAPL (Running Average Power Limit) driver

From: Jacob Pan <jacob.jun.pan@linux.intel.com>
To: LKML <linux-kernel@vger.kernel.org>, Platform Driver <platform-driver-x86@vger.kernel.org>, Matthew Garrett <matthew.garrett@nebula.com>
Subject: [PATCH 0/1] RAPL (Running Average Power Limit) driver
Date: Tue, 2 Apr 2013 15:15:35 -0700
Message-ID: <1364940936-20846-1-git-send-email-jacob.jun.pan@linux.intel.com>
Cc: Zhang Rui <rui.zhang@intel.com>, Rafael Wysocki <rafael.j.wysocki@intel.com>, Len Brown <len.brown@intel.com>, Srinivas Pandruvada <srinivas.pandruvada@linux.intel.com>, Arjan van de Ven <arjan@linux.intel.com>, Jacob Pan <jacob.jun.pan@linux.intel.com>
Archive-link: Article

```
RAPL(Running Average Power Limit) interface provides platform software
with the ability to monitor, control, and get notifications on SOC
power consumptions. Since its first appearance on Sandy Bridge, more
features have being added to extend its usage. In RAPL, platforms are
divided into domains for fine grained control. These domains include
package, DRAM controller, CPU core (Power Plane 0), graphics uncore
(power plane 1), etc.

The purpose of this driver is to expose RAPL for userspace
consumption. Overall, RAPL fits in the generic thermal layer in
that platform level power capping and monitoring are mainly used for
thermal management and thermal layer provides the abstracted interface
needed to have portable applications.

Specifically, userspace is presented with per domain cooling device
with sysfs links to its kobject. Although RAPL domain provides many
parameters for fine tuning, long term power limit is exposed as the
single knob via cooling device state. Whereas the rest of the
parameters are still accessible via the linked kobject. This simplifies
the interface for both simple and advanced use cases.

DETAILS
=======
1. sysfs layout

As an x86 platform driver, RAPL driver binds with supported CPU ids
during probing phase. Once domains are discovered, kobjets are created
for each domain which are also linked with cooling devices after its
registration with the generic thermal layer.

e.g.package RAPL domain registered as cooling device #15, link "device"
back to its kobject.
```

jRAPL



jRAPL -- A framework for profiling energy consumption of Java programs

What is jRAPL?

jRAPL is framework for profiling Java programs running on CPUs with Running Average Power Limit (RAPL) support.

But, what is RAPL?

RAPL is a set of low-level interfaces with the ability to monitor, control, and get notifications of energy and power consumption data of different hardware levels.

Originally designed by Intel for enabling chip-level power management, RAPL is widely supported in today's Intel architectures, including

```
double beginning = EnergyCheck.statCheck();
doWork();
double end = EnergyCheck.statCheck();
```

The user interface for jRAPL is simple.

For any block of code in the application whose energy information is to the interest of the user, the programmer simply needs to enclose the code block with a pair of `statCheck` invocations.

For example, the following code snippet attempts to measure the energy consumption of the `doWork` method, whose value is the difference between `beginning` and `end`:

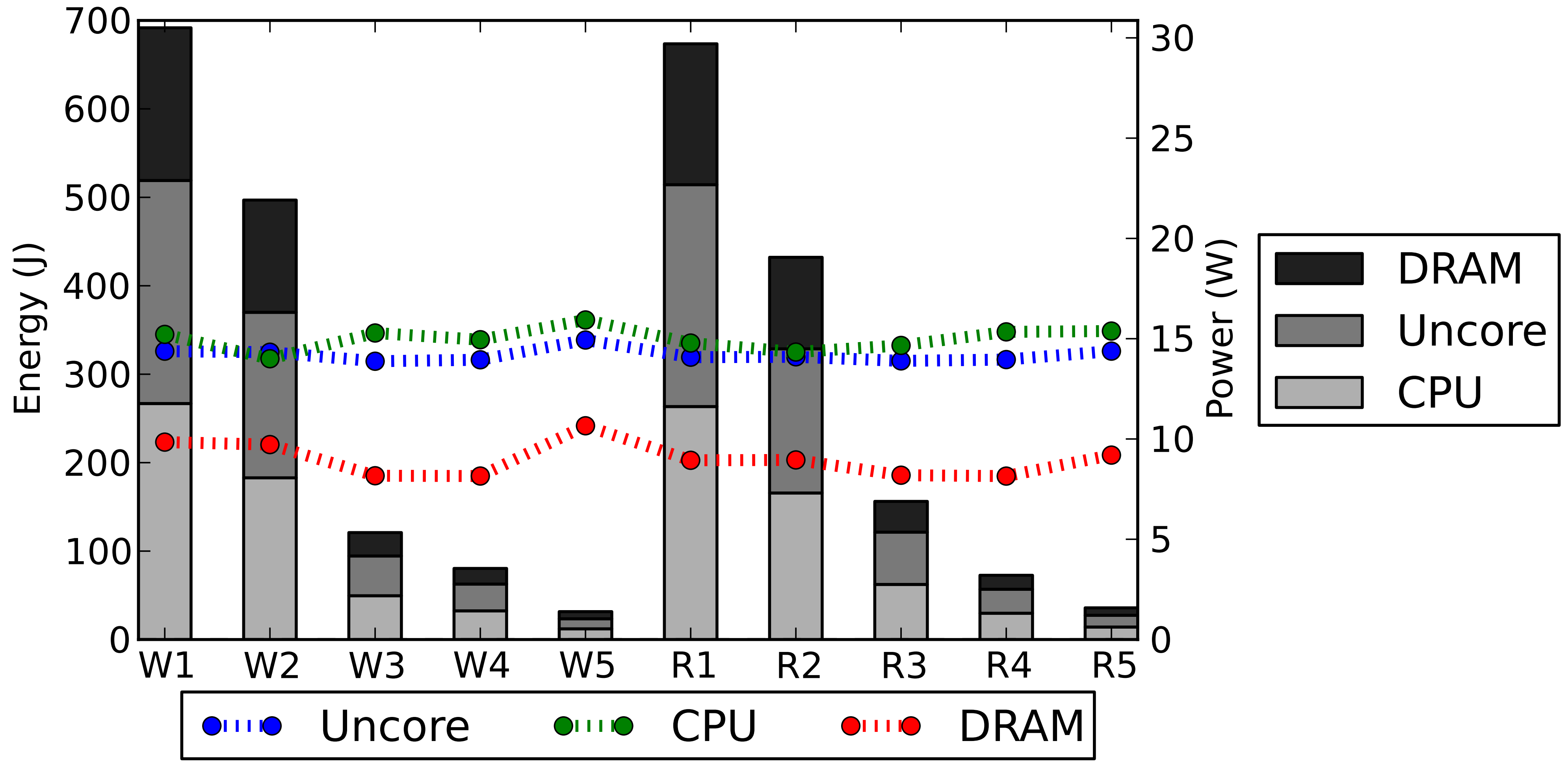
```
double beginning = EnergyCheck.statCheck();
doWork();
double end = EnergyCheck.statCheck();
```

http://kliu20.github.io/jRAPL/

- *data access pattern*: For a large amount of data, does the pattern of access (sequential vs. random, read vs. write) impact energy consumption?
- *data organization and representation*: For different representations of the same data (unboxed vs. boxed data, primitive arrays vs. array lists) have impact on energy consumption?
- *data precision*: Do precision levels (short, integer, floating points, double, long) of data have significant impact on energy consumption?
- *data I/O strategies*: For I/O-intensive applications, do different choices of buffering and different levels of data intensity have impact on energy consumption?

