Off the beaten path

Sponsored by TIBCO

TOC

Welcome
Overview
Problems to solve
Case example
Algorithms
Parallel Tasks
Google's Page Index
Demo
TPU

Overview

Welcome to the first ‘Off the beaten path sponsored by TIBCO’ meeting

Who we are- Technology enthusiasts that are interested in seeing solutions to things that might just be ***off the beaten path***

Topics will range wildly….

Logistics

700-730 Snacks and light fare

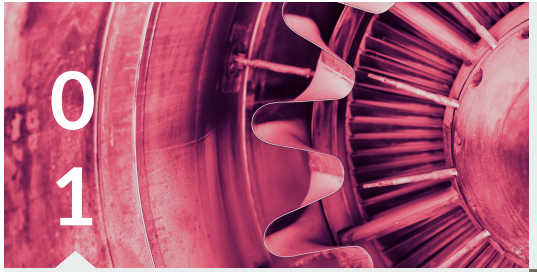
730-830 Meeting discussion

830-900 Move up stairs - networking - social til close

Problems to solve

1. What is matrix multiplication?
2. Problems solved with matrix multiplication
3. What happens when the matrices get too big?
4. A short demo

Uses of Matrix Multiplication



Physics - Equations of Motion

Track and predict the motion of an object-
position, speed, velocity
Model electrical circuits

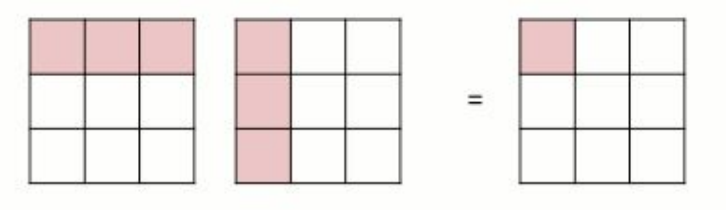
Graphics - Gaming

Gaming objects and FPV are rotated and translated in 2D and 3D space

Machine Learning

Inputs to ML problems described as columns of data.
Power recommendation engines and facial recognition

Matrix Multiplication



Definition [edit]

If $\mathbf{A}$ is an $n \times m$ matrix and $\mathbf{B}$ is an $m \times p$ matrix,

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mp} \end{pmatrix}$$

the *matrix product* $\mathbf{C} = \mathbf{AB}$ (denoted without multiplication signs or dots) is defined to be the $n \times p$ matrix

$$\mathbf{C} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{np} \end{pmatrix}$$

such that

$$c_{ij} = a_{i1}b_{1j} + \cdots + a_{im}b_{mj} = \sum_{k=1}^{m} a_{ik}b_{kj},$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, p$.

The curse of dimensionality

phenomena that arises when analyzing and organizing data in high-dimensional spaces (often with hundreds or thousands of dimensions) that do not occur in low-dimensional settings such as the three-dimensional physical space of everyday experience

The curse of dimensionality

the volume of the space increases so fast that the available data become sparse. This sparsity is problematic for any method that requires statistical significance. In order to obtain a statistically sound and reliable result, the amount of data needed to support the result often grows exponentially with the dimensionality.

Naively, each additional dimension doubles the effort needed

Case example

AB where A and B are both 10 x 10 matrices, i.e. n=10

O(n^3) = 1000 operations or 1e3

Modern commodity CPU can do 40 GFLOPS (giga floating point operations per second, 10^9)

Cool…problem solved in .025 μs

Case example

AB where A and B are both 10,000 x 10,000 matrices, i.e. n=10,000, 1e4

O(n^3) = 10^12 operations, 1 trillion operations, 1e12

Modern commodity CPU can do 40 GFLOPS (giga floating point operations per second, 10^9)

OK…problem solved in 25 s

Case example

AB where A and B are both 100,000 x 100,000 matrices, i.e. n=100,000, 1e5

O(n^3) = 10^15 operations, 1 quadrillion operations, 1e15

Modern commodity CPU can do 40 GFLOPS (giga floating point operations per second, 10^9)

Uh oh…problem solved in 25,000 seconds, or 416 minutes, or almost 7 hours

Case example

Is it real to have matrices of size 100,000 or greater?

Yes, in fact, this is a very small figure compared to some of the largest matrix computations…

Search engines index - page ranking:

2002 - 2.7 billion

2008 - 1 trillion

2016 - 60 trillion

Case example

Specialized hardware - cram many processors onto a single device - GPUs

Nvidia TESLA V100 - 7 double-precision TFLOPS (2560 cores)

100,000 x 100,000 multiplication now only takes 142 s with a GPU

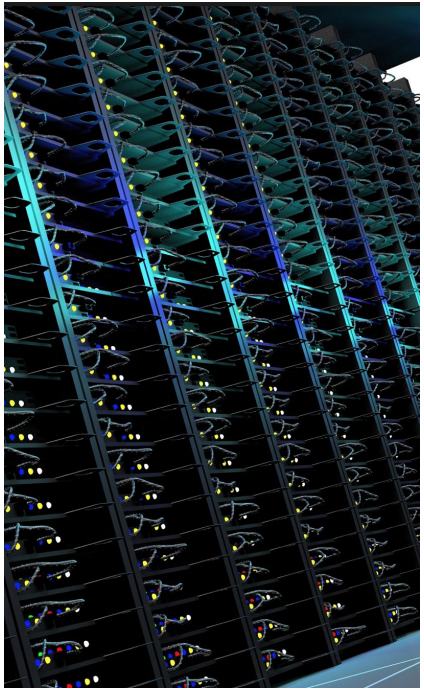
7 hours (commodity) vs 2 minutes (GPU)

And if we used the GPU with the 2.7 billion web index from 2002…
...it would only take 10 million years

Algorithms and Parallelization

Optimized algorithms - reduction in multiplications and total number of operations

Individual calculations can be worked on independently and combined later on - parallelization



‘New’ Math (Common Core) vs Old Math

35 x 12 = ?

x	30	5
10	300	50
2	60	10

```
    300
     50
     60
+    10
--------
    420
```

2 factors
4 multiplications
7 additions

```
     35
   x 12
--------
     70
+   350
--------
    420
```

4 multiplications
3 additions

Matrix Multiplication Algorithms

Algorithm	Omega	Note
Francois Le Gall	2.3728639	(1)
Virginia Vassilevska Williams	2.3728642	
Stothers	2.3740000	
Coppersmith-Winograd	2.3760000	
Bini	2.7790000	
Pan	2.7950000	
Strassen	2.8070000	(2)
Strassen-Winograd	2.8070000	
Classic	3.0000000	(3)

Omega - a measure of efficiency for matrix multiplication

$O(n^{\omega})$

Strassen Algorithm

Important - not because it is the fastest....
...because it kicked off much more research into the study of matrix multiplication algorithms that led to faster algorithms

Volker Strassen recognized that the n^3 MMA was not optimal…
...basically showed that the matrix multiplication could be done in 7 subblock multiplications vs 8 operations.

https://en.wikipedia.org/wiki/Strassen_algorithm

So what can we do today? Scale dramatically

TIBCO DataSynapse GridServer

ABOUT HELP ADMIN

You are logged in as admin | Last login Feb 12, 2019 8:30:47 AM

Broker Name: 638500212 | Hostname: 169.254.74.52 | Director: primary | Broker: enabled

Dashboard | Services | Engines | Users | Brokers | Batch | Admin | Diagnostics

Engines 4 | Drivers 2 | Active Sessions 0 | Running Tasks 0 | Queued Tasks 0

Service Session Admin

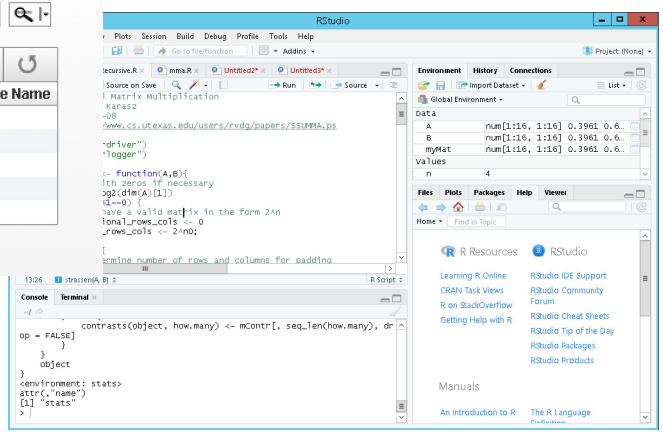
Page Help

Find In Finish Time Search

Global Actions Page: 5 of 5

#	Actions	Service Session ID	Session Name	Status	Submit Time	Finish Time	Username	Hostname	Times	Tasks	Priority	Service Type Name
81		1599990974117293816	R Matrix Multiplication	Finished	2/12/19 10:52 AM	2/12/19 10:52 AM	admin	mft72demo	0 / 0	0 / 0 / 7 / 7	5	Rmma
82		1599990972953719688	R Matrix Multiplication	Finished	2/12/19 10:52 AM	2/12/19 10:52 AM	admin	mft72demo	0 / 0	0 / 0 / 7 / 7	5	Rmma
83		1599990975403231451	R Matrix Multiplication	Finished	2/12/19 10:52 AM	2/12/19 10:52 AM	admin	mft72demo	0 / 0	0 / 0 / 7 / 7	5	Rmma
84		1599990973018091032	R Matrix Multiplication	Finished	2/12/19 10:52 AM	2/12/19 10:52 AM	admin	mft72demo	0 / 0	0 / 0 / 7 / 7	5	Rmma
85		1599990975644079016	R Matrix Multiplication	Finished	2/12/19 10:52 AM	2/12/19 10:52 AM	admin	mft72demo	0 / 0	0 / 0 / 7 / 7	5	Rmma

Results Per Page: 20 Page: 5 of 5



Revisiting Google's 2.7 Billion Page Index

Using a 1.3 million vCPU Grid ≅1,000 teraFLOPS or 1 peta FLOPS(1e15)

624,000 years using classic algorithm

944 years using Strassen

Or 9 months using fastest algo

AWS News Blog

Creating a 1.3 Million vCPU Grid on AWS using EC2 Spot Instances and TIBCO GridServer

by Jeff Barