CRDTs for Fun and Eventual Profit

Starring **Noel Welsh**

An

_.underscore

production

Showing at

Big-O London

Why are we here?

CRDTs

(An overview)

#1

Motivation

#2

Algorithms

#3

Implementation

Conflict-free Replicated Data Type

Conflict-free Replicated Data Type

Conflict-free Replicated Data Type

Conflict-free Replicated Data Type

Conflict-free Convergent Commutative

Merge data automatically

#1

Motivation

Web Services

Web

DB

Web

Stateless web servers

RDMS ensures consistency

Consistency

All nodes see the same data

Simple

Implications

RDBMs can't scale writes

Distributed consistency increases latency

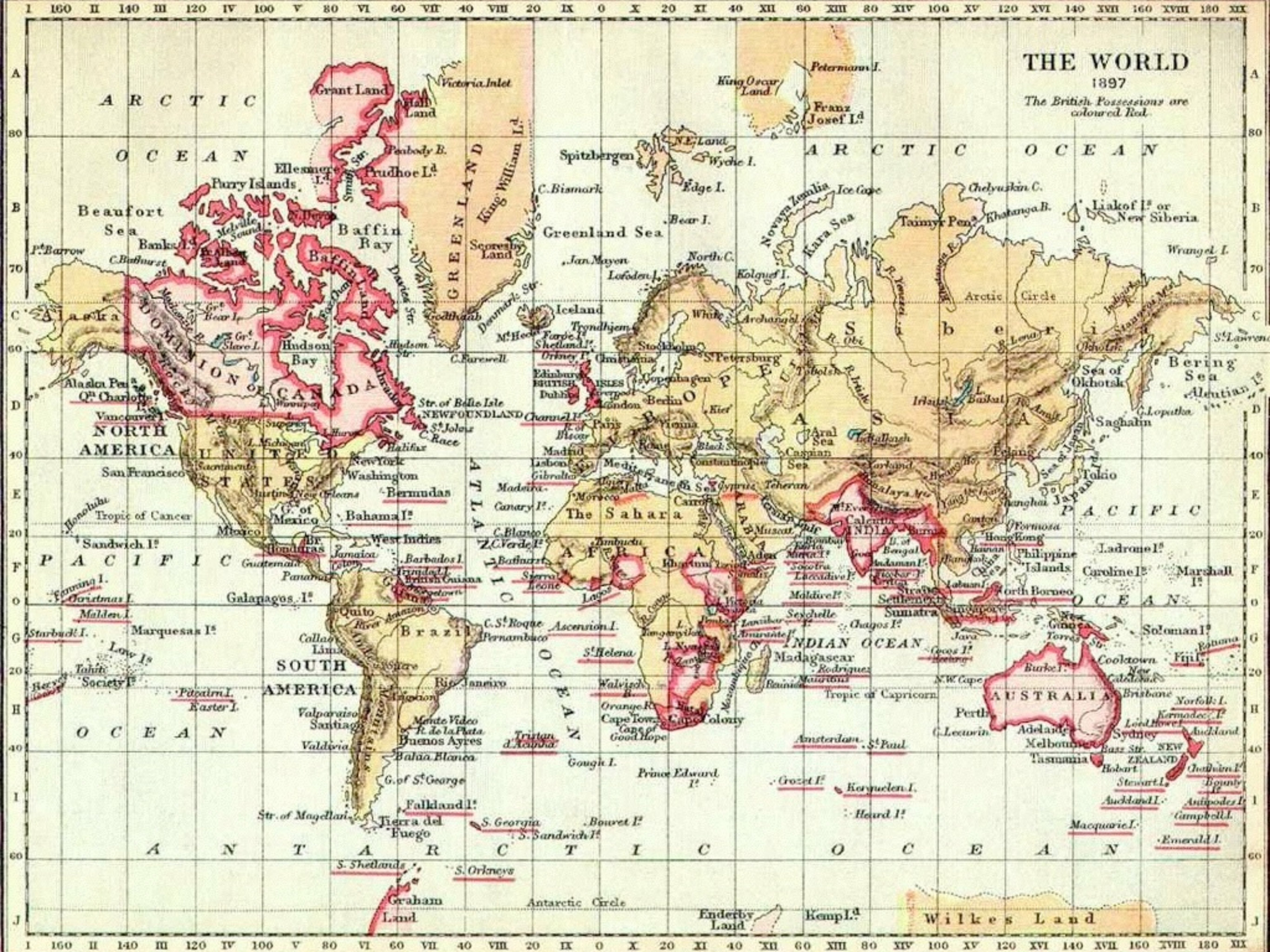
Monolithic codebase doesn't scale

Microservices

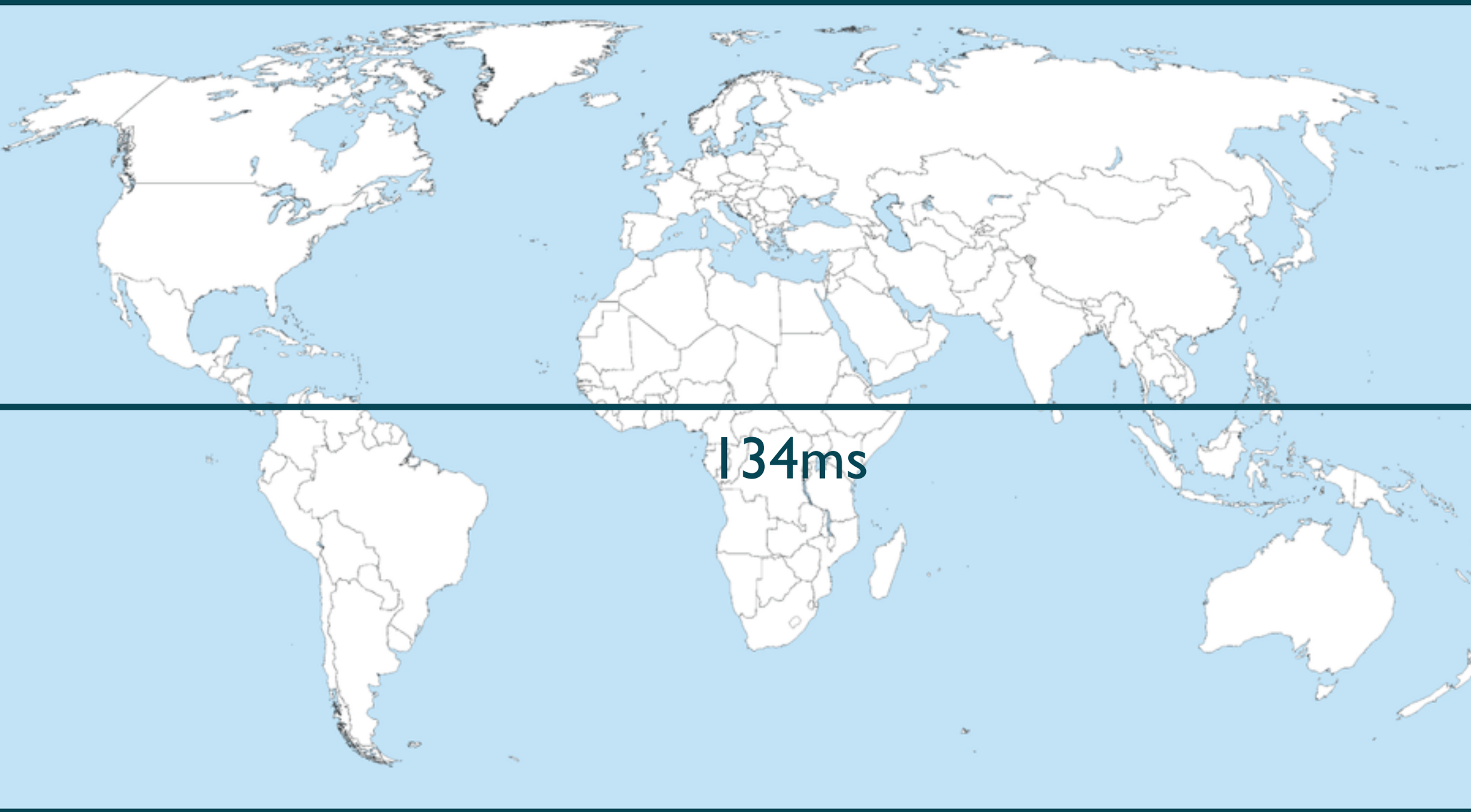
increases latency

Latency

We really want sub-Second page load

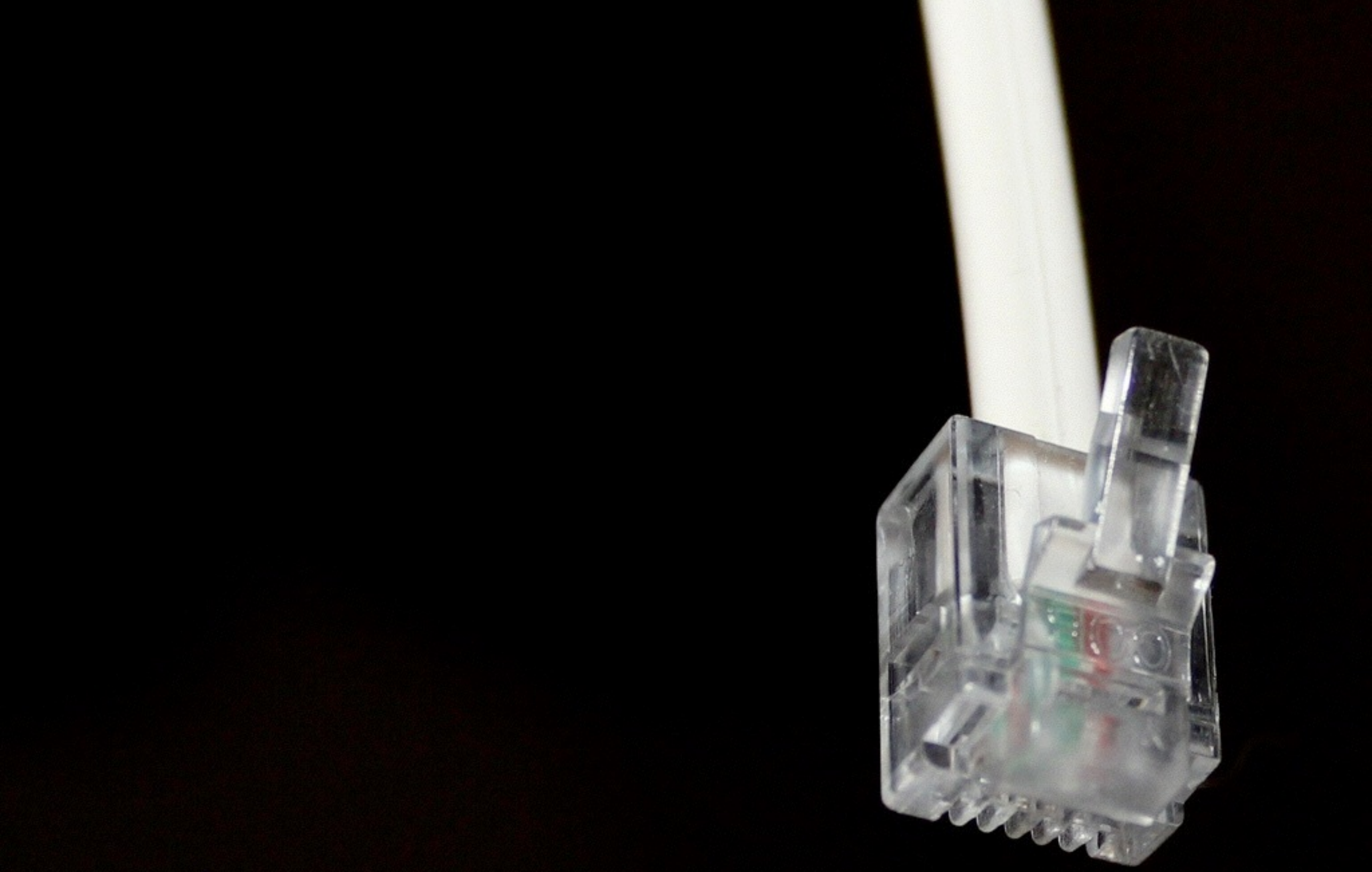

THE WORLD
1897
The British Possessions are coloured Red.
ARCTIC OCEAN
Grant Land
Victoria Inlet
King Oscar Land
Petermann I.
Franz Josef Ld
Spitzbergen
Greenland Sea
GREENLAND
King William Ld
Baffin Bay
Beaufort Sea
Parry Islands
Ellesmere Ld
Prudhoe Ld
Banks I.
Pt Barrow
Iceland
Jan Mayen
Novaya Zemlia
Kara Sea
Taimyr Pen.
Chelyuskin C.
Liakof Is or New Siberia
Wrangel I.
Alaska
DOMINION OF CANADA
Hudson Bay
Alaska Pen.
Vancouver
NORTH AMERICA
UNITED STATES
San Francisco
New York
Washington
NEWFOUNDLAND
Str. of Belle Isle
C. Race
Halifax
Bermudas
Bahama Is
West Indies
Jamaica
Barbados I.
Trinidad I.
British Guiana
G. of Mexico
Mexico
Honduras
Tropic of Cancer
Sandwich Is
Honolulu
PACIFIC OCEAN
Fanning I.
Christmas I.
Malden I.
Starbuck I.
Marquesas Is
Galapagos Is
Quito
Brazil
C. St Roque
Pernambuco
SOUTH AMERICA
Callao
Lima
Rio Janeiro
Valparaiso
Santiago
Valdivia
Monte Video
R. de la Plata
Buenos Ayres
Bahia Blanca
G. of St George
Falkland Is
Str. of Magellan
Tierra del Fuego
S. Georgia
S. Sandwich Is
S. Shetlands
S. Orkneys
Graham Land
Society Is
Tahiti
Pitcairn I.
Easter I.
ATLANTIC OCEAN
Ascension I.
St Helena
Tristan d'Acunha
Gough I.
Bouvet I.
Canary Is
Madeira
C. Verde Is
Sierra Leone
Lagos
C. Farewell
BRITISH ISLES
Edinburgh
Dublin
London
Channel Is
Paris
Berlin
Vienna
Madrid
Lisbon
Gibraltar
Mediterranean Sea
Malta
Cyprus
Morocco
Black Sea
Constantinople
Caspian Sea
Aral Sea
Teheran
Cairo
Stockholm
Christiania
Copenhagen
St Petersburg
Archangel
EUROPE
ASIA
Siberia
Arctic Circle
Tobolsk
R. Obi
R. Lena
L. Baikal
Sea of Okhotsk
Bering Sea
Aleutian Is
Saghalin
Sea of Japan
Japan
Tokio
Peking
Shanghai
Hong Kong
Formosa
Philippine Islands
Ladrone Is
Caroline Is
Marshall Is
Himalaya Mts
INDIA
Calcutta
Bombay
Ceylon
Burma
B. of Bengal
Andaman I.
Nicobar I.
Laccadive I.
Maldive Is
ARABIA
Muscat
Aden
Socotra
The Sahara
AFRICA
Timbuctu
Khartum
Zanzibar
Seychelle
Madagascar
Mauritius
Rodriguez
Natal
Cape Colony
Cape Town
Cape of Good Hope
Walvisch B.
Orange R.
Prince Edward Is
Crozet Is
Kerguelen I.
Heard I.
Amsterdam
St Paul
INDIAN OCEAN
Chagos Is
Cocos Is
Tropic of Capricorn
Straits Settlements
Sumatra
Singapore
Java
Labuan I.
North Borneo
New Guinea
Torres Str.
Solomon Is
Fiji Is
New Caledonia
AUSTRALIA
Perth
Adelaide
Melbourne
Sydney
Brisbane
Cooktown
Bass Str.
Tasmania
Hobart
NEW ZEALAND
Auckland
Norfolk I.
Kermadec Is
Chatham I.
Bounty I.
Antipodes I.
Campbell I.
Auckland I.
Macquarie I.
Emerald I.
Stewart I.
ANTARCTIC OCEAN
Antarctic Circle
Enderby Land
Kemp Ld
Wilkes Land

$$E=MC^2$$



134ms

World map from Wikimedia Commons

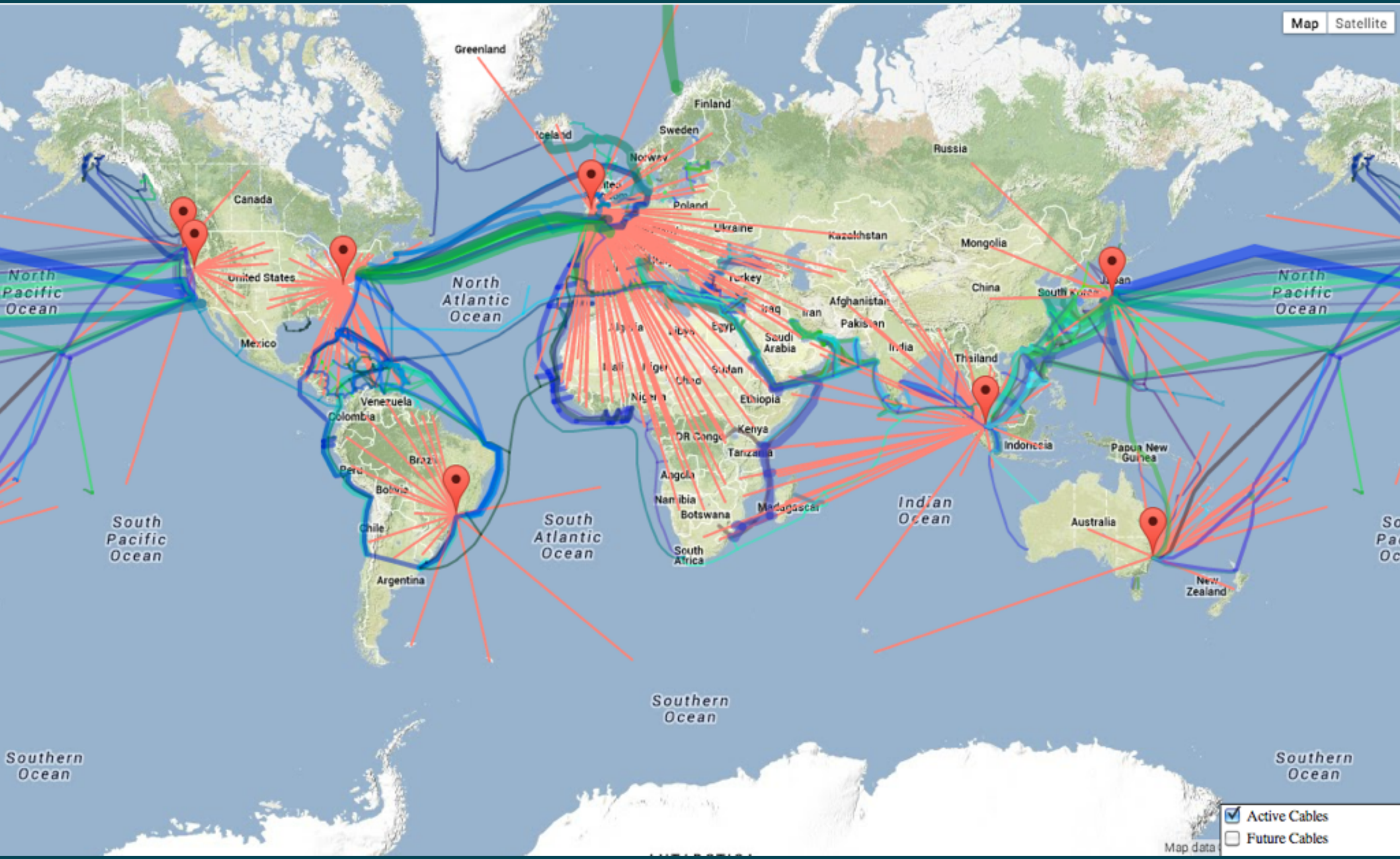


http://www.flickr.com/photos/21561428@N03/5185781936/

LOCATION

LOCATION

LOCATION



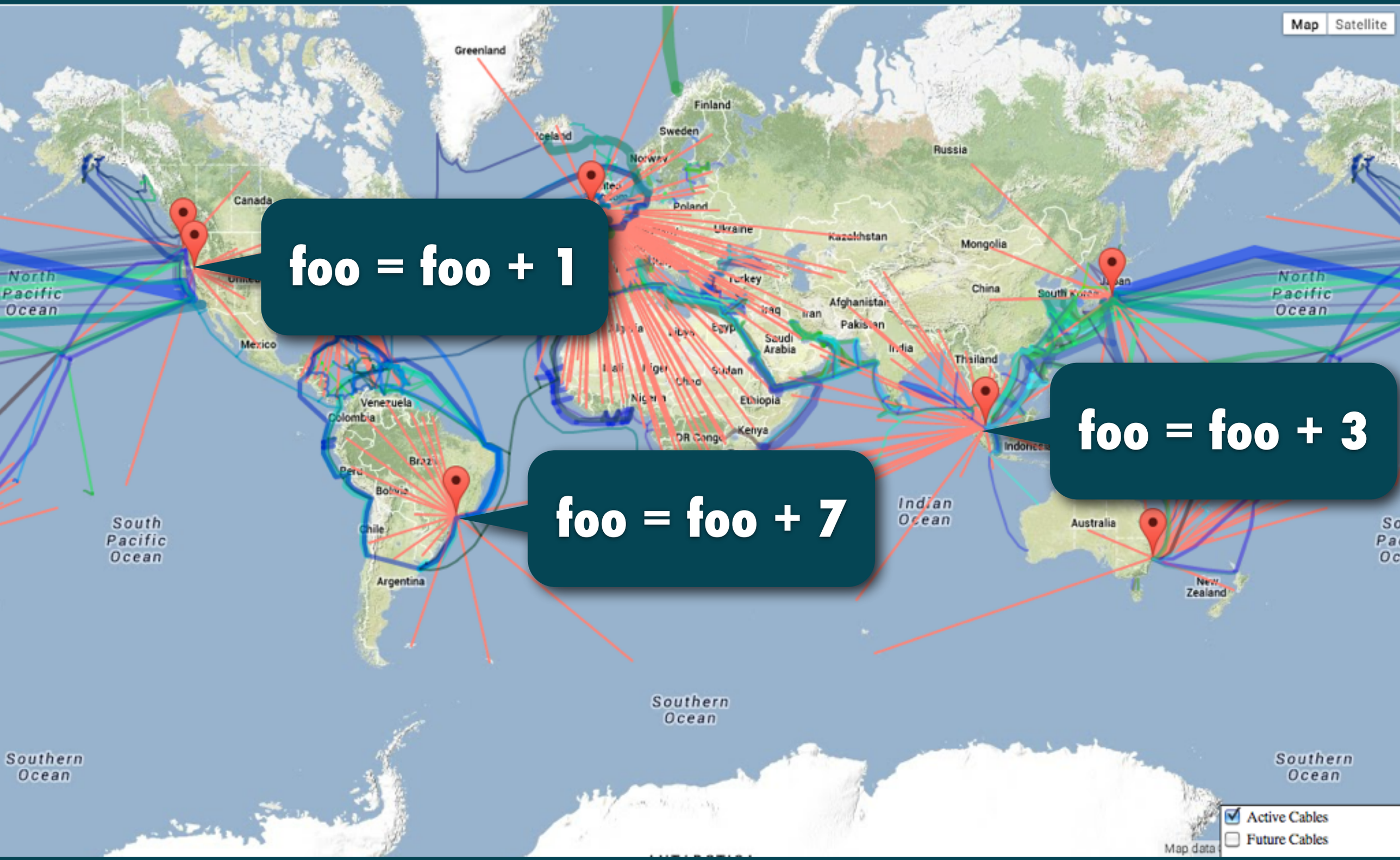
http://turnkeylinux.github.io/aws-datacenters/

Problem

SOLVED!

Problem

SOLVED?



http://turnkeylinux.github.io/aws-datacenters/

What the FOO?

(0,0,0)

foo = foo +1

(1,0,0) (1,1,0) (1,0,1) (1,1,1)

(1,7,0) (1,7,7)

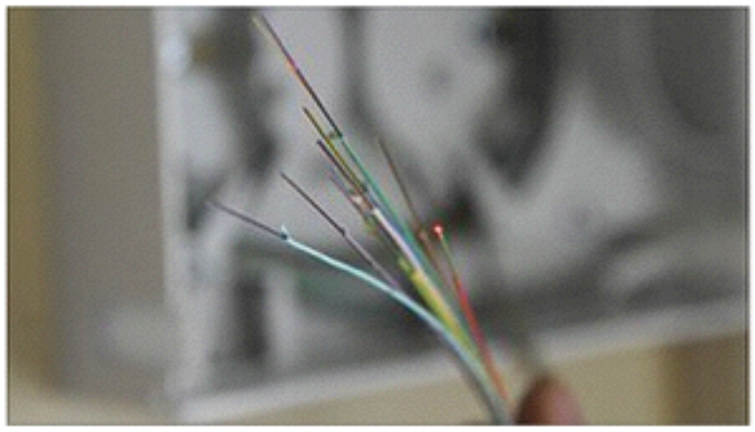
…

foo = foo +7

6 April 2011 Last updated at 12:39

Pensioner in Georgia cuts Armenia off from internet

An elderly woman in Georgia is facing a prison sentence after reportedly causing internet services in neighbouring Armenia to crash.



Fibre-optic cables carry services via Georgia to Armenia

The country found itself offline for hours on 28 March after cables linking Georgia to Armenia were damaged.

A Georgian interior ministry spokesman said a 75-year-old woman had admitted damaging fibre-optic cables while scavenging for copper.

She has been charged and reportedly faces up to three years in prison.

"Taking into account her advancing years, she has been released pending the end of the investigation and subsequent trial," spokesman Zura Gvenetadze told AFP news agency.

She had been searching for copper in the Georgian village of Ksani.

The cables, owned by the Georgian Railway Telecom company, serve eastern Georgia, Armenia and Azerbaijan.

All three wholesale internet providers in Armenia - ArmenTel, FiberNet Communication and GNC-Alfa - were unable to provide their usual service on the evening of 28 March, Armenia's Arka news agency reported.

Services were eventually restored after midnight.

Top Stories

Osborne's 'Budget for resilient UK'

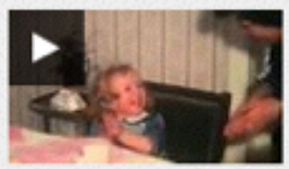
Pro-Russians storm Crimea naval base

China widens ship search for plane

UK unemployment falls by 63,000 NEW

New clues in Claudia Lawrence hunt

Features



On the doorstep

Adopted woman's search for birth mother ends five minutes away

Under the sun

Solar power for African school computers

Royal kidnap

The day I took three bullets for Princess Anne

Outback Bowie

When a pop star took a remote Aussie town by storm

Forgotten man

Who pipped Tony Benn to longest-serving Labour MP title?

We have conquered

Latency

We have lost

Consistency

CAP Theorem

Consistency
Availability
Partition Tolerance

Choose 2

Solution #1 (Google)

Simply use atomic clocks to establish temporal ordering of events and distributed transactions ...

Solution #2 (Eventual Consistency)

Reconcile diverging values to eventually arise at consistency

How?

#2

Algorithms

CRDTs

G-Counter

A counter that can only GROW

G-Counter insight: Store a separate counter for each machine

G-Counter

Machine A

A:0

B:0

Machine B

A:0

B:0

A machine can only increment its own counter

G-Counter

Machine A

A:5

B:0

Machine B

A:0

B:7

Merge is simply the

max

G-Counter

Machine A

A:5

B:7

Machine B

A:5

B:7

The counter's value is simply the total

G-Counter

Machine A

A:5

B:7

Machine B

A:5

B:7

Total is 12

We have a distributed eventually-consistent increment-only counter

Can we abstract this idea?

```
trait GCounter[Id,Elt] {
  def inc(id: Id, amt: Elt)
  def total: Elt
  def merge(c: GCounter[Id, Elt]): GCounter[Id, Elt]

}
```

total requires Elt has +

inc requires Elt has +,0

+ must be

Invariant to order

Formally:

Associative

$$(x \bullet y) \bullet z = x \bullet (y \bullet z)$$

Formally:

Commutative

$$x \bullet y = y \bullet x$$

A Commutative Monoid!

High 5!

```
def inc(id: Id, amt: Elt)
(implicit m: Monoid[Elt])
```

```
def total(implicit m:
Monoid[Elt]): Elt
```

merge requires

Elt has max

max must be

Invariant to

order

Formally:

Associative

$$(x \bullet y) \bullet z = x \bullet (y \bullet z)$$

Formally:

Commutative

$$x \bullet y = y \bullet x$$

max must be

Invariant to

repeated

application

Formally:

Idempotent

An Idempotent Commutative Monoid!!!

```
def merge(c: GCounter[Id, Elt])
(implicit m: Monoid[Elt @@
Max]): GCounter[Id, Elt]
```

Number (+)
Number (*)
Tuple
Map
Option
Average
Moving average
t-digest
Set (intersection)
Set (union)
Hyperloglog
Bloom filter
Count-min
Vector
Q-Tree
SGD

G-Set

Machine A

A:{x}

B:{}

Machine B

A:{}

B:{y,z}

G-Set Merge

Machine A

A:{x}

B:{y,z}

Machine B

A:{x}

B:{y,z}

G-Set Total

Machine A

A:{x}

B:{y,z}

Machine B

A:{x}

B:{y,z}

Set is

{x,y,z}

PN-Counter

A counter that can GROW and SHRINK

Can't use a G-Counter as we can't use max to merge

Use

Two

G-counters!

PN-Counter

Machine A

Additions
A: 4, B: 2

Subtractions
A: 5, B: 3

Machine B

Additions
A: 4, B: 7

Subtractions
A: 3, B: 4

Merge is simply the MAX

PN-Counter

Machine A

Additions
A: 4, B: 7

Subtractions
A: 5, B: 4

Machine B

Additions
A: 4, B: 7

Subtractions
A: 5, B: 4

The counter's value is simply the TOTAL

PN-Counter

Machine A

Additions
A: 4, B: 7

Subtractions
A: 5, B: 4

Machine B

Additions
A: 4, B: 7

Subtractions
A: 5, B: 4

Total is

(4+7) - (5+4)=

2

```
trait PNCounter[Id,Elt] {

  def inc(id: Id, amt: Elt)

  def dec(id: Id, amt: Elt)

  def total: Elt

  def merge(c: PNCounter[Id, Elt]):
PNCounter[Id, Elt]

}
```

PN-Counter requires Elt has addition, zero, and subtraction

A Commutative Group!

```scala
trait PNCounter[Id,Elt] {

  def inc(id: Id, amt: Elt)(implicit m:
Monoid[Elt])

  def dec(id: Id, amt: Elt)(implicit m:
Monoid[Elt])

  def total(implicit m: Group[Elt]): Elt

  def merge(c: PNCounter[Id, Elt])(implicit m:
Monoid[Elt @@ Max]): PNCounter[Id, Elt]

}
```

Numbers are clearly a commutative group

Sets with union and set difference are a commutative group

PN-Set

2P-Set

Machine A

Additions

A: {x}, B: {y}

Subtractions

A: {x}, B: {}

Machine B

Additions

A: {x}, B: {y, z}

Subtractions

A: {}, B: {y}

2P-Set Merge

Machine A

Additions

A: {x}, B: {y, z}

Subtractions

A: {x}, B: {y}

Machine B

Additions

A: {x}, B: {y, z}

Subtractions

A: {x}, B: {y}

2P-Set Total

Machine A

Additions

A: {x}, B: {y, z}

Subtractions

A: {x}, B: {y}

Machine B

Additions

A: {x}, B: {y, z}

Subtractions

A: {x}, B: {y}

Set is

{z}

Deleted elements stored indefinitely.

Called

tombstones

2P-Set allows elements to be added and removed **once**

C-Set

Store element and count

C-Set

Machine A

Additions

A: {(x, 2)},

B: {(y, 1)}

Subtractions

A: {(x, 1)},

B: {}

Machine B

Additions

A: {(x, 1)},

B: {(y, 1), (z, 2)}

Subtractions

A: {},

B: {(y, 1)}

C-Set Merge

Machine A

Additions

A: {(x, 2)},

B: {(y, 1), (z, 2)}

Subtractions

A: {(x, 1)},

B: {(y, 1)}

Machine B

Additions

A: {(x, 2)},

B: {(y, 1), (z, 2)}

Subtractions

A: {(x, 1)},

B: {(y, 1)}

C-Set Total

Machine A

Additions

A: {(x, 2)},

B: {(y, 1), (z, 2)}

Subtractions

A: {(x, 1)},

B: {(y, 1)}

Machine B

Additions

A: {(x, 2)},

B: {(y, 1), (z, 2)}

Subtractions

A: {(x, 1)},

B: {(y, 1)}

Set is

{x, z}

C-Set allows elements to be added and removed many times

C-Set allows elements to be removed

more times than they have been added

OR-Set

Store element and unique token

OR-Set

Machine A

Additions

A: {(x, #a), (x, #d)},
B: {(y, #b)}

Subtractions

A: {(x, #a)},
B: {}

Machine B

Additions

A: {(x, #a)},
B: {(y, #b), (z, #c)}

Subtractions

A: {},
B: {(y, #b)}

OR-Set Merge

Machine A

Additions

A: {(x, #a), (x, #d)},
B: {(y, #b), (z, #c)}

Subtractions

A: {(x, #a)},
B: {(y, #b)}

Machine B

Additions

A: {(x, #a), (x, #d)},
B: {(y, #b), (z, #c)}

Subtractions

A: {(x, #a)},
B: {(y, #b)}

OR-Set Total

Machine A

Additions

A: {(x, #a), (x, #a)},

B: {(y, #b), (z, #c)}

Subtractions

A: {(x, #a)},

B: {(y, #b)}

Machine B

Additions

A: {(x, #a), (x, #d)},

B: {(y, #b), (z, #c)}

Subtractions

A: {(x, #a)},

B: {(y, #b)}

Set is

{x, z}

OR-Set works the way we expect

From sets, build

trees, graphs,

etc.

#3

Implementation

Architecture Memory Usage Related Research

Architecture

AP Systems

Dynamo, Riak, Cassandra, etc.

Gossip Protocol
Failure Detector
Consistent Hashing

Very simple operationally

Riak 2.0

Akka Cluster

(AP framework in Scala)

Soundcloud Roshi

(limited CRDT)

Twitter Summingbird

(monoids)

Memory Usage

Tombstones

Machine IDs

Tombstones: Establish causal order and delete (Bieniusa et al. 2012)

Machine IDs: Hierarchical organisation allows pruning (Almeida & Baquero. 2013)

Machine IDs: OR-Sets don't need them

Tombstones: Prune with heuristics (often based on time)

Establish consistency and delete

Related Work

Monotone logic

LVars

Summary

#1

Eventually consistent data is natural

#2

CRDTs are a simple solution

#3

Building AP systems is well established


Thank you!
Now go forth and
DISTRIBUTE!
http://stjost.deviantart.com/art/Stomping-Off-Into-the-Sunset-277086274

More:

underscoreconsulting.com