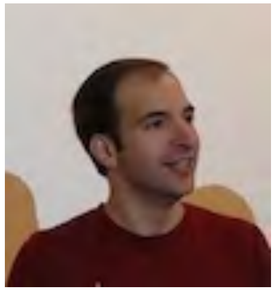
The State of Contemporary Computing Substrates for Optimization Methods

Benjamin Recht
UC Berkeley

My quixotic quest for superlinear algorithms

Benjamin Recht
UC Berkeley

Collaborators





Slides extracted by torturing Eric Jonas, Vaishaal Shankar, Stephen Tu, Shivaram Venkataraman, and Ashia Wilson

The holy grail of optimization

Fix a positive definite, n x n matrix A

Find x such that $Ax = b$

$$\text{minimize} \quad \frac{1}{2}x^T Ax - b^T x$$

Linear regression/classification
Interior Point Methods
Sum of Squares

- Quadratic Programming
- Sequential Quadratic Programming
- Newton's Method

Hard case for Nesterov

Dense case ~ bigger sparse cases

Iterative Solvers

Fix a positive definite, n x n matrix A

Find x such that $Ax = b$

complexity = (number of iterations) x (flops per iteration)

$$O\left(\frac{L}{\mu}\log(1/\epsilon)\right) \qquad O(n^2)$$

$$x_{k+1} = x_k - s(Ax_k - b)$$

Steepest Descent

Iterative Solvers

Fix a positive definite, n x n matrix A

Find x such that $Ax = b$

complexity = (number of iterations) x (flops per iteration)

$$O\left(n\frac{L_{\max}}{\mu}\log(1/\epsilon)\right) \qquad O(n)$$

$$x_{k+1}^{(i)} = x_k^{(i)} - A_{ii}^{-1}(a_i^T x_k - b_i)$$

Coordinate Descent
(Gauss-Seidel)

Iterative Solvers

Fix a positive definite, n x n matrix A

Find x such that $Ax = b$

complexity = (number of iterations) x (flops per iteration)

$$\approx O\left(n^2 K(A) \log(1/\epsilon)\right)$$

Not sublinear in dimension….

direct solve complexity = $O(n^3)$

Iterative or Direct?

Fix a positive definite, n x n matrix A

Find x such that $Ax = b$

n=60,000

b 60,000 x 10

iterative complexity $\approx O\left(n^2 K(A) \log(1/\epsilon)\right)$

direct solve complexity = $O(n^3)$

direct solve wall clock = 111s

iterative wall clock = 127s

$$\frac{\|x - x_\star\|_F}{\|x_\star\|_F} = 0.58$$

Are we selling ourselves short with these crummy iterative methods?

Why is direct faster?

iterative complexity $\approx O\left(n^2 K(A) \log(1/\epsilon)\right)$

direct solve complexity = $O(n^3)$

n=60,000

b 60,000 x 10

direct solve wall clock = 111s

iterative wall clock = 127s

$$\frac{\|x - x_\star\|_F}{\|x_\star\|_F} = 0.58$$

Are we selling ourselves short with these crummy iterative methods?

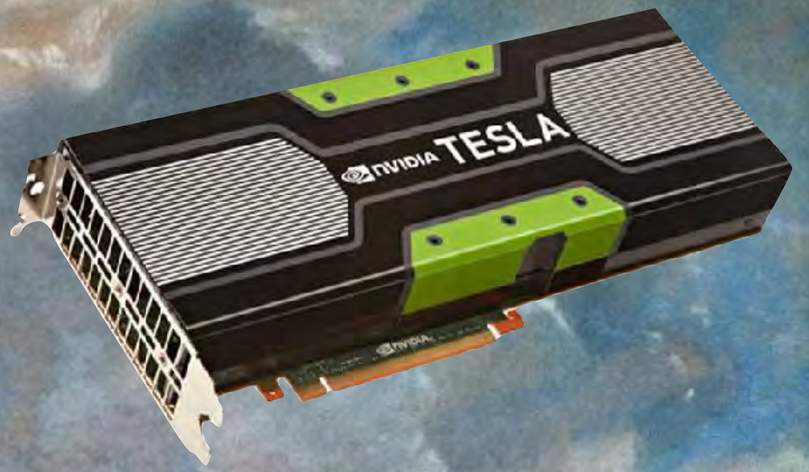
- To achieve maximum speed, data shuffling is required
- Memory bandwidth is critical
- Maybe we just need more RAM?

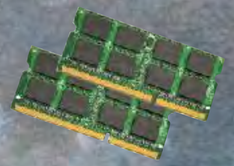
I NEED MOAR RAM

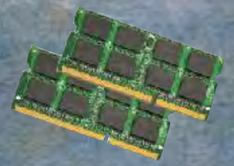
n	RAM	Time
60,000	30GB	111s
350K	1TB	6 hours
1.2M	12TB	10 days

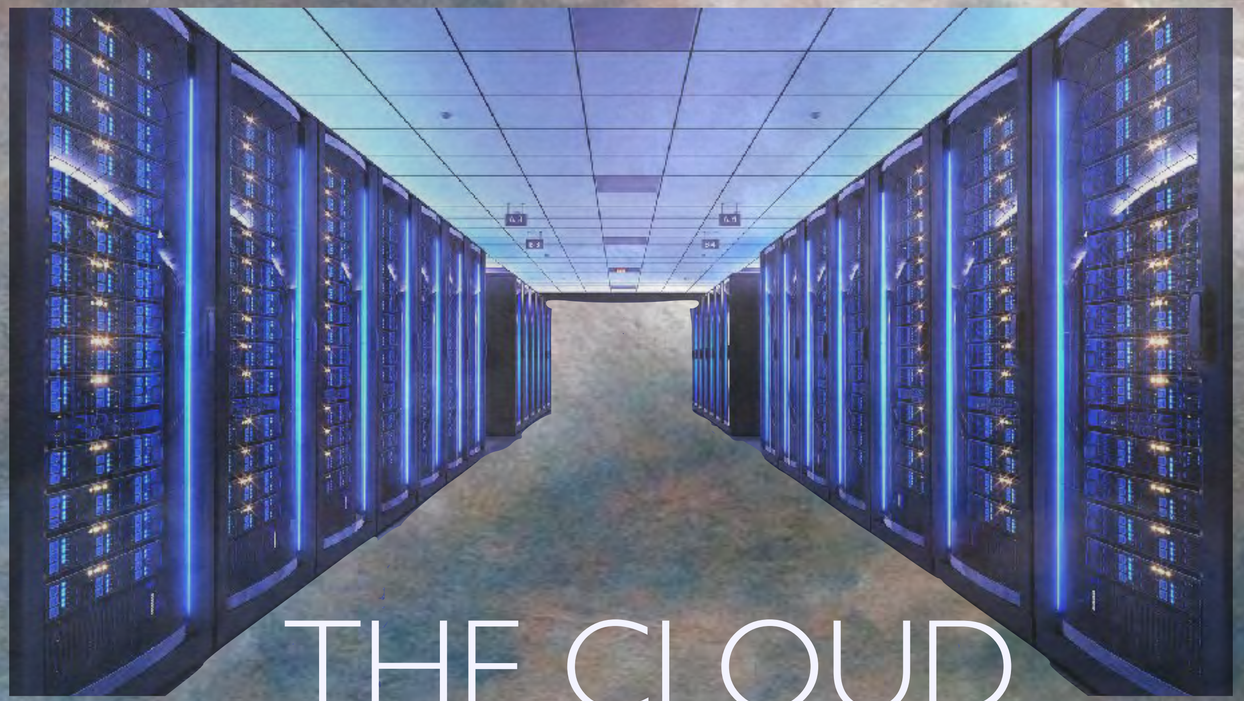
Time extrapolation based on 64 cores, 2TB RAM

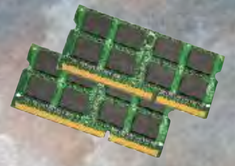
- Superlinear scaling is the worst.
- Memory demand growing quadratically means more than one computer is needed for anything larger than 400K.
- Cubic time scaling means a lot of cores are needed to minimize time.
- Where can I get all of this compute?

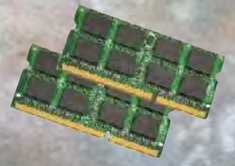


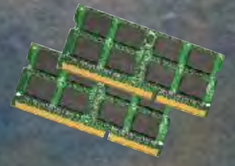












THE CLOUD

Cloud Computing CHOICEs

t2.nano, t2.micro, t2.small
m4.large, m4.xlarge, m4.2xlarge,
m4.4xlarge, m3.medium, c4.large,
c4.xlarge, c4.2xlarge,
c3.large, c3.xlarge, c3.4xlarge,
r3.large, r3.xlarge, r3.4xlarge,
i2.2xlarge, i2.4xlarge, d2.xlarge
d2.2xlarge, d2.4xlarge,…

Amazon EC2

Basic tier: A0, A1, A2, A3, A4
Optimized Compute : D1, D2, D3, D4,
D11, D12, D13
D1v2, D2v2, D3v2, D11v2,…
Latest CPUs: G1, G2, G3, …
Network Optimized: A8, A9
Compute Intensive: A10, A11,…

Microsoft Azure

n1-standard-1, ns1-standard-2, ns1-standard-4, ns1-standard-8, ns1-standard-16, ns1highmem-2, ns1-highmem-4, ns1-highmem-8, n1-highcpu-2, n1-highcpu-4, n1-highcpu-8, n1-highcpu-16, n1-highcpu-32, f1-micro, g1-small…

Google Cloud Engine

EC2Instances.info Easy Amazon EC2 Instance Comparison

EC2 RDS

Region: US East (N. Virginia) | Cost: Hourly | Reserved: 1 yr - No Upfront | Columns | Compare Selected | Clear Filters