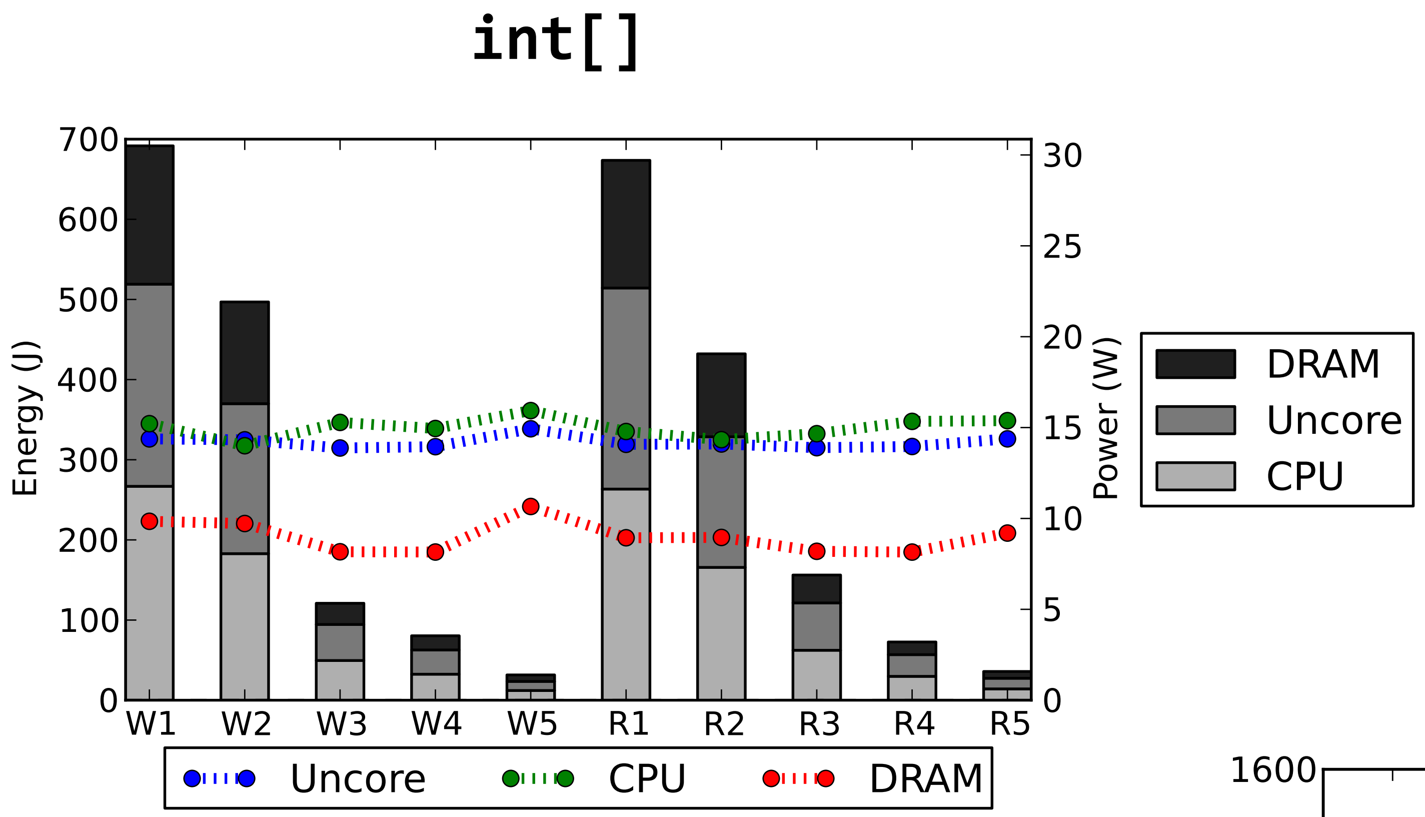
K. Liu, G. Pinto, and Y. D. Liu. Data-oriented characterization of application-level energy optimization. In A. Egyed and I. Schaefer, editors, Fundamental Approaches to Software Engineering, pages 316–331, Berlin, Heidelberg, 2015.

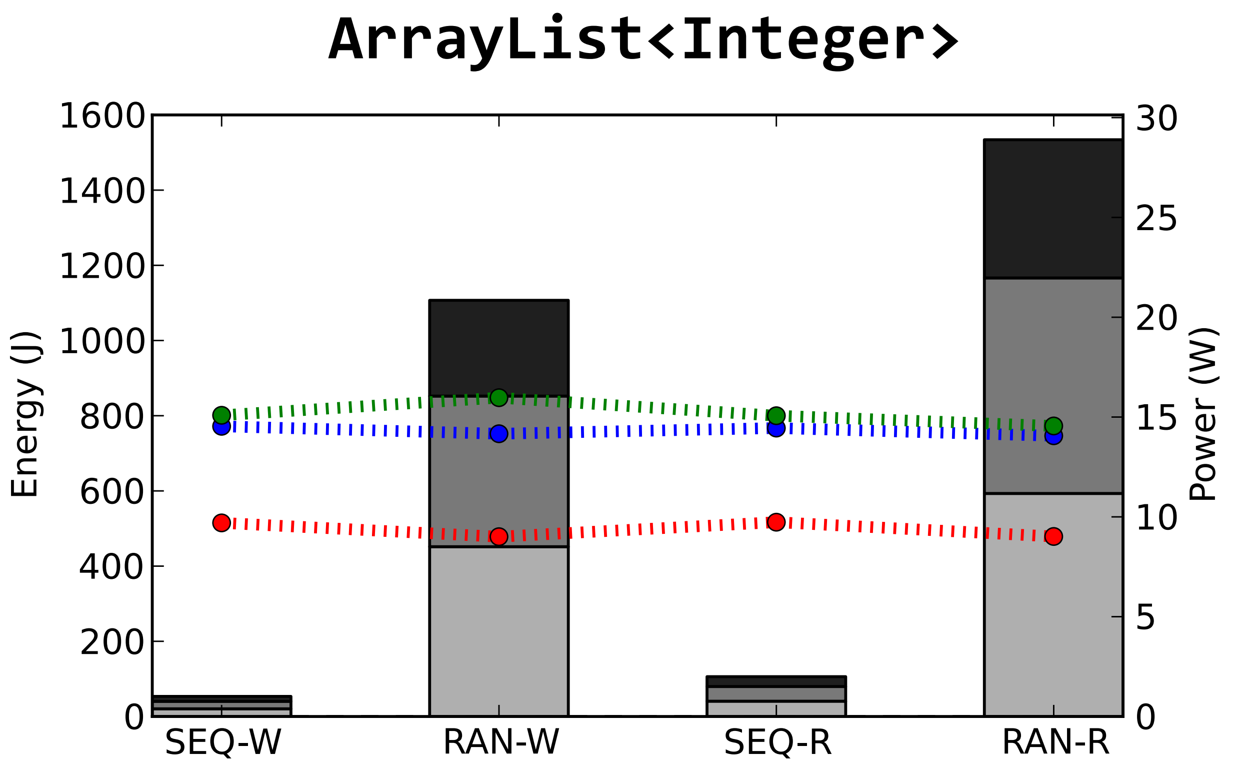
Data Access Pattern



Reading/writing 50,000,000 integers from an array.
Cache locality really works.
Counterintuitively, writing is not more expensive.

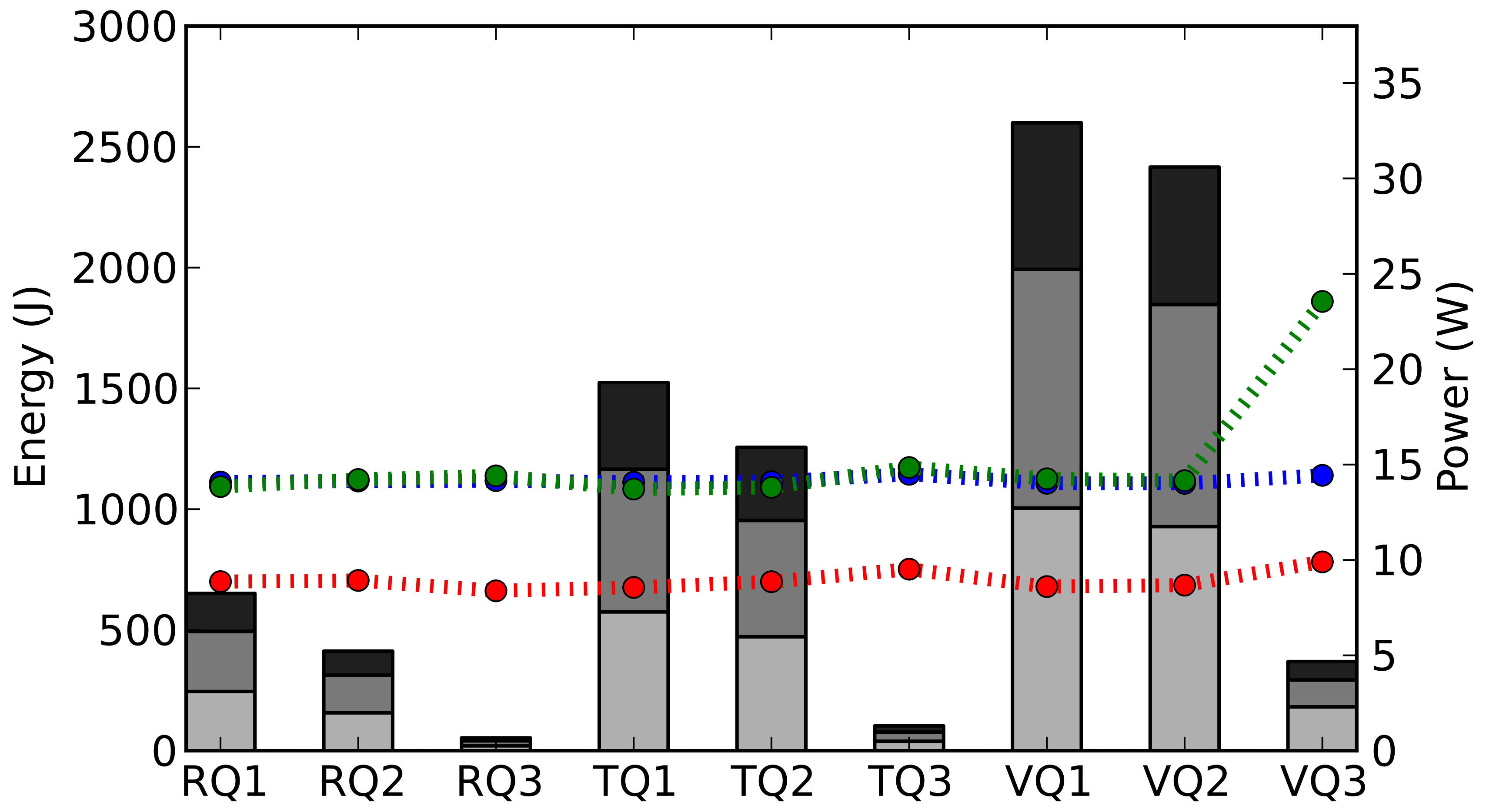
Data Representation





Abstract data types are significantly more expensive.

Types of Query



Reference Query (RQ) accesses stack, whereas the Value Query (VQ) accesses the heap (much slower).

Type Query (TQ) access object metadata on the heap, but instances of the same type shares the same metadata, so caching kicks in.

1: 100% random access, 2: 25% random access, 3: sequential access

Data Organisation

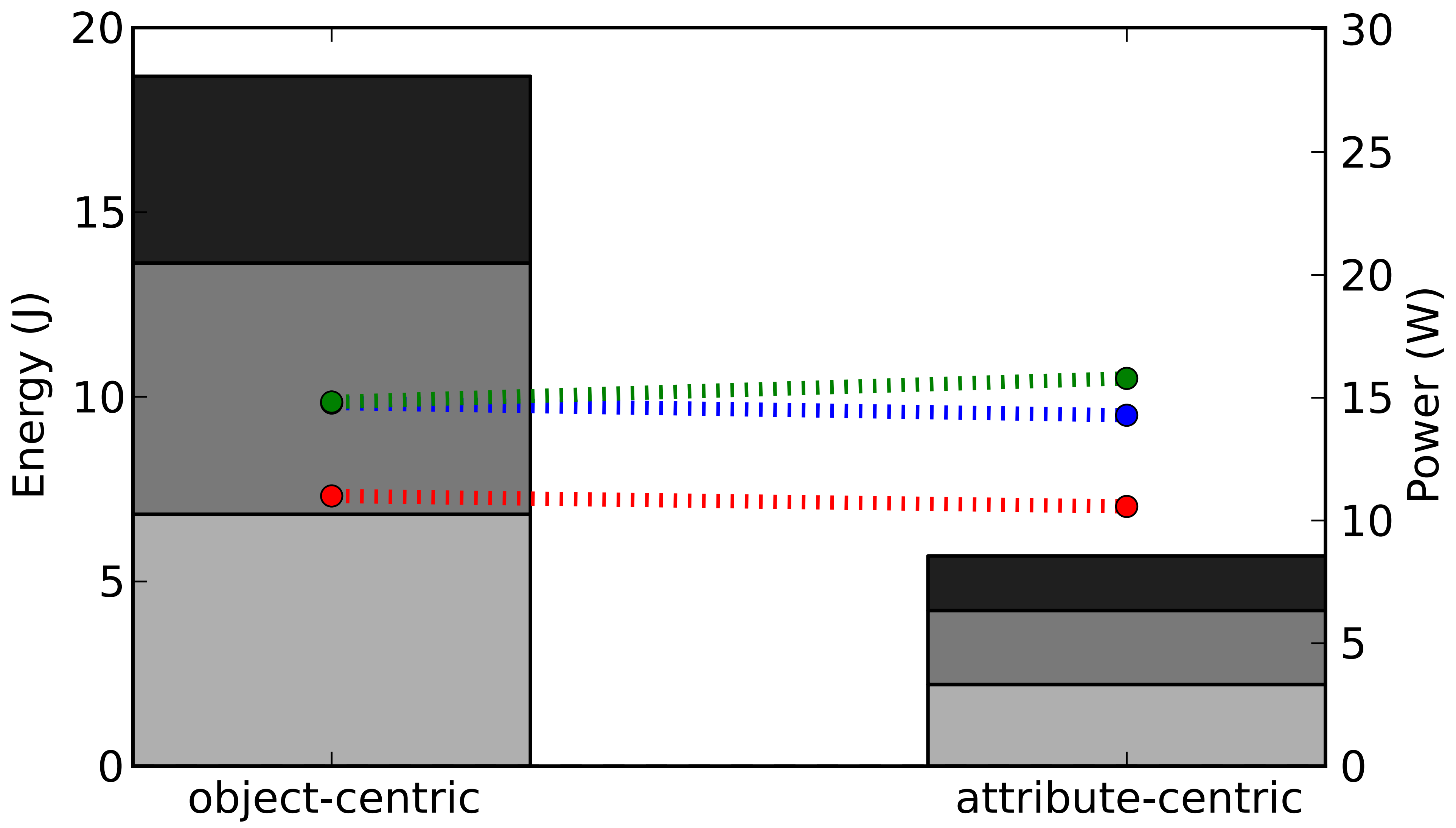
```
class Grouped {
  int a, b, c, d, e = ...;
}
class Main {
  Grouped[] group = ...;
  void calc() {
    for (int i = 0; i < N; i++) {
      group[i].e = group[i].a * group[i].b * group[i].c * group[i].d;
}}}
```

Fig. 1. Object-Centric Data Grouping

```
class Main {
  int[] a = ..; int[] b = ..; int[] c = ..; int[] d = ..; int[] e = ..;
  void calc() {
    for (int i = 0; i < N; i++) {
      e[i] = a[i] * b[i] * c[i] * d[i];
}}}
```

Fig. 2. Attribute-Centric Data Grouping

Data Organisation



Recall Dan Luu: what are we buying with the extra power consumption here?

Convenience in programming: the high level abstraction of OO

Java I/O

TABLE I: Characteristics of the studied Java I/O APIs.

Class	Acronym	Method instrumented	Extends from	Available from	# in OSS projects
BufferedWriter	BW	`void write(String str)`	java.io.Writer	JDK 1.1	4,705
FileWriter	FW	`void write(String str)`	java.io.Writer	JDK 1.1	3,353
StringWriter	SW	`void write(String str)`	java.io.Writer	JDK 1.1	2,026
PrintWriter	PW	`void write(String str)`	java.io.Writer	JDK 1.1	10,501
CharArrayWriter	CAW	`void write(String str)`	java.io.Writer	JDK 1.1	572
BufferedReader	BR	`int read()`	java.io.Reader	JDK 1.1	12,441
LineNumberReader	LNR	`int read()`	java.io.Reader	JDK 1.1	897
CharArrayReader	CAR	`int read()`	java.io.Reader	JDK 1.1	187
PushbackReader	PBR	`int read()`	java.io.Reader	JDK 1.1	779
FileReader	FR	`int read()`	java.io.Reader	JDK 1.1	1,695
StringReader	SR	`int read()`	java.io.Reader	JDK 1.1	536
FileOutputStream	FOS	`void write(byte[] b)`	java.io.OutputStream	JDK 1.0	3,541
ByteArrayOutputStream	BAOS	`void write(byte[] b)`	java.io.OutputStream	JDK 1.0	6,946
BufferedOutputStream	BOS	`void write(byte[] b)`	java.io.OutputStream	JDK 1.0	1,753
PrintStream	PST	`void print(String str)`	java.io.OutputStream	JDK 1.0	8,424
FileInputStream	FIS	`int read()`	java.io.InputStream	JDK 1.0	2,823
BufferedInputStream	BIS	`int read()`	java.io.InputStream	JDK 1.0	1,832
PushbackInputStream	PBIS	`int read()`	java.io.InputStream	JDK 1.0	688
ByteArrayInputStream	BAIS	`int read()`	java.io.InputStream	JDK 1.0	1,532

G. Rocha, F. Castor, and G. Pinto. Comprehending energy behaviors of java i/o apis. In 2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), pages 1–12, Sep. 2019.

Java I/O

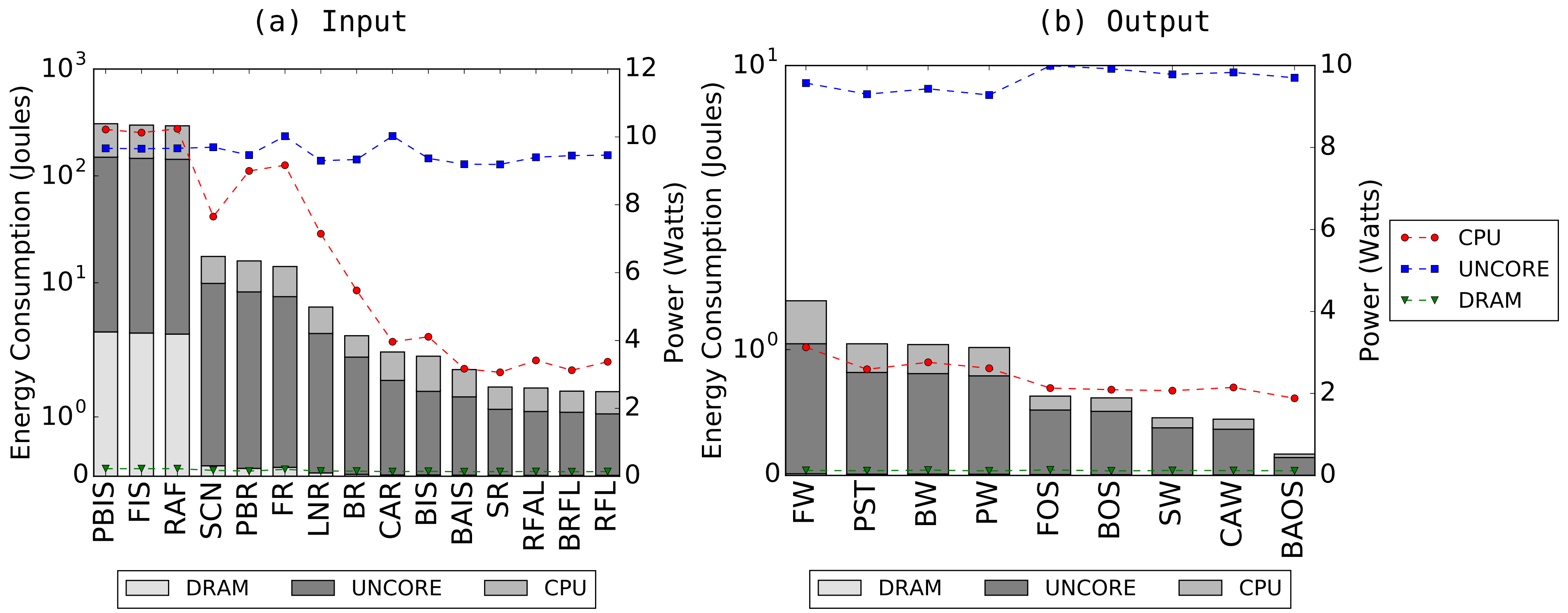
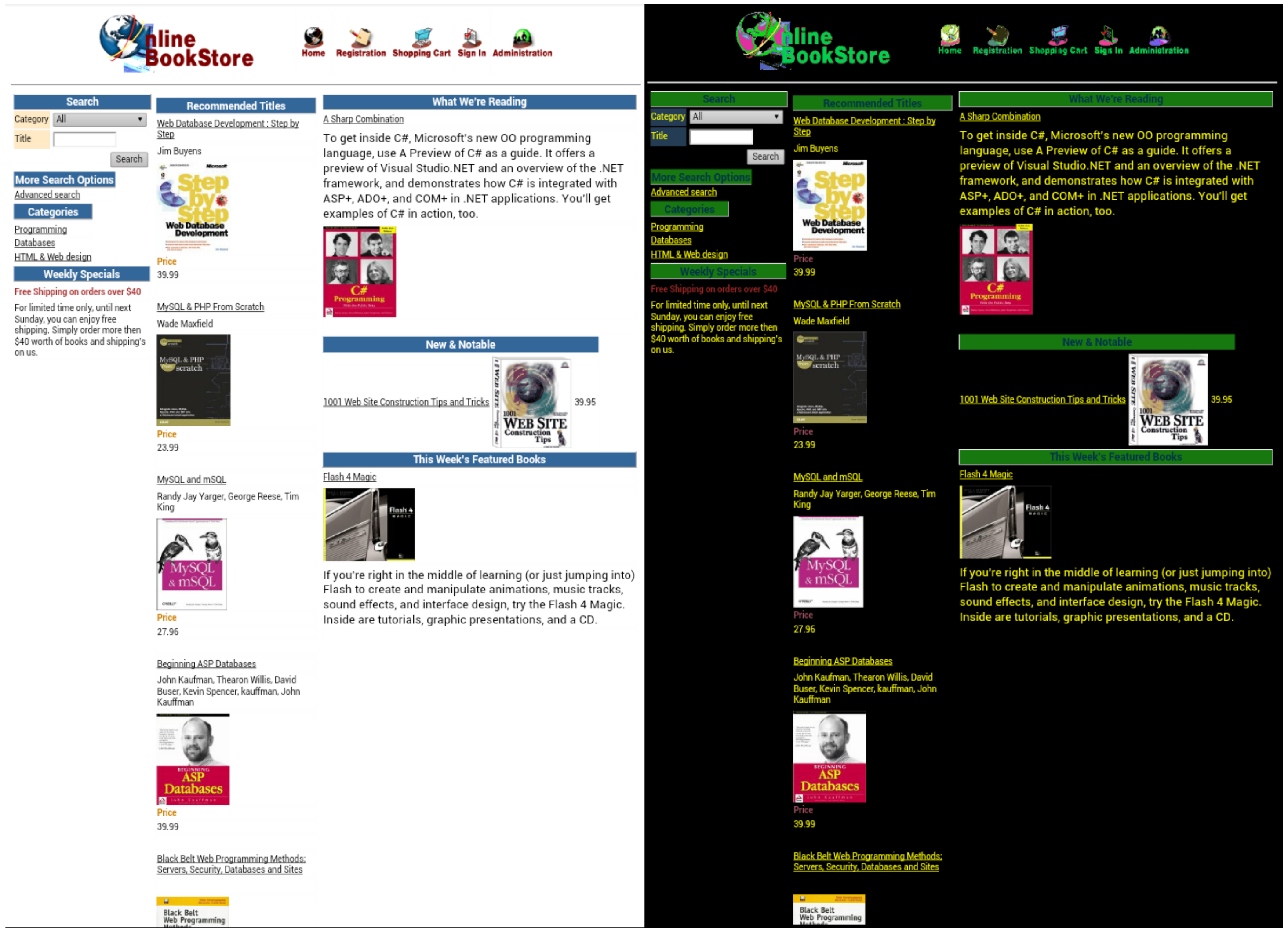


Fig. 1: Energy consumption behavior of Java I/O APIs. Energy data is presented in a logarithmic scale. For the figure on the left, PBIS stands for `PushbackInputStream`, FIS stands for `FileInputStream`, RAF stands for `RadomAccessFile`, SCN stands for `Scanner`, PBR stands for `PushbackReader`, FR stands for `FileReader`, LNR stands for `LineNumberReader`, BR stands for `BufferedReader`, CAR stands for `CharArrayReader`, BIS stands for `BufferedInputStream`, BAIS stands for `ByteArrayInputStream`, SR stands for `StringReader`, RFAL stands for `Files.readAllLines`, BRFL stands for `Files.newBufferedReader`, and RFL stands for `Files.lines`. For the figure on the right, FW stands for `FileWriter`, PST stands for `PrintStream`, BW stands for `BufferedWriter`, PW stands for `PrintWriter`, FOS stands for `FileOutputStream`, BOS stands for `BufferedOutputStream`, SW stands for `StringWriter`, CAW stands for `CharArrayWriter`, BAOS stands for `ByteArrayOutputStream`.

Display Energy Optimisation for OLED Displays



Free Shipping on orders over $40

For limited time only, until next Sunday, you can enjoy free shipping. Simply order more then $40 worth of books and shipping's on us.

Web Database Development : Step by Step
Jim Buyens
Price 39.99

MySQL & PHP From Scratch
Wade Maxfield
Price 23.99

MySQL and mSQL
Randy Jay Yarger, George Reese, Tim King
Price 27.96

Beginning ASP Databases
John Kaufman, Thearon Willis, David Buser, Kevin Spencer, kauffman, John Kauffman
Price 39.99

Black Belt Web Programming Methods: Servers, Security, Databases and Sites

A Sharp Combination

To get inside C#, Microsoft's new OO programming language, use A Preview of C# as a guide. It offers a preview of Visual Studio.NET and an overview of the .NET framework, and demonstrates how C# is integrated with ASP+, ADO+, and COM+ in .NET applications. You'll get examples of C# in action, too.

1001 Web Site Construction Tips and Tricks 39.95

Flash 4 Magic

If you're right in the middle of learning (or just jumping into) Flash to create and manipulate animations, music tracks, sound effects, and interface design, try the Flash 4 Magic. Inside are tutorials, graphic presentations, and a CD.

Figure 5: Comparison in questionnaires

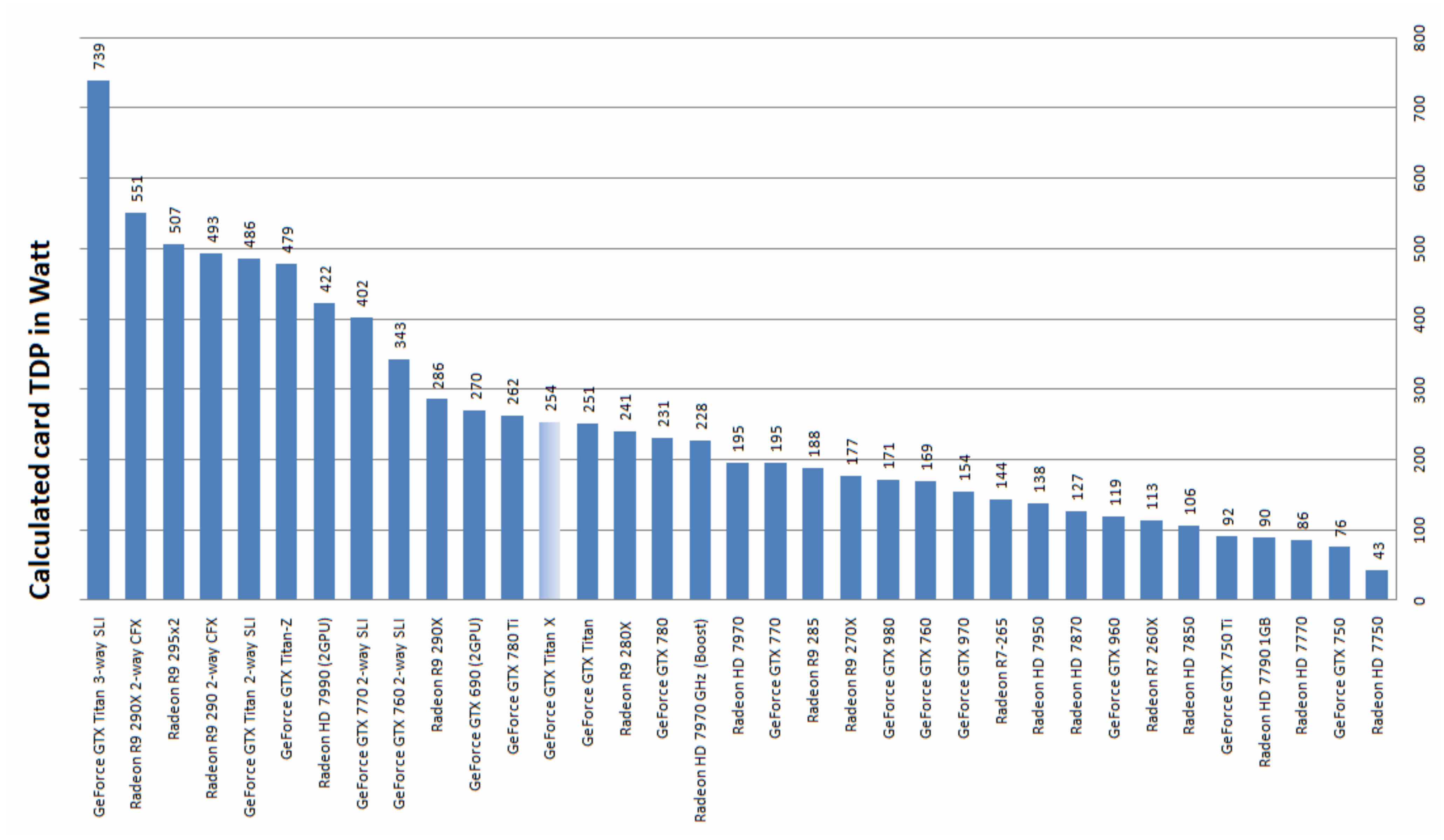
OLED displays consume more power when displaying brighter colours: this research automatically changes web application's colour scheme for low energy consumption.

It can reduce display energy consumption by up to 40%

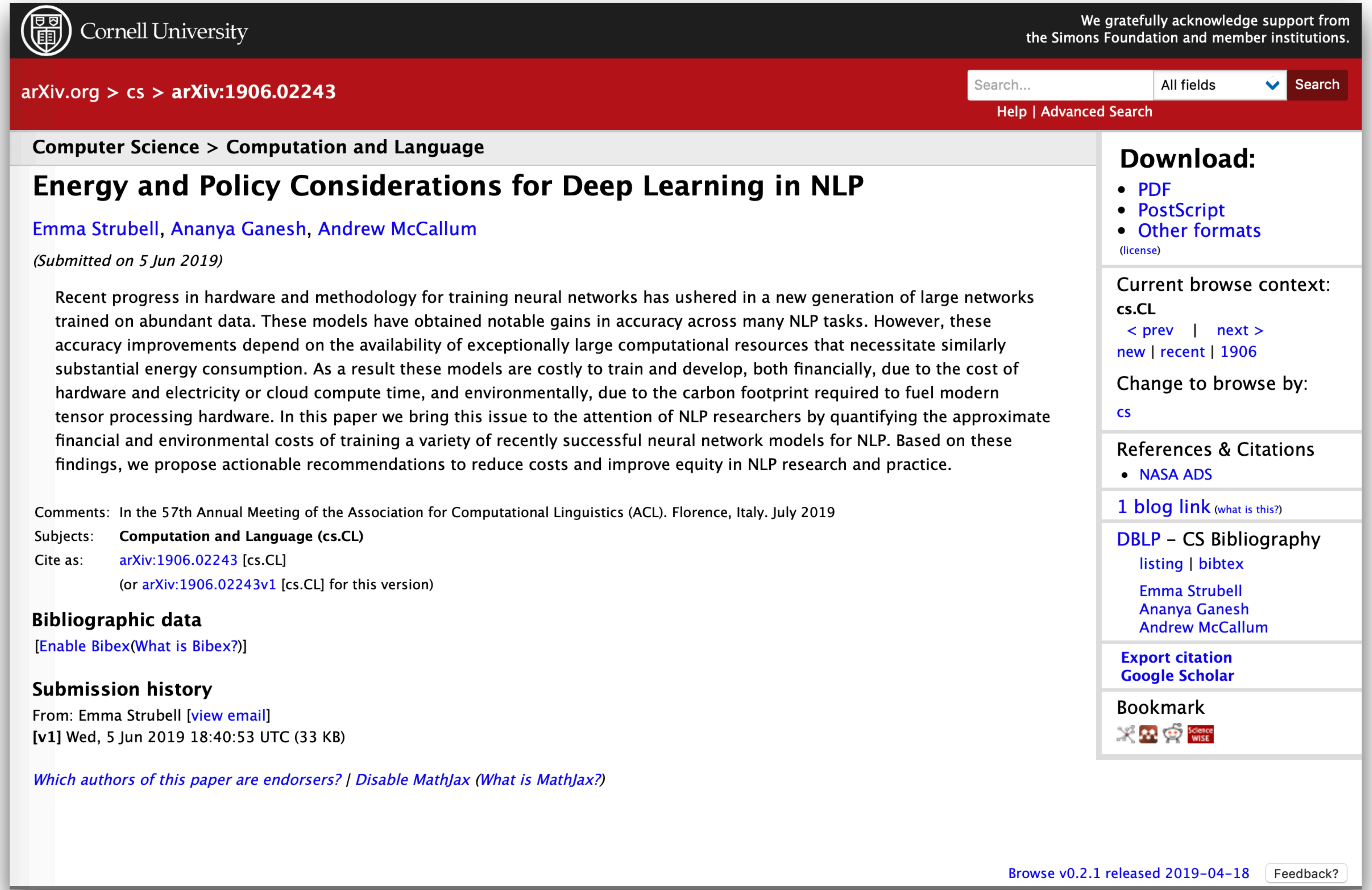
D. Li, A. H. Tran, and W. G. J. Halfond. Making web applications more energy efficient for oled smartphones. In Proceedings of the 36th International Conference on Software Engineering, ICSE 2014, pages 527–538, 2014. ACM.

There are areas of SW technology that are increasingly burning more fuel. Guess what they are…?

Deep Learning



GPU is a power hog.


Cornell University

We gratefully acknowledge support from the Simons Foundation and member institutions.

arXiv.org > cs > arXiv:1906.02243

Computer Science > Computation and Language

Energy and Policy Considerations for Deep Learning in NLP

Emma Strubell, Ananya Ganesh, Andrew McCallum

(Submitted on 5 Jun 2019)

Recent progress in hardware and methodology for training neural networks has ushered in a new generation of large networks trained on abundant data. These models have obtained notable gains in accuracy across many NLP tasks. However, these accuracy improvements depend on the availability of exceptionally large computational resources that necessitate similarly substantial energy consumption. As a result these models are costly to train and develop, both financially, due to the cost of hardware and electricity or cloud compute time, and environmentally, due to the carbon footprint required to fuel modern tensor processing hardware. In this paper we bring this issue to the attention of NLP researchers by quantifying the approximate financial and environmental costs of training a variety of recently successful neural network models for NLP. Based on these findings, we propose actionable recommendations to reduce costs and improve equity in NLP research and practice.

Comments:	In the 57th Annual Meeting of the Association for Computational Linguistics (ACL). Florence, Italy. July 2019
Subjects:	**Computation and Language (cs.CL)**
Cite as:	arXiv:1906.02243 [cs.CL]
	(or arXiv:1906.02243v1 [cs.CL] for this version)

Bibliographic data

[Enable Bibex(What is Bibex?)]

Submission history

From: Emma Strubell [view email]
[v1] Wed, 5 Jun 2019 18:40:53 UTC (33 KB)

Which authors of this paper are endorsers? | *Disable MathJax* (*What is MathJax?*)

Download:
- PDF
- PostScript
- Other formats

(license)

Current browse context:
cs.CL
< prev | next >
new | recent | 1906

Change to browse by:
cs

References & Citations
- NASA ADS

1 blog link (what is this?)

DBLP – CS Bibliography
listing | bibtex
Emma Strubell
Ananya Ganesh
Andrew McCallum

Export citation
Google Scholar

Bookmark

Browse v0.2.1 released 2019-04-18 Feedback?

https://arxiv.org/abs/1906.02243

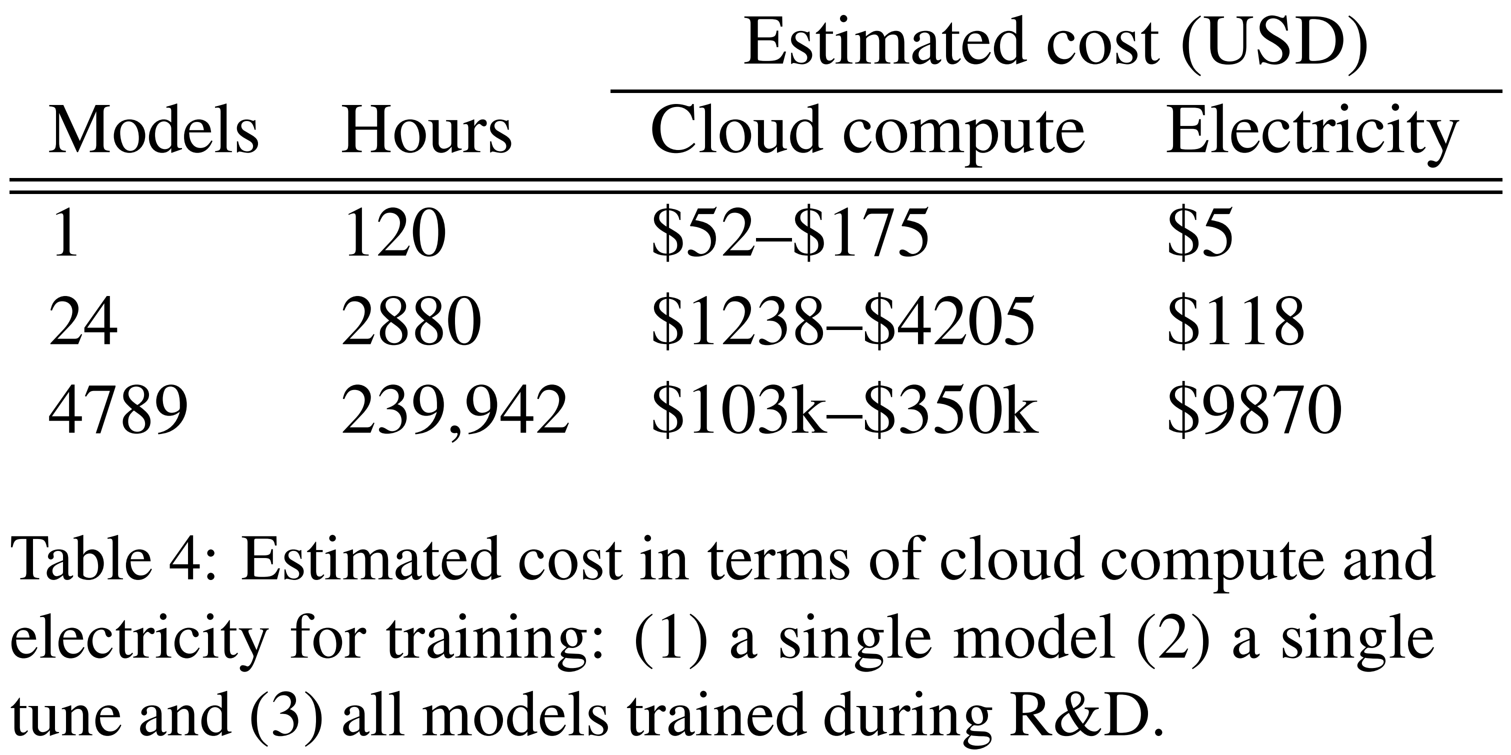
Consumption	**CO_2e (lbs)**
Air travel, 1 passenger, NY↔SF	1984
Human life, avg, 1 year	11,023
American life, avg, 1 year	36,156
Car, avg incl. fuel, 1 lifetime	126,000

Training one model (GPU)	
NLP pipeline (parsing, SRL)	39
w/ tuning & experimentation	78,468
Transformer (big)	192
w/ neural architecture search	626,155

Table 1: Estimated CO_2 emissions from training common NLP models, compared to familiar consumption.[1]

Model	Hardware	Power (W)	Hours	kWh·PUE	CO_2e	Cloud compute cost
$\text{Transformer}_{base}$	P100x8	1415.78	12	27	26	\$41–\$140
Transformer_{big}	P100x8	1515.43	84	201	192	\$289–\$981
ELMo	P100x3	517.66	336	275	262	\$433–\$1472
BERT_{base}	V100x64	12,041.51	79	1507	1438	\$3751–\$12,571
BERT_{base}	TPUv2x16	—	96	—	—	\$2074–\$6912
NAS	P100x8	1515.43	274,120	656,347	626,155	\$942,973–\$3,201,722
NAS	TPUv2x1	—	32,623	—	—	\$44,055–\$146,848
GPT-2	TPUv3x32	—	168	—	—	\$12,902–\$43,008

Table 3: Estimated cost of training a model in terms of CO_2 emissions (lbs) and cloud compute cost (USD).[7] Power and carbon footprint are omitted for TPUs due to lack of public information on power draw for this hardware.



		Estimated cost (USD)	
Models	Hours	Cloud compute	Electricity
1	120	\$52–\$175	\$5
24	2880	\$1238–\$4205	\$118
4789	239,942	\$103k–\$350k	\$9870

Table 4: Estimated cost in terms of cloud compute and electricity for training: (1) a single model (2) a single tune and (3) all models trained during R&D.

Estimated cost of running all experiments for the paper:

E. Strubell, P. Verga, D. Andor, D. Weiss, and A. McCallum. Linguistically-informed self-attention for semantic role labeling. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 5027–5038, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics.

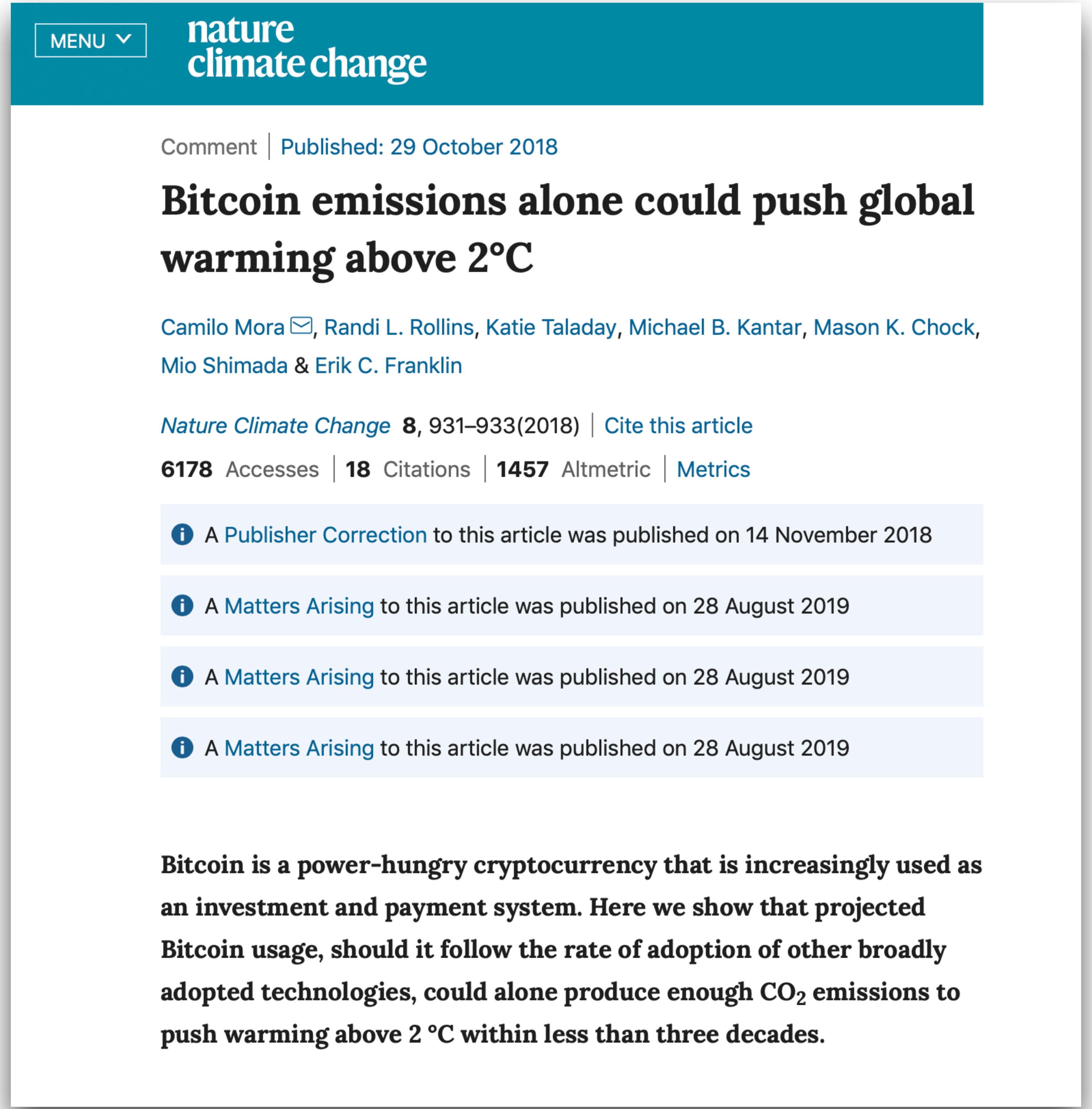
Cryptocurrency 🤯

Cryptocurrency 101

- Transactions on decentralised cryptocurrency are recorded on blockchain
    - Blockchain is simply distributed chain of key-value storage: each “block” contains the cryptographic hash of the previous block
    - You cannot modify a block in the middle without changing the consensus of the entire chain: decentralised transparency!

Proof-of-Work

- Some computation that is reasonably difficult to do, but easy to verify: this is often used to prevent denial-of-service
- Hashcash PoW for spam prevention
  - Sender generates a random number and adds this to the message header
  - Sender computes 160 bit SHA1 of the header until the first 20 bits are all zero: there are 2^{120} such hash values out of 2^{160}
  - Receiver just needs to compute the SHA1 of the header
- All major cryptocurrencies use the Proof-of-Work scheme to verify the generation of new block: when you add a new block, you get a portion of the fees from the transactions recorded in the block. The difficulty of the PoW is controlled so that a new block can be generated at a desired interval.


MENU

nature climate change

Comment | Published: 29 October 2018

Bitcoin emissions alone could push global warming above 2°C

Camilo Mora, Randi L. Rollins, Katie Taladay, Michael B. Kantar, Mason K. Chock, Mio Shimada & Erik C. Franklin

Nature Climate Change **8**, 931–933(2018) | Cite this article

6178 Accesses | **18** Citations | **1457** Altmetric | Metrics

A Publisher Correction to this article was published on 14 November 2018

A Matters Arising to this article was published on 28 August 2019

A Matters Arising to this article was published on 28 August 2019

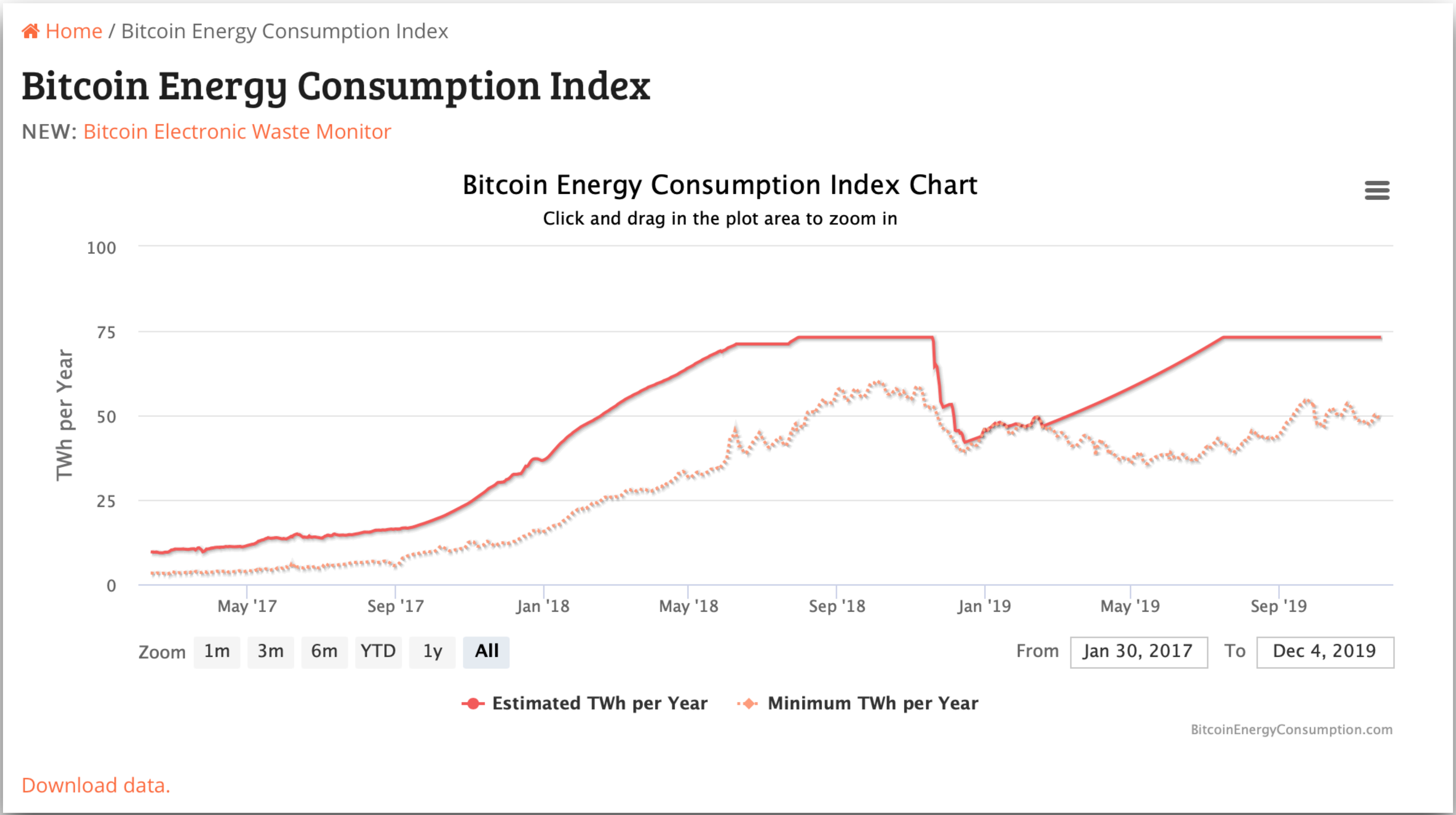
A Matters Arising to this article was published on 28 August 2019

Bitcoin is a power-hungry cryptocurrency that is increasingly used as an investment and payment system. Here we show that projected Bitcoin usage, should it follow the rate of adoption of other broadly adopted technologies, could alone produce enough CO_2 emissions to push warming above 2 °C within less than three decades.

https://www.nature.com/articles/s41558-018-0321-8

From 1860 to 2014, humanity emitted ~584.4 GtC from fossil fuel combustion, industry processes and land-use change, which has been mirrored by ~0.9 °C of global warming (green line in Fig. 1a). Temperature projections from 42 Earth system models (ESMs) developed for the recent Coupled Model Intercomparison Project Phase 5 (CMIP5) under four alternative emission scenarios show that **an additional 231.4 to 744.8 GtC would push global warming across the 2°C threshold.** Reducing emissions to keep warming below 2°C is already regarded as a very difficult challenge given the increasing human population and consumption1 as well as a lack of political will. **Then came Bitcoin.**

Compiling data on the electricity consumption of the various computing systems used for Bitcoin verification at present and the emissions from electricity production in the countries of the companies that performed such computing, **we estimated that in 2017, Bitcoin usage emitted 69 MtC.**



Home / Bitcoin Energy Consumption Index

Bitcoin Energy Consumption Index

NEW: Bitcoin Electronic Waste Monitor

Bitcoin Energy Consumption Index Chart

Click and drag in the plot area to zoom in

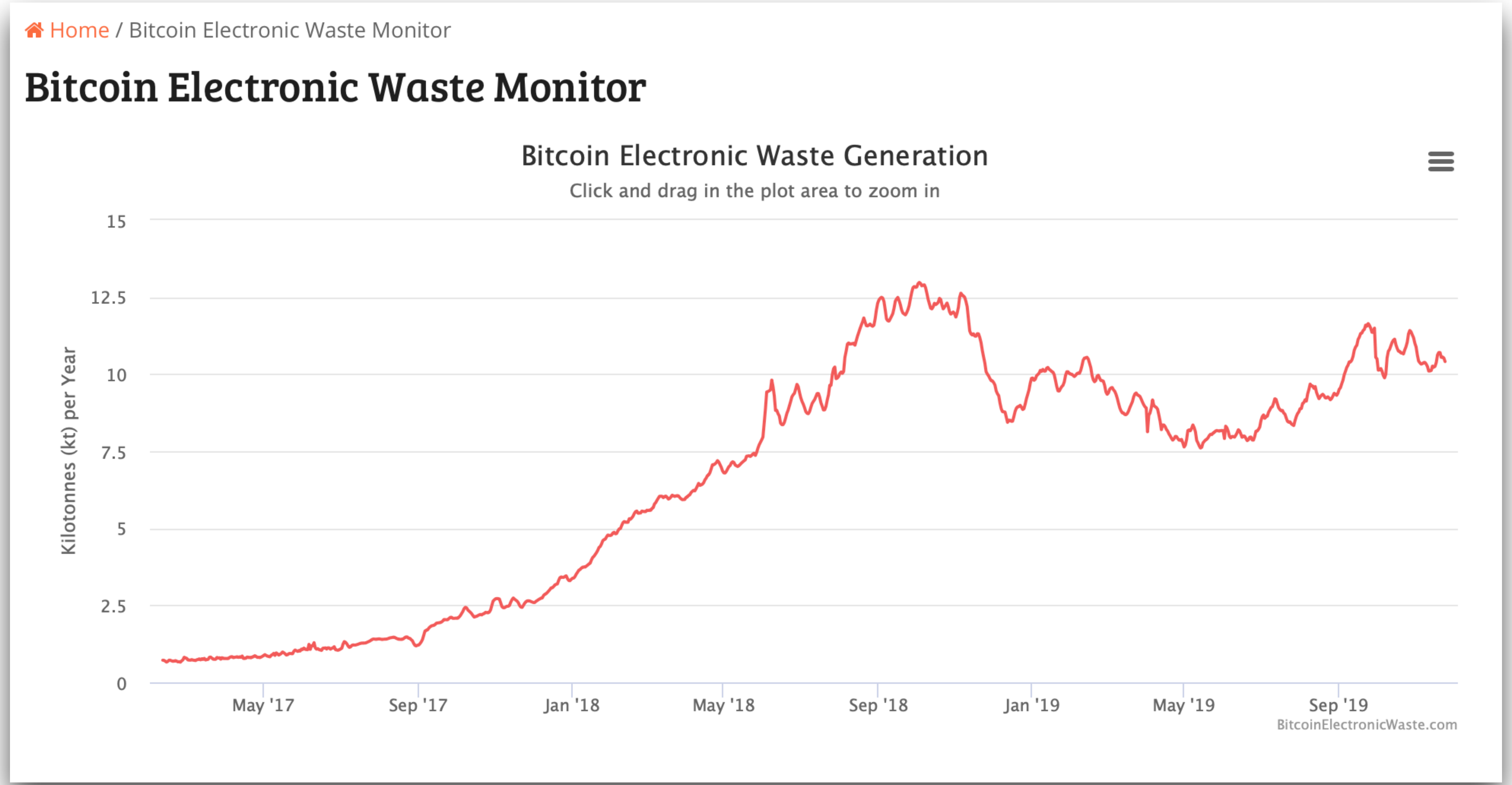
From Jan 30, 2017 To Dec 4, 2019

● Estimated TWh per Year ◆ Minimum TWh per Year

BitcoinEnergyConsumption.com

Download data.

https://digiconomist.net/bitcoin-energy-consumption



Home / Bitcoin Electronic Waste Monitor

Bitcoin Electronic Waste Monitor

Bitcoin Electronic Waste Generation
Click and drag in the plot area to zoom in
Kilotonnes (kt) per Year
15
12.5
10
7.5
5
2.5
0
May '17
Sep '17
Jan '18
May '18
Sep '18
Jan '19
May '19
Sep '19
BitcoinElectronicWaste.com

https://digiconomist.net/bitcoin-electronic-waste-monitor/

Proof-of-Stake

- In PoW, whoever first solves (pays) the hashcash for the next block *wins* the block

- In Proof-of-Stake scheme, the next block is assigned to an order based on various “stake”:

    - Ethereum 2.0 proposal: anyone who wants to become a validator makes a deposit of stake - then one of the validators is randomly chosen to generate the next valid block on the chain

    - This has been discussed for a number of years, but Ethereum is yet to move onto PoS scheme.

Existential Threats?

- To me personally, the environmental concerns are the most painful to think about; they pose the most direct dilemma to us engineers and scientists.
    - Whatever we do is likely to add complexity to the system.
    - Yet the complexity itself may be the source of all this problems.
- Should we just disappear? Is giving up the only logical conclusion? Or should we still believe in technology?

Concluding Thoughts

- Do you believe in “appropriate technology”? Are we spoiled by too much convenience?
- Are you prepared to give up any of the technological convenience for the environment? As a developer, not a consumer?