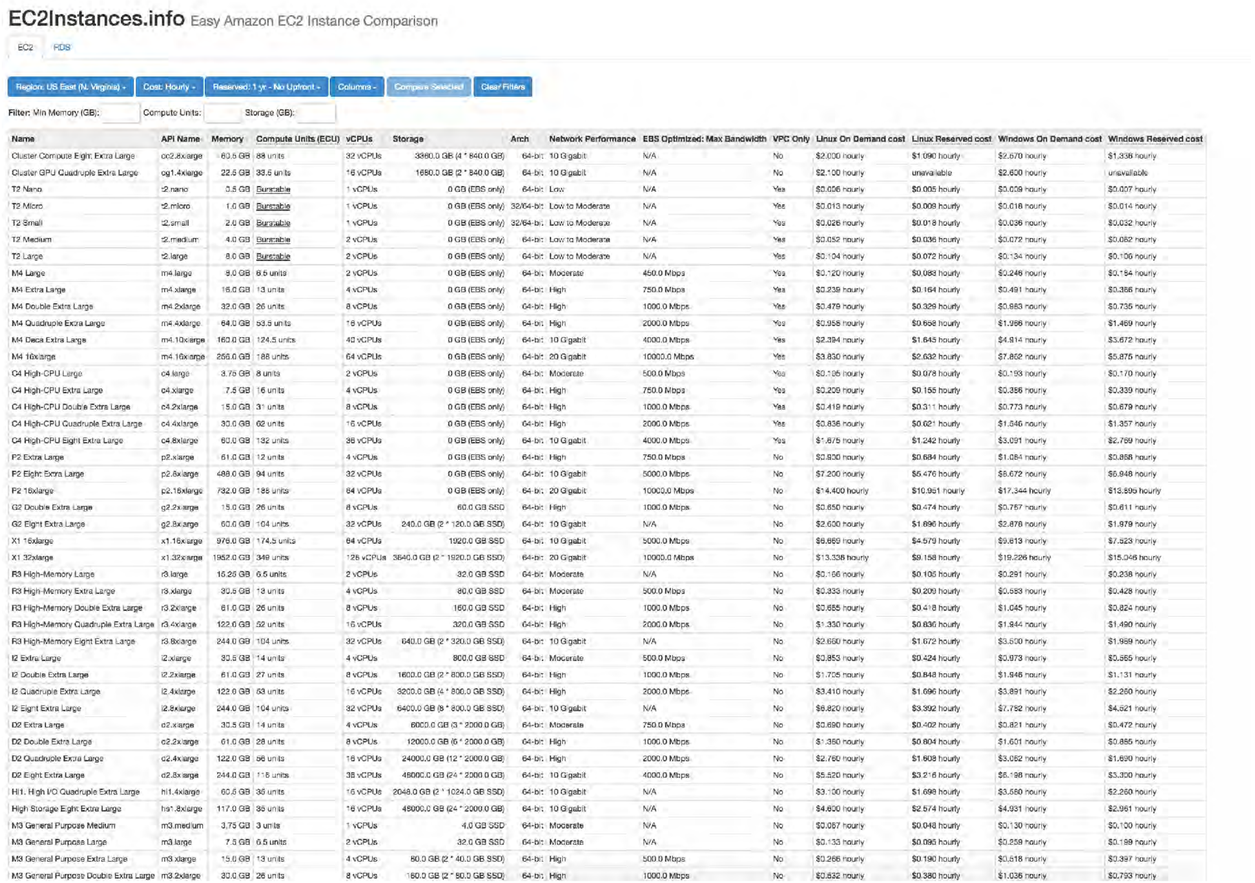
Filter: Min Memory (GB): Compute Units: Storage (GB):

Name	API Name	Memory	Compute Units (ECU)	vCPUs	Storage	Arch	Network Performance	EBS Optimized: Max Bandwidth	VPC Only	Linux On Demand cost	Linux Reserved cost	Windows On Demand cost	Windows Reserved cost
Cluster Compute Eight Extra Large	cc2.8xlarge	60.5 GB	88 units	32 vCPUs	3360.0 GB (4 * 840.0 GB)	64-bit	10 Gigabit	N/A	No	$2.000 hourly	$1.090 hourly	$2.570 hourly	$1.336 hourly
Cluster GPU Quadruple Extra Large	cg1.4xlarge	22.5 GB	33.5 units	16 vCPUs	1680.0 GB (2 * 840.0 GB)	64-bit	10 Gigabit	N/A	No	$2.100 hourly	unavailable	$2.600 hourly	unavailable
T2 Nano	t2.nano	0.5 GB	Burstable	1 vCPUs	0 GB (EBS only)	64-bit	Low	N/A	Yes	$0.006 hourly	$0.005 hourly	$0.009 hourly	$0.007 hourly
T2 Micro	t2.micro	1.0 GB	Burstable	1 vCPUs	0 GB (EBS only)	32/64-bit	Low to Moderate	N/A	Yes	$0.013 hourly	$0.009 hourly	$0.018 hourly	$0.014 hourly
T2 Small	t2.small	2.0 GB	Burstable	1 vCPUs	0 GB (EBS only)	32/64-bit	Low to Moderate	N/A	Yes	$0.026 hourly	$0.018 hourly	$0.036 hourly	$0.032 hourly
T2 Medium	t2.medium	4.0 GB	Burstable	2 vCPUs	0 GB (EBS only)	64-bit	Low to Moderate	N/A	Yes	$0.052 hourly	$0.036 hourly	$0.072 hourly	$0.062 hourly
T2 Large	t2.large	8.0 GB	Burstable	2 vCPUs	0 GB (EBS only)	64-bit	Low to Moderate	N/A	Yes	$0.104 hourly	$0.072 hourly	$0.134 hourly	$0.106 hourly
M4 Large	m4.large	8.0 GB	6.5 units	2 vCPUs	0 GB (EBS only)	64-bit	Moderate	450.0 Mbps	Yes	$0.120 hourly	$0.083 hourly	$0.246 hourly	$0.184 hourly
M4 Extra Large	m4.xlarge	16.0 GB	13 units	4 vCPUs	0 GB (EBS only)	64-bit	High	750.0 Mbps	Yes	$0.239 hourly	$0.164 hourly	$0.491 hourly	$0.366 hourly
M4 Double Extra Large	m4.2xlarge	32.0 GB	26 units	8 vCPUs	0 GB (EBS only)	64-bit	High	1000.0 Mbps	Yes	$0.479 hourly	$0.329 hourly	$0.983 hourly	$0.735 hourly
M4 Quadruple Extra Large	m4.4xlarge	64.0 GB	53.5 units	16 vCPUs	0 GB (EBS only)	64-bit	High	2000.0 Mbps	Yes	$0.958 hourly	$0.658 hourly	$1.966 hourly	$1.469 hourly
M4 Deca Extra Large	m4.10xlarge	160.0 GB	124.5 units	40 vCPUs	0 GB (EBS only)	64-bit	10 Gigabit	4000.0 Mbps	Yes	$2.394 hourly	$1.645 hourly	$4.914 hourly	$3.672 hourly
M4 16xlarge	m4.16xlarge	256.0 GB	188 units	64 vCPUs	0 GB (EBS only)	64-bit	20 Gigabit	10000.0 Mbps	Yes	$3.830 hourly	$2.632 hourly	$7.862 hourly	$5.875 hourly
C4 High-CPU Large	c4.large	3.75 GB	8 units	2 vCPUs	0 GB (EBS only)	64-bit	Moderate	500.0 Mbps	Yes	$0.105 hourly	$0.078 hourly	$0.193 hourly	$0.170 hourly
C4 High-CPU Extra Large	c4.xlarge	7.5 GB	16 units	4 vCPUs	0 GB (EBS only)	64-bit	High	750.0 Mbps	Yes	$0.209 hourly	$0.155 hourly	$0.386 hourly	$0.339 hourly
C4 High-CPU Double Extra Large	c4.2xlarge	15.0 GB	31 units	8 vCPUs	0 GB (EBS only)	64-bit	High	1000.0 Mbps	Yes	$0.419 hourly	$0.311 hourly	$0.773 hourly	$0.679 hourly
C4 High-CPU Quadruple Extra Large	c4.4xlarge	30.0 GB	62 units	16 vCPUs	0 GB (EBS only)	64-bit	High	2000.0 Mbps	Yes	$0.836 hourly	$0.621 hourly	$1.546 hourly	$1.357 hourly
C4 High-CPU Eight Extra Large	c4.8xlarge	60.0 GB	132 units	36 vCPUs	0 GB (EBS only)	64-bit	10 Gigabit	4000.0 Mbps	Yes	$1.675 hourly	$1.242 hourly	$3.091 hourly	$2.769 hourly
P2 Extra Large	p2.xlarge	61.0 GB	12 units	4 vCPUs	0 GB (EBS only)	64-bit	High	750.0 Mbps	No	$0.930 hourly	$0.684 hourly	$1.084 hourly	$0.868 hourly
P2 Eight Extra Large	p2.8xlarge	488.0 GB	94 units	32 vCPUs	0 GB (EBS only)	64-bit	10 Gigabit	5000.0 Mbps	No	$7.200 hourly	$5.476 hourly	$8.672 hourly	$6.948 hourly
P2 16xlarge	p2.16xlarge	732.0 GB	188 units	64 vCPUs	0 GB (EBS only)	64-bit	20 Gigabit	10000.0 Mbps	No	$14.400 hourly	$10.951 hourly	$17.344 hourly	$13.895 hourly
G2 Double Extra Large	g2.2xlarge	15.0 GB	26 units	8 vCPUs	60.0 GB SSD	64-bit	High	1000.0 Mbps	No	$0.650 hourly	$0.474 hourly	$0.767 hourly	$0.611 hourly
G2 Eight Extra Large	g2.8xlarge	60.0 GB	104 units	32 vCPUs	240.0 GB (2 * 120.0 GB SSD)	64-bit	10 Gigabit	N/A	No	$2.600 hourly	$1.896 hourly	$2.878 hourly	$1.979 hourly
X1 16xlarge	x1.16xlarge	976.0 GB	174.5 units	64 vCPUs	1920.0 GB SSD	64-bit	10 Gigabit	5000.0 Mbps	No	$6.669 hourly	$4.579 hourly	$9.613 hourly	$7.523 hourly
X1 32xlarge	x1.32xlarge	1952.0 GB	349 units	128 vCPUs	3840.0 GB (2 * 1920.0 GB SSD)	64-bit	20 Gigabit	10000.0 Mbps	No	$13.338 hourly	$9.158 hourly	$19.226 hourly	$15.046 hourly
R3 High-Memory Large	r3.large	15.25 GB	6.5 units	2 vCPUs	32.0 GB SSD	64-bit	Moderate	N/A	No	$0.166 hourly	$0.105 hourly	$0.291 hourly	$0.238 hourly
R3 High-Memory Extra Large	r3.xlarge	30.5 GB	13 units	4 vCPUs	80.0 GB SSD	64-bit	Moderate	500.0 Mbps	No	$0.333 hourly	$0.209 hourly	$0.583 hourly	$0.428 hourly
R3 High-Memory Double Extra Large	r3.2xlarge	61.0 GB	26 units	8 vCPUs	160.0 GB SSD	64-bit	High	1000.0 Mbps	No	$0.665 hourly	$0.418 hourly	$1.045 hourly	$0.824 hourly
R3 High-Memory Quadruple Extra Large	r3.4xlarge	122.0 GB	52 units	16 vCPUs	320.0 GB SSD	64-bit	High	2000.0 Mbps	No	$1.330 hourly	$0.836 hourly	$1.944 hourly	$1.490 hourly
R3 High-Memory Eight Extra Large	r3.8xlarge	244.0 GB	104 units	32 vCPUs	640.0 GB (2 * 320.0 GB SSD)	64-bit	10 Gigabit	N/A	No	$2.660 hourly	$1.672 hourly	$3.500 hourly	$1.989 hourly
I2 Extra Large	i2.xlarge	30.5 GB	14 units	4 vCPUs	800.0 GB SSD	64-bit	Moderate	500.0 Mbps	No	$0.853 hourly	$0.424 hourly	$0.973 hourly	$0.565 hourly
I2 Double Extra Large	i2.2xlarge	61.0 GB	27 units	8 vCPUs	1600.0 GB (2 * 800.0 GB SSD)	64-bit	High	1000.0 Mbps	No	$1.705 hourly	$0.848 hourly	$1.946 hourly	$1.131 hourly
I2 Quadruple Extra Large	i2.4xlarge	122.0 GB	53 units	16 vCPUs	3200.0 GB (4 * 800.0 GB SSD)	64-bit	High	2000.0 Mbps	No	$3.410 hourly	$1.696 hourly	$3.891 hourly	$2.260 hourly

TYRANNY of CHOICE

#THECLOUDISTOODAMNHARD

- What type? what instance? What base image?
- What is the cheapest configuration to run my job in 2 hours?
- How many to spin up? What price? spot?
- wait, Wait, WAIT oh god



EC2Instances.info Easy Amazon EC2 Instance Comparison

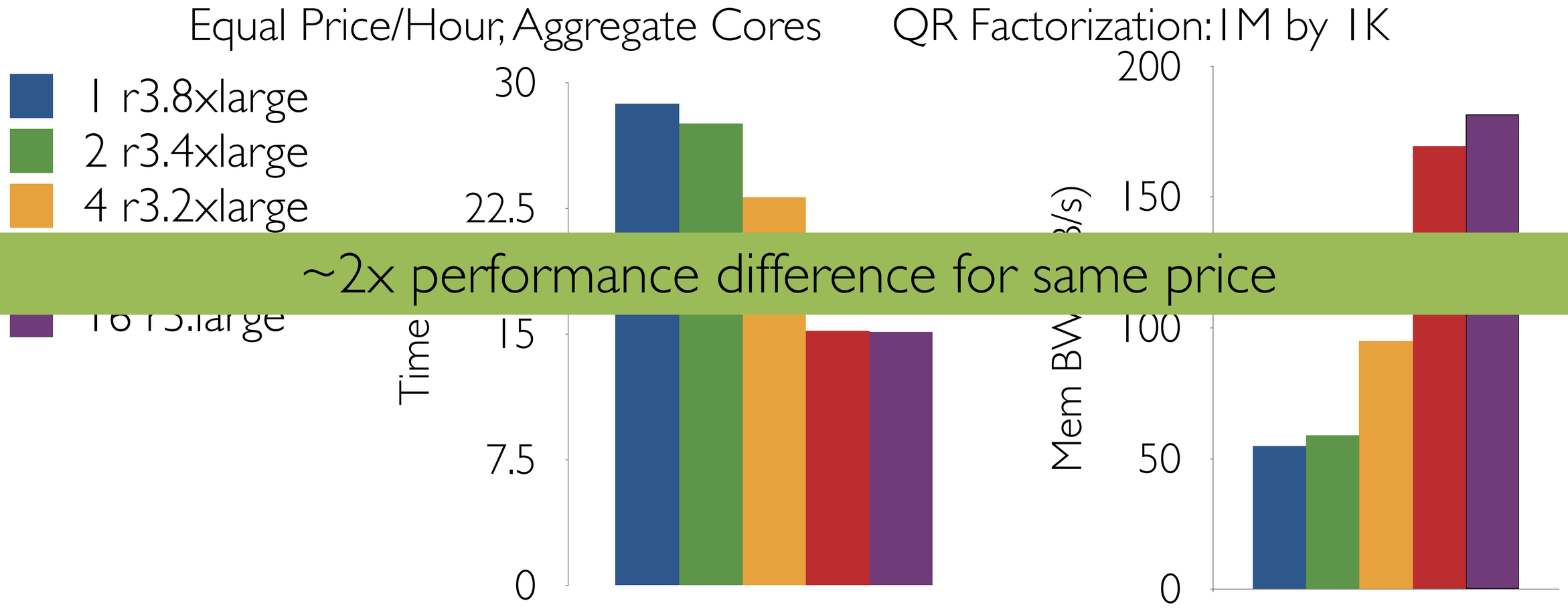
EC2 RDS

Region: US East (N. Virginia) Cost: Hourly Reserved: 1 yr - No Upfront Columns Compare Selected! Clear Filters

Filter: Min Memory (GB): Compute Units: Storage (GB):

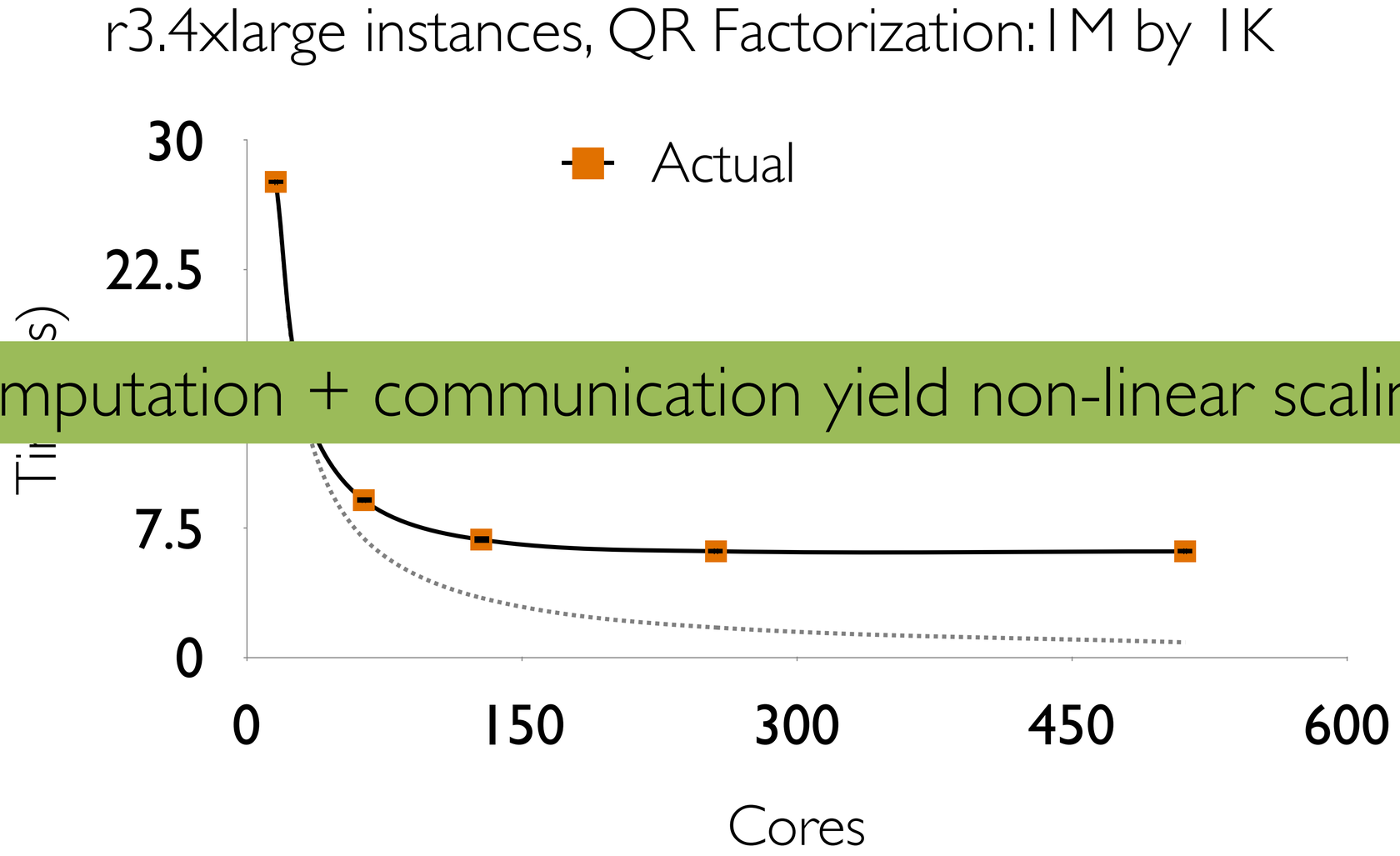
Name	API Name	Memory	Compute Units (ECU)	vCPUs	Storage	Arch	Network Performance	EBS Optimized: Max Bandwidth	VPC Only	Linux On Demand cost	Linux Reserved cost	Windows On Demand cost	Windows Reserved cost
Cluster Compute Eight Extra Large	cc2.8xlarge	60.5 GB	88 units	32 vCPUs	3360.0 GB (4 * 840.0 GB)	64-bit	10 Gigabit	N/A	No	$2.000 hourly	$1.090 hourly	$2.670 hourly	$1.336 hourly
Cluster GPU Quadruple Extra Large	cg1.4xlarge	22.5 GB	33.5 units	16 vCPUs	1690.0 GB (2 * 840.0 GB)	64-bit	10 Gigabit	N/A	No	$2.100 hourly	unavailable	$2.600 hourly	unavailable
T2 Nano	t2.nano	0.5 GB	Burstable	1 vCPUs	0 GB (EBS only)	64-bit	Low	N/A	Yes	$0.006 hourly	$0.005 hourly	$0.029 hourly	$0.027 hourly
T2 Micro	t2.micro	1.0 GB	Burstable	1 vCPUs	0 GB (EBS only)	32/64-bit	Low to Moderate	N/A	Yes	$0.013 hourly	$0.009 hourly	$0.016 hourly	$0.014 hourly
T2 Small	t2.small	2.0 GB	Burstable	1 vCPUs	0 GB (EBS only)	32/64-bit	Low to Moderate	N/A	Yes	$0.026 hourly	$0.018 hourly	$0.036 hourly	$0.032 hourly
T2 Medium	t2.medium	4.0 GB	Burstable	2 vCPUs	0 GB (EBS only)	64-bit	Low to Moderate	N/A	Yes	$0.052 hourly	$0.036 hourly	$0.072 hourly	$0.062 hourly
T2 Large	t2.large	8.0 GB	Burstable	2 vCPUs	0 GB (EBS only)	64-bit	Low to Moderate	N/A	Yes	$0.104 hourly	$0.072 hourly	$0.134 hourly	$0.106 hourly
M4 Large	m4.large	8.0 GB	6.5 units	2 vCPUs	0 GB (EBS only)	64-bit	Moderate	450.0 Mbps	Yes	$0.120 hourly	$0.080 hourly	$0.246 hourly	$0.184 hourly
M4 Extra Large	m4.xlarge	16.0 GB	13 units	4 vCPUs	0 GB (EBS only)	64-bit	High	750.0 Mbps	Yes	$0.239 hourly	$0.164 hourly	$0.491 hourly	$0.366 hourly
M4 Double Extra Large	m4.2xlarge	32.0 GB	26 units	8 vCPUs	0 GB (EBS only)	64-bit	High	1000.0 Mbps	Yes	$0.479 hourly	$0.329 hourly	$0.983 hourly	$0.735 hourly
M4 Quadruple Extra Large	m4.4xlarge	64.0 GB	53.5 units	16 vCPUs	0 GB (EBS only)	64-bit	High	2000.0 Mbps	Yes	$0.958 hourly	$0.658 hourly	$1.966 hourly	$1.469 hourly
M4 Deca Extra Large	m4.10xlarge	160.0 GB	124.5 units	40 vCPUs	0 GB (EBS only)	64-bit	10 Gigabit	4000.0 Mbps	Yes	$2.394 hourly	$1.645 hourly	$4.914 hourly	$3.672 hourly
M4 16xlarge	m4.16xlarge	256.0 GB	188 units	64 vCPUs	0 GB (EBS only)	64-bit	20 Gigabit	10000.0 Mbps	Yes	$3.830 hourly	$2.632 hourly	$7.862 hourly	$5.875 hourly
C4 High-CPU Large	c4.large	3.75 GB	8 units	2 vCPUs	0 GB (EBS only)	64-bit	Moderate	500.0 Mbps	Yes	$0.105 hourly	$0.073 hourly	$0.193 hourly	$0.170 hourly
C4 High-CPU Extra Large	c4.xlarge	7.5 GB	16 units	4 vCPUs	0 GB (EBS only)	64-bit	High	750.0 Mbps	Yes	$0.209 hourly	$0.155 hourly	$0.386 hourly	$0.339 hourly
C4 High-CPU Double Extra Large	c4.2xlarge	15.0 GB	31 units	8 vCPUs	0 GB (EBS only)	64-bit	High	1000.0 Mbps	Yes	$0.419 hourly	$0.311 hourly	$0.773 hourly	$0.679 hourly
C4 High-CPU Quadruple Extra Large	c4.4xlarge	30.0 GB	62 units	16 vCPUs	0 GB (EBS only)	64-bit	High	2000.0 Mbps	Yes	$0.836 hourly	$0.621 hourly	$1.546 hourly	$1.357 hourly
C4 High-CPU Eight Extra Large	c4.8xlarge	60.0 GB	132 units	36 vCPUs	0 GB (EBS only)	64-bit	10 Gigabit	4000.0 Mbps	Yes	$1.675 hourly	$1.242 hourly	$3.091 hourly	$2.709 hourly
P2 Extra Large	p2.xlarge	61.0 GB	12 units	4 vCPUs	0 GB (EBS only)	64-bit	High	750.0 Mbps	No	$0.830 hourly	$0.684 hourly	$1.084 hourly	$0.858 hourly
P2 Eight Extra Large	p2.8xlarge	488.0 GB	94 units	32 vCPUs	0 GB (EBS only)	64-bit	10 Gigabit	5000.0 Mbps	No	$7.200 hourly	$5.476 hourly	$8.672 hourly	$6.848 hourly
P2 16xlarge	p2.16xlarge	732.0 GB	188 units	64 vCPUs	0 GB (EBS only)	64-bit	20 Gigabit	10000.0 Mbps	No	$14.400 hourly	$10.951 hourly	$17.344 hourly	$13.699 hourly
G2 Double Extra Large	g2.2xlarge	15.0 GB	26 units	8 vCPUs	60.0 GB SSD	64-bit	High	1000.0 Mbps	No	$0.650 hourly	$0.474 hourly	$0.767 hourly	$0.611 hourly
G2 Eight Extra Large	g2.8xlarge	60.0 GB	104 units	32 vCPUs	240.0 GB (2 * 120.0 GB SSD)	64-bit	10 Gigabit	N/A	No	$2.600 hourly	$1.896 hourly	$2.876 hourly	$1.979 hourly
X1 16xlarge	x1.16xlarge	976.0 GB	174.5 units	64 vCPUs	1920.0 GB SSD	64-bit	10 Gigabit	5000.0 Mbps	No	$6.669 hourly	$4.579 hourly	$9.613 hourly	$7.523 hourly
X1 32xlarge	x1.32xlarge	1952.0 GB	349 units	128 vCPUs	3840.0 GB (2 * 1920.0 GB SSD)	64-bit	20 Gigabit	10000.0 Mbps	No	$13.338 hourly	$9.158 hourly	$19.226 hourly	$15.046 hourly
R3 High-Memory Large	r3.large	15.25 GB	6.5 units	2 vCPUs	32.0 GB SSD	64-bit	Moderate	N/A	No	$0.166 hourly	$0.105 hourly	$0.291 hourly	$0.238 hourly
R3 High-Memory Extra Large	r3.xlarge	30.5 GB	13 units	4 vCPUs	80.0 GB SSD	64-bit	Moderate	500.0 Mbps	No	$0.333 hourly	$0.209 hourly	$0.583 hourly	$0.428 hourly
R3 High-Memory Double Extra Large	r3.2xlarge	61.0 GB	26 units	8 vCPUs	160.0 GB SSD	64-bit	High	1000.0 Mbps	No	$0.665 hourly	$0.418 hourly	$1.045 hourly	$0.824 hourly
R3 High-Memory Quadruple Extra Large	r3.4xlarge	122.0 GB	52 units	16 vCPUs	320.0 GB SSD	64-bit	High	2000.0 Mbps	No	$1.330 hourly	$0.836 hourly	$1.944 hourly	$1.490 hourly
R3 High-Memory Eight Extra Large	r3.8xlarge	244.0 GB	104 units	32 vCPUs	640.0 GB (2 * 320.0 GB SSD)	64-bit	10 Gigabit	N/A	No	$2.660 hourly	$1.672 hourly	$3.600 hourly	$1.989 hourly
I2 Extra Large	i2.xlarge	30.5 GB	14 units	4 vCPUs	800.0 GB SSD	64-bit	Moderate	500.0 Mbps	No	$0.853 hourly	$0.424 hourly	$0.973 hourly	$0.565 hourly
I2 Double Extra Large	i2.2xlarge	61.0 GB	27 units	8 vCPUs	1600.0 GB (2 * 800.0 GB SSD)	64-bit	High	1000.0 Mbps	No	$1.705 hourly	$0.848 hourly	$1.946 hourly	$1.131 hourly
I2 Quadruple Extra Large	i2.4xlarge	122.0 GB	53 units	16 vCPUs	3200.0 GB (4 * 800.0 GB SSD)	64-bit	High	2000.0 Mbps	No	$3.410 hourly	$1.696 hourly	$3.891 hourly	$2.260 hourly
I2 Eight Extra Large	i2.8xlarge	244.0 GB	104 units	32 vCPUs	6400.0 GB (8 * 800.0 GB SSD)	64-bit	10 Gigabit	N/A	No	$6.820 hourly	$3.392 hourly	$7.782 hourly	$4.521 hourly
D2 Extra Large	d2.xlarge	30.5 GB	14 units	4 vCPUs	6000.0 GB (3 * 2000.0 GB)	64-bit	Moderate	750.0 Mbps	No	$0.690 hourly	$0.402 hourly	$0.821 hourly	$0.472 hourly
D2 Double Extra Large	d2.2xlarge	61.0 GB	28 units	8 vCPUs	12000.0 GB (6 * 2000.0 GB)	64-bit	High	1000.0 Mbps	No	$1.380 hourly	$0.804 hourly	$1.601 hourly	$0.885 hourly
D2 Quadruple Extra Large	d2.4xlarge	122.0 GB	56 units	16 vCPUs	24000.0 GB (12 * 2000.0 GB)	64-bit	High	2000.0 Mbps	No	$2.760 hourly	$1.608 hourly	$3.052 hourly	$1.690 hourly
D2 Eight Extra Large	d2.8xlarge	244.0 GB	116 units	36 vCPUs	48000.0 GB (24 * 2000.0 GB)	64-bit	10 Gigabit	4000.0 Mbps	No	$5.520 hourly	$3.216 hourly	$6.198 hourly	$3.300 hourly
HI1. High I/O Quadruple Extra Large	hi1.4xlarge	60.5 GB	35 units	16 vCPUs	2048.0 GB (2 * 1024.0 GB SSD)	64-bit	10 Gigabit	N/A	No	$3.100 hourly	$1.698 hourly	$3.580 hourly	$2.260 hourly
High Storage Eight Extra Large	hs1.8xlarge	117.0 GB	35 units	16 vCPUs	48000.0 GB (24 * 2000.0 GB)	64-bit	10 Gigabit	N/A	No	$4.600 hourly	$2.574 hourly	$4.931 hourly	$2.961 hourly
M3 General Purpose Medium	m3.medium	3.75 GB	3 units	1 vCPUs	4.0 GB SSD	64-bit	Moderate	N/A	No	$0.067 hourly	$0.048 hourly	$0.130 hourly	$0.100 hourly
M3 General Purpose Large	m3.large	7.5 GB	6.5 units	2 vCPUs	32.0 GB SSD	64-bit	Moderate	N/A	No	$0.133 hourly	$0.090 hourly	$0.259 hourly	$0.199 hourly
M3 General Purpose Extra Large	m3.xlarge	15.0 GB	13 units	4 vCPUs	80.0 GB (2 * 40.0 GB SSD)	64-bit	High	500.0 Mbps	No	$0.266 hourly	$0.180 hourly	$0.518 hourly	$0.397 hourly
M3 General Purpose Double Extra Large	m3.2xlarge	30.0 GB	26 units	8 vCPUs	160.0 GB (2 * 80.0 GB SSD)	64-bit	High	1000.0 Mbps	No	$0.532 hourly	$0.360 hourly	$1.036 hourly	$0.793 hourly

Do choices matter?



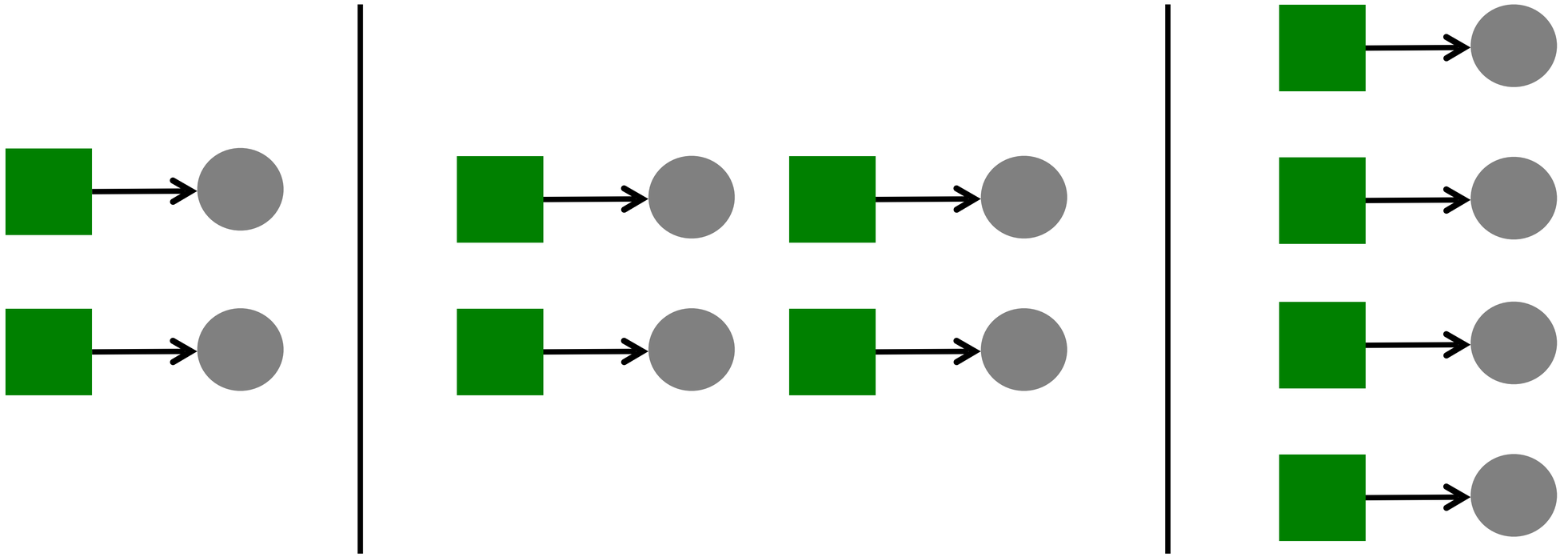
~2x performance difference for same price

Do choices MATTER ?



computation + communication yield non-linear scaling

Computation patterns

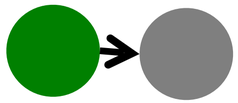


Time $\propto$ INPUT

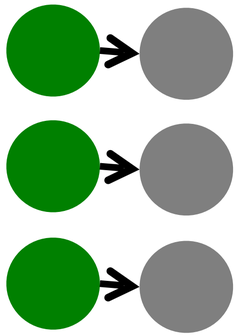
Time $\propto \frac{1}{\text{machines}}$

Communication Patterns

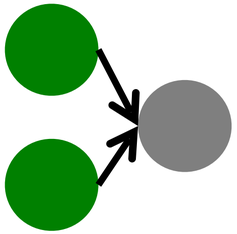
ONE-To-ONE



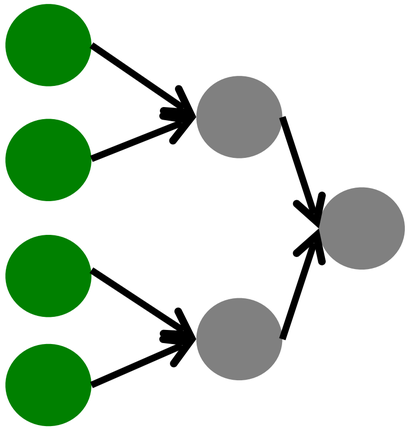
CONSTANT



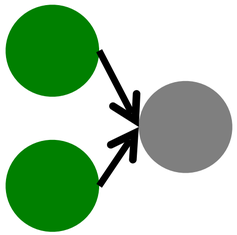
Tree DAG



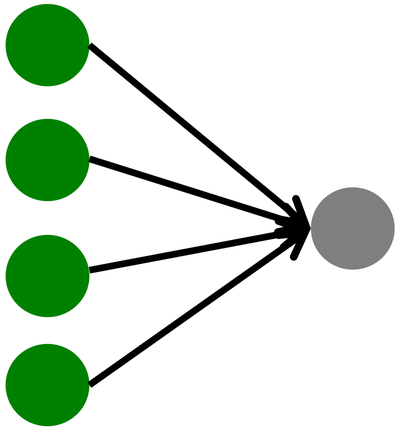
LOG



All-to-one



LINEAR



BASIC Model

Computation (linear)

All-to-One DAG

$$time = x_1 + x_2 * \frac{input}{machines} + x_3 * \log(machines) + x_4 * (machines)$$

Serial Execution

Tree DAG

Collect Training Data

Fit Linear Regression

Optimal Design of Experiments

Given a Linear Model

$$y_i = a_i^T x + w_i, \qquad i = 1, \ldots, m$$

Collection of (input, machines)

Associate cost with each experiment

$\boldsymbol{\lambda_i}$ - Fraction of times each experiment is run

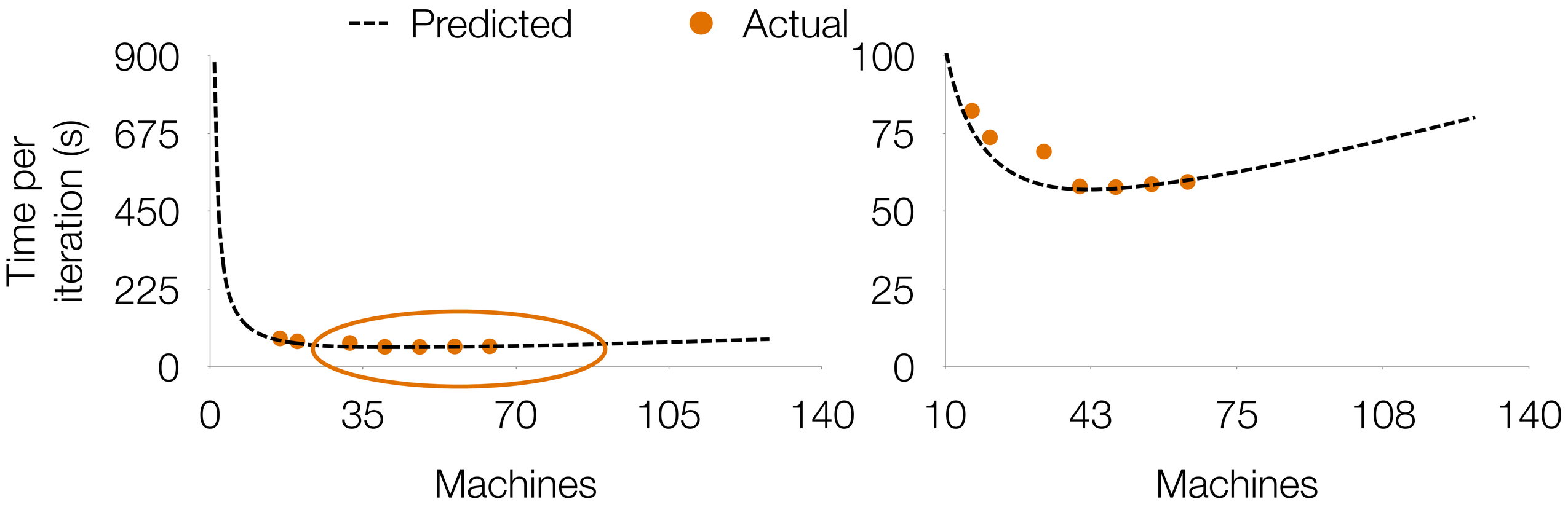
$$\begin{array}{ll} \text{minimize} & \mathbf{tr}\left\{\left(\sum_{i=1}^m \lambda_i a_i a_i^T\right)^{-1}\right\} \\ \text{subject to} & \lambda_i \in [0,1] \\ & \sum_{I=1}^m c_i \lambda_i \leq B \end{array}$$

Lower variance → Better model

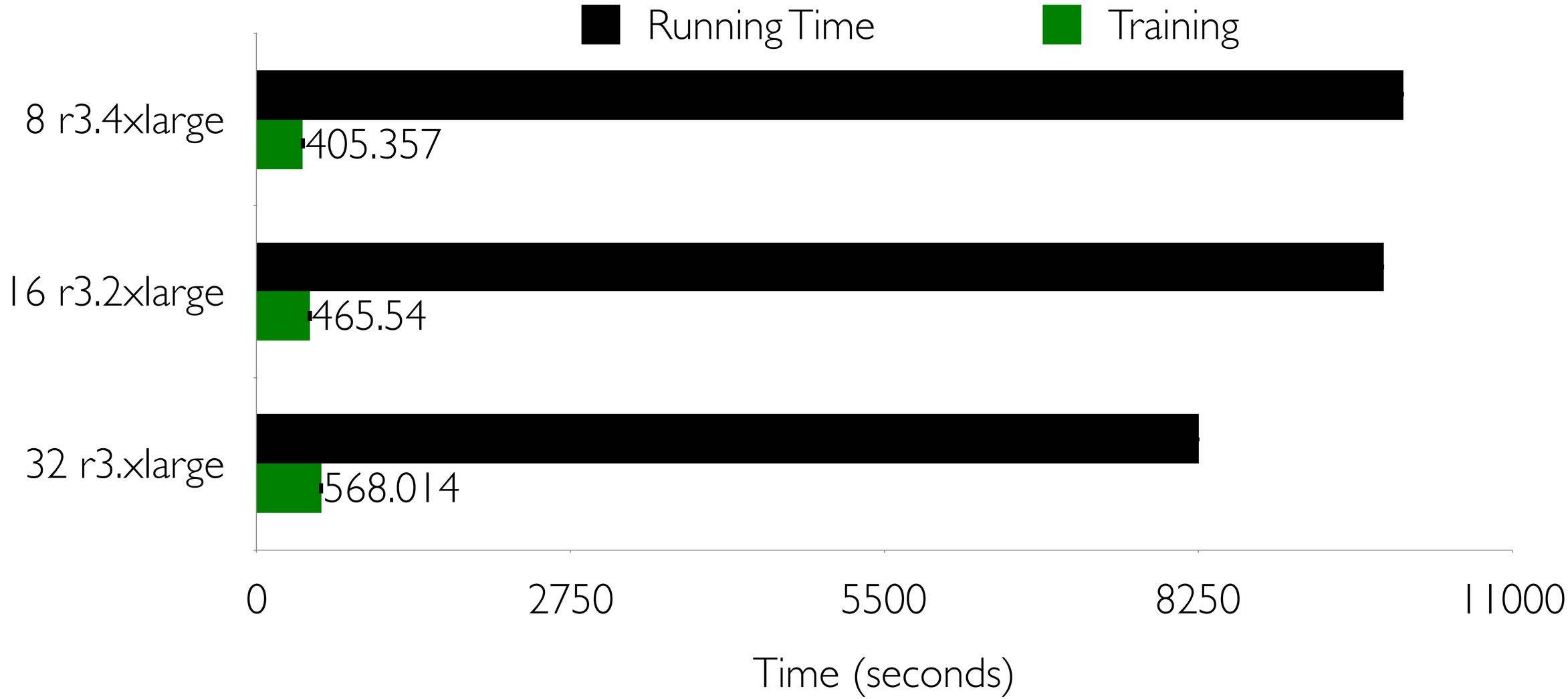
Bound total cost

Number of instances

Large Least-squares solve (TIMIT) on r3.xlarge instances



CHOOSING INSTANCE TYPES

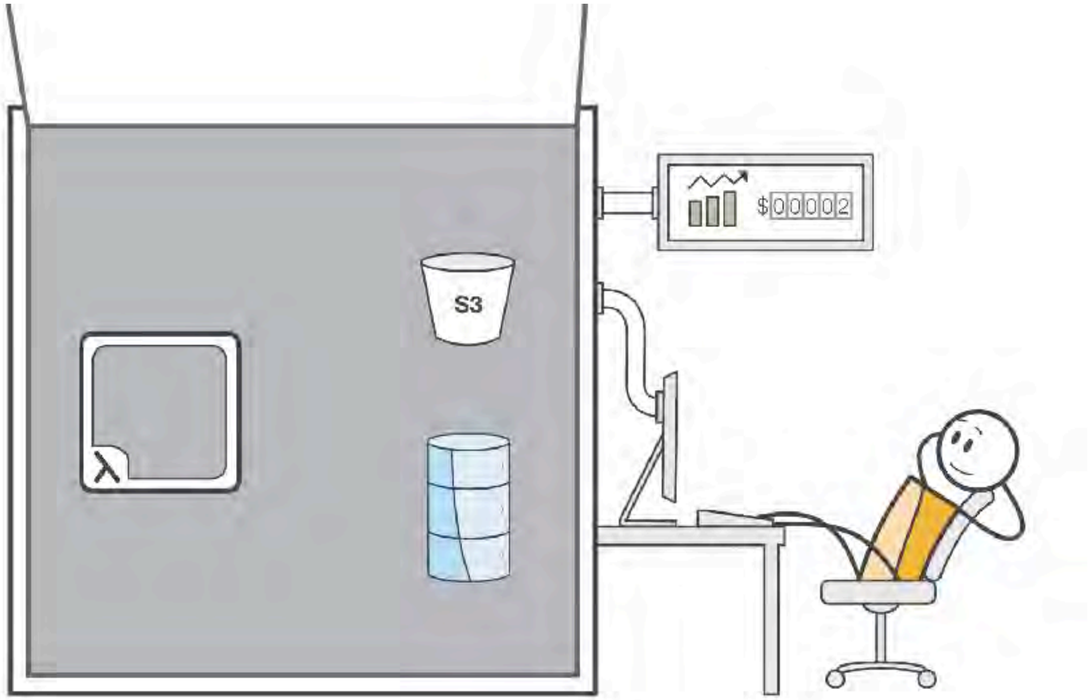


Wait. That still seems hard!

…and what happened to the RAM?

AWS LAMBDA

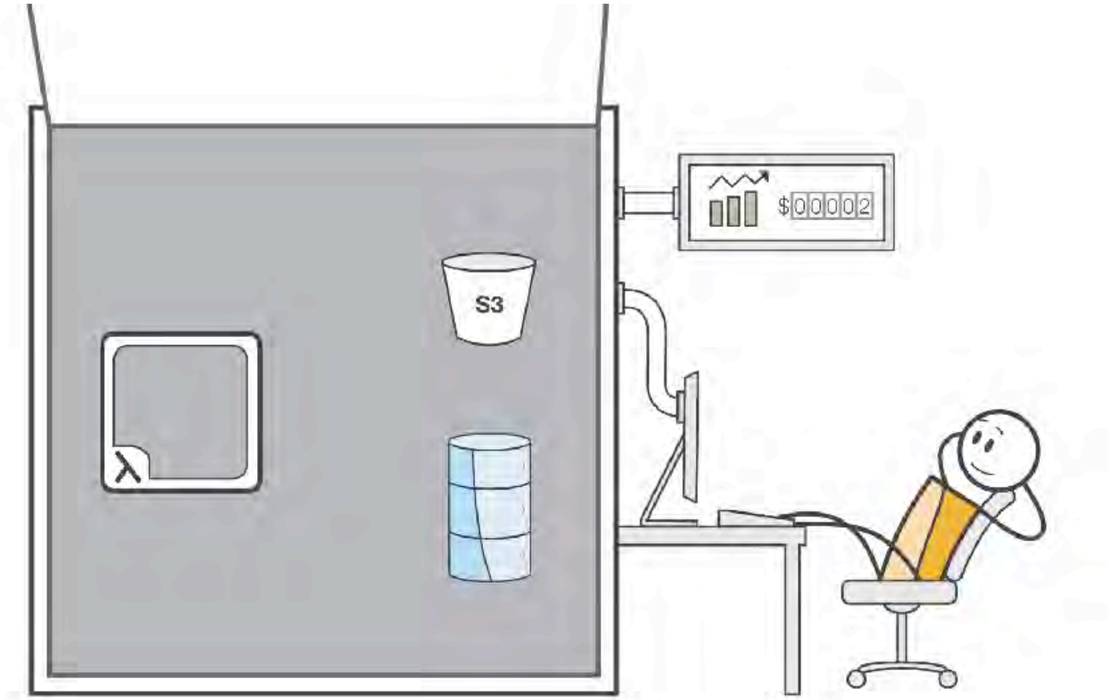
- 300 seconds single-core (AVX2)
- 512 MB in /tmp
- 1.5GB RAM
- Python, Java, Node



- S3 = Simple Storage Service. Essentially infinite RAM
- Communication at 600MB/s per machine (same speed as SATA)
- (PCI Bus is 1GB/s, Memory Bus is 80GB/s)

AWS LAMBDA

- 300 seconds single-core (AVX2)
- 512 MB in /tmp
- 1.5GB RAM
- Python, Java, Node



Google Cloud Platform

CLOUD FUNCTIONS ALPHA

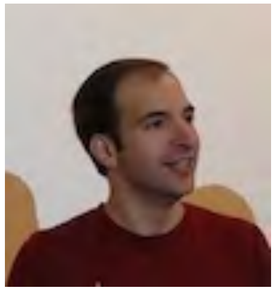
A serverless platform for building event-based microservices

Microsoft Azure

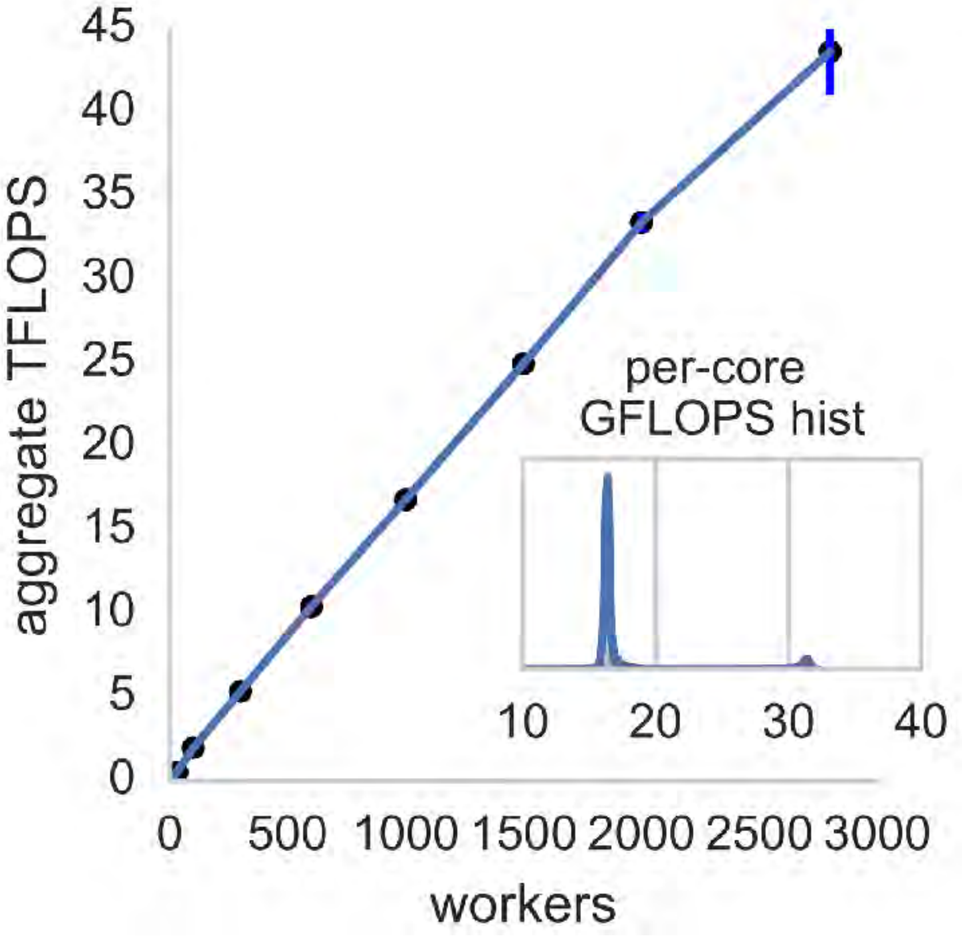
Azure Functions

Process events with a serverless code architecture

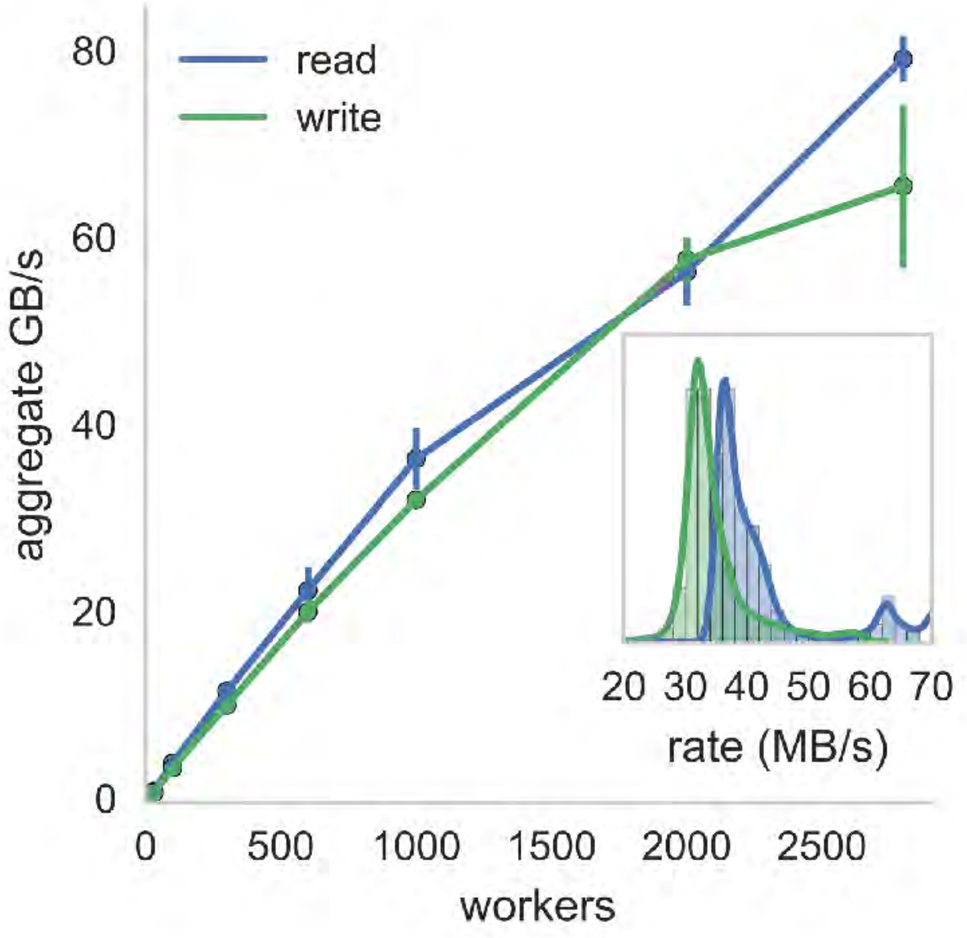
LAMBDA SCALABILITY

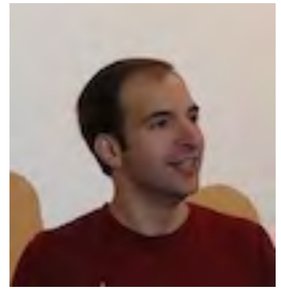


Compute



Data

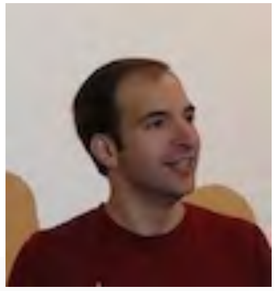




"Most wrens are small and rather inconspicuous, except for their loud and often complex songs."

pywren

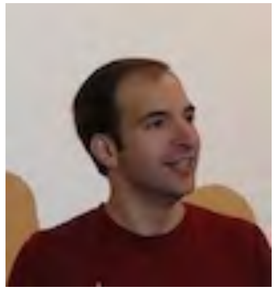
THE API



```
import pywren
import numpy as np

def addone(x):
    return x + 1

wrenexec = pywren.default_executor()
xlist = np.arange(10)
futures = wrenexec.map(addone, xlist)

print [f.result() for f in futures]
```

The output is as expected:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```



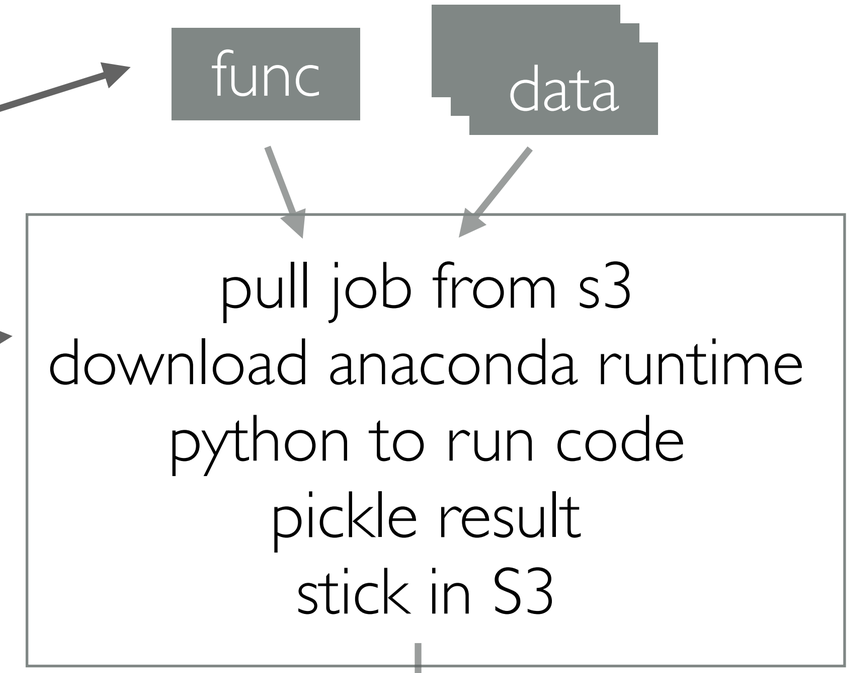
How it works

```
future = runner.map(fn, data)
```

Serialize func and data

Put on S3

Invoke Lambda




```
future.result()
```

poll S3

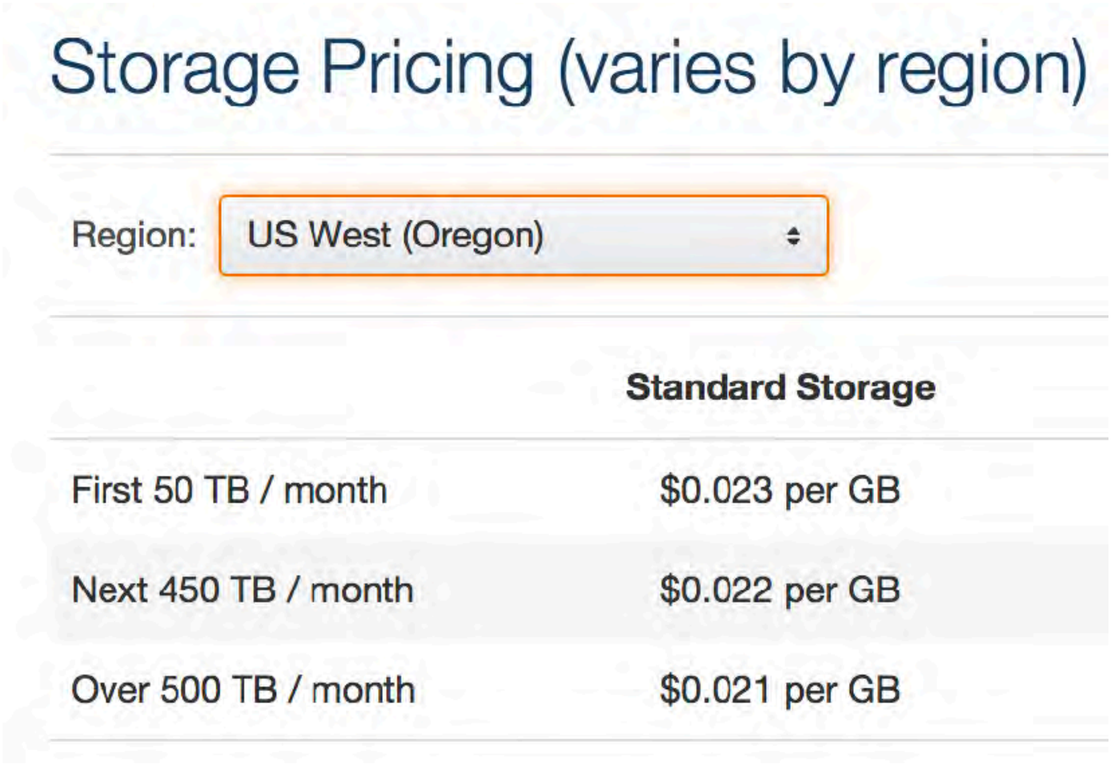
unpickle and return

your laptop

the cloud

How expensive is **S3**?

(Taking dimensionality analysis seriously, or "beyond PSPACE")

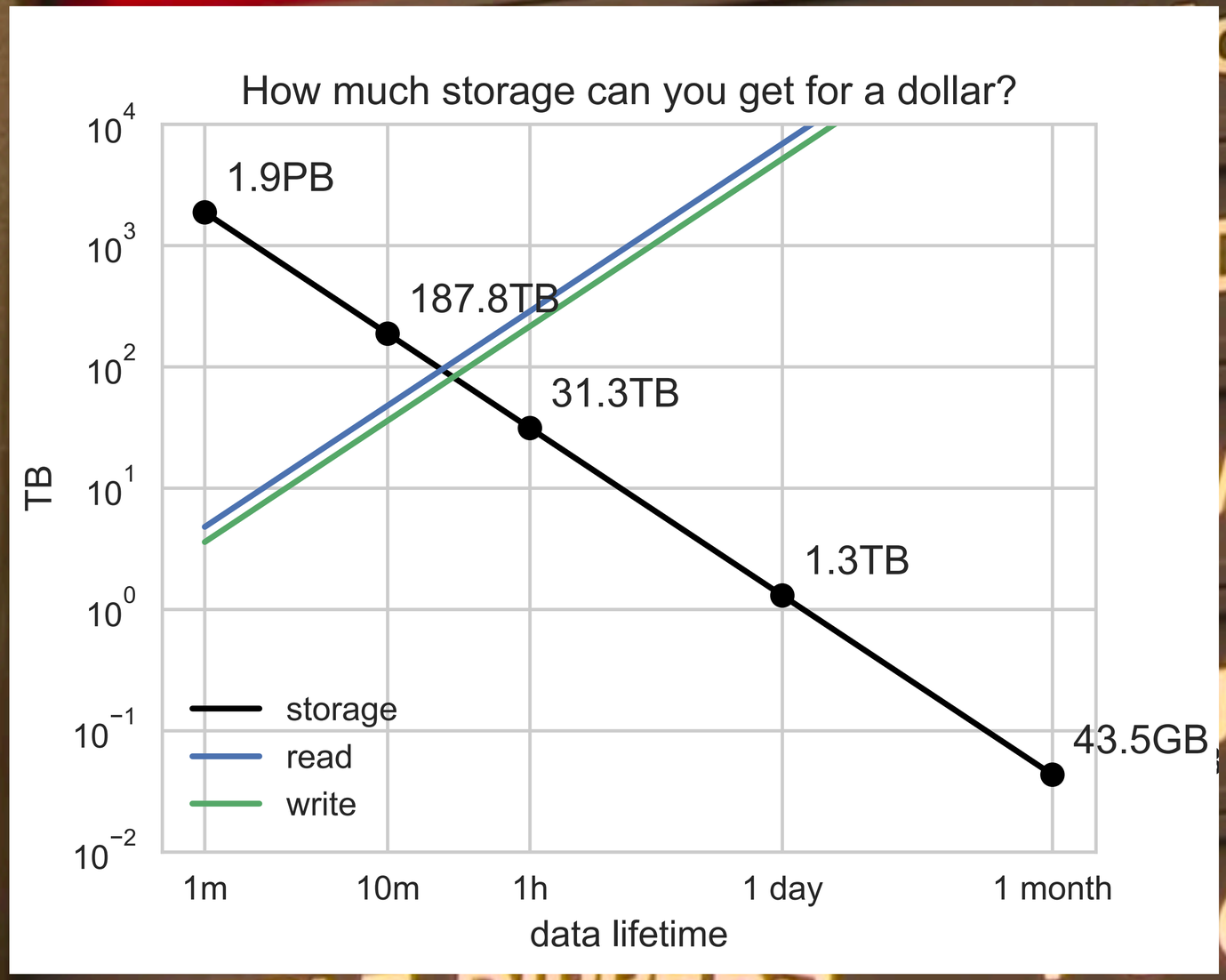


Storage Pricing (varies by region)

Region: US West (Oregon)

	Standard Storage
First 50 TB / month	$0.023 per GB
Next 450 TB / month	$0.022 per GB
Over 500 TB / month	$0.021 per GB

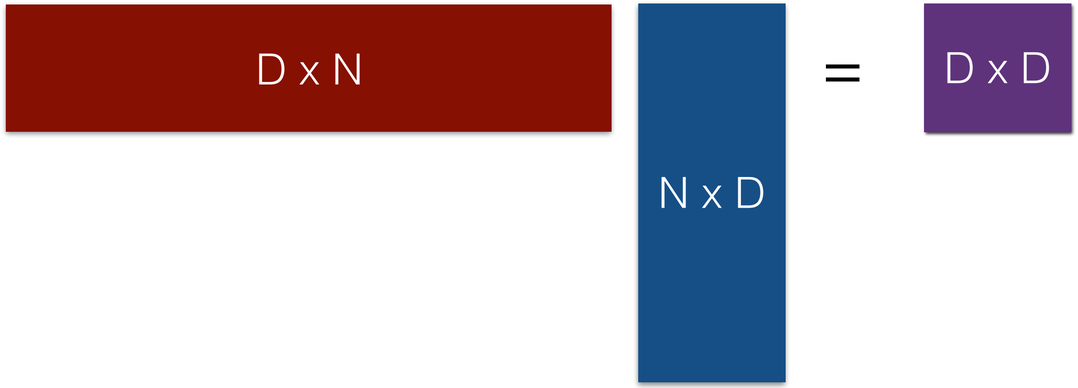
- Simple Storage Service
- Essentially infinite RAM
- Communication at 600MB/s per machine
- Same speed as SATA
- (PCI Bus is 1GB/s, Memory Bus is 80GB/s)



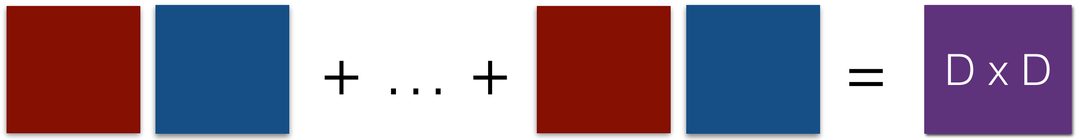
- How do algorithms change when you have infinite memory (through a straw)
- Never discard intermediate information

numpywren

- That's a lot of SIMD cores!
- Parallel matrix multiplication is easy when output matrix is small



- Fits cleanly into map-reduce framework

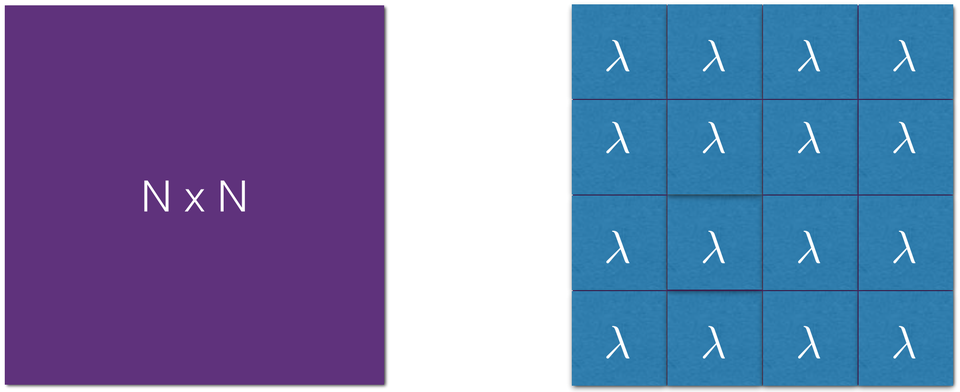


numpywren

- However when output matrix is very large it becomes very difficult or expensive to store in memory

- For example for N = 1e6 and D=1e4
  - D x D matrix of doubles is 800 Mb
  - N x N matrix of doubles is 8 TB
- Storing 8 TB in memory traditional cluster is expensive!

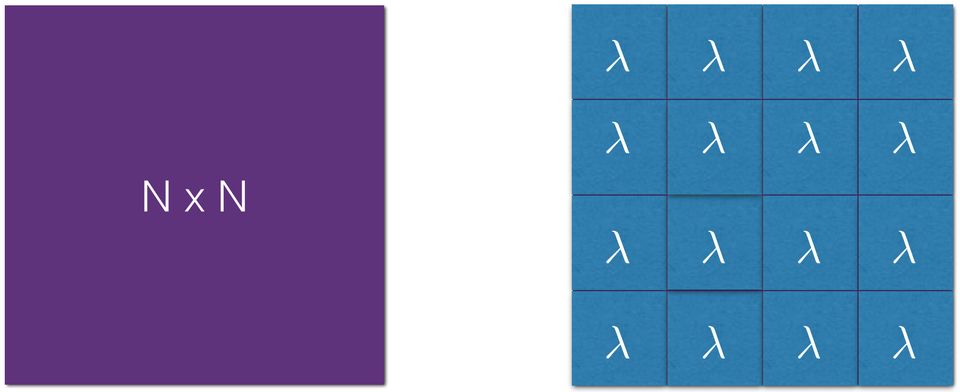
numpywren



- Solution: Use S3 to store matrices, stream blocks to Lambdas to compute output matrix in parallel

N	D	Lambdas	Runtime	Output Size
50000	784	225	192s	20 GB
50000	18432	225	271s	20 GB
1.2 Millon	4096	3000	1320s	11 TB
1.2 Million	18432	3000	2520s	11 TB

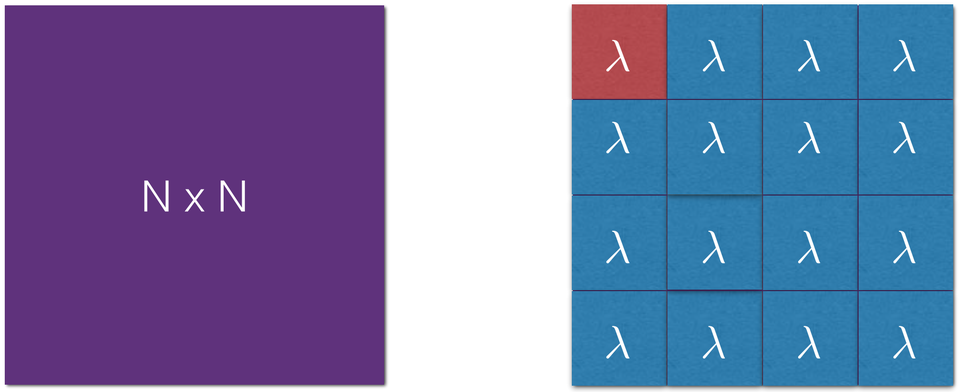
Cholesky in numpywren



$$\begin{bmatrix} A_{11} & A_{12}^T \\ A_{12} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{12} & L_{22} \end{bmatrix} \begin{bmatrix} L_{11}^T & L_{12}^T \\ 0 & L_{22}^T \end{bmatrix}$$

- Solution: Use S3 to store matrices, stream blocks to Lambdas to compute output matrix in parallel

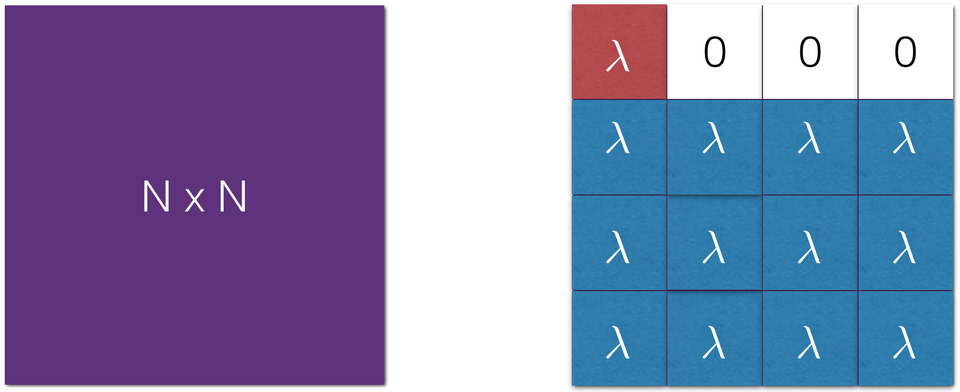
Cholesky in numpywren



$$\begin{bmatrix} A_{11} & A_{12}^T \\ A_{12} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{12} & L_{22} \end{bmatrix} \begin{bmatrix} L_{11}^T & L_{12}^T \\ 0 & L_{22}^T \end{bmatrix}$$

- Solution: Use S3 to store matrices, stream blocks to Lambdas to compute output matrix in parallel

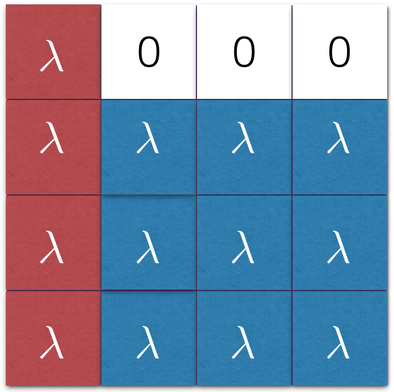
Cholesky in numpywren



$$\begin{bmatrix} A_{11} & A_{12}^T \\ A_{12} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{12} & L_{22} \end{bmatrix} \begin{bmatrix} L_{11}^T & L_{12}^T \\ 0 & L_{22}^T \end{bmatrix}$$

- Solution: Use S3 to store matrices, stream blocks to Lambdas to compute output matrix in parallel

Cholesky in numpywren

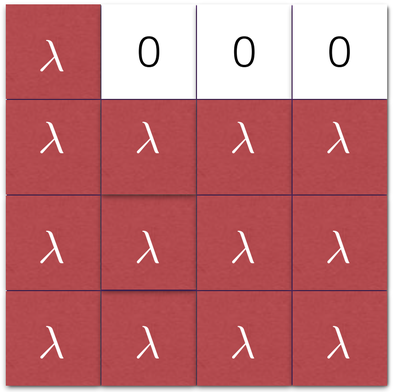


$$\begin{bmatrix} A_{11} & A_{12}^T \\ A_{12} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{12} & L_{22} \end{bmatrix} \begin{bmatrix} L_{11}^T & L_{12}^T \\ 0 & L_{22}^T \end{bmatrix}$$

$$L_{21} = A_{21} L_{11}^{-T}$$

- Solution: Use S3 to store matrices, stream blocks to Lambdas to compute output matrix in parallel

Cholesky in numpywren



$$\begin{bmatrix} A_{11} & A_{12}^T \\ A_{12} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{12} & L_{22} \end{bmatrix} \begin{bmatrix} L_{11}^T & L_{12}^T \\ 0 & L_{22}^T \end{bmatrix}$$

$$L_{21} = A_{21} L_{11}^{-T}$$

$$A_{22}^{(\text{new})} = A_{22} - L_{12} L_{12}^T$$

- Solution: Use S3 to store matrices, stream blocks to Lambdas to compute output matrix in parallel

The future is direct solves

Fix a positive definite, n x n matrix A

Find x such that $Ax = b$

$$\text{minimize} \quad \frac{1}{2}x^T Ax - b^T x$$

- Iterative methods are not fast
- New substrates make large-scale direct solve reachable
- Need simpler tools and cost management
- There are algorithmic challenges in noniterative methods

References

- If you'd like to try Pywren visit pywren.io 

- "Occupy the Cloud: Distributed Computing for the 99%." Eric Jonas, Qifan Pu, Shivaram Venkataraman, Ion Stoica, and Benjamin Recht. In *Proceedings of the ACM Symposium on Cloud Computing (SOCC)*. 2017.
- "Breaking Locality Accelerates Block Gauss-Seidel." Stephen Tu, Shivaram Venkataraman, Ashia C. Wilson, Alex Gittens, Michael I. Jordan, and Benjamin Recht. In *Proceedings of the International Conference on Machine Learning (ICML).* 2017.
- "Ernest: Efficient Performance Prediction for Large Scale Advanced Analytics." Shivaram Venkataraman, Zongheng Yang, Michael J Franklin, Benjamin Recht, and Ion Stoica. In *Proceedings of the Symposium on Networked Systems Design (NSDI).* 2016.

`https://people.eecs.berkeley.edu/~brecht/publications.html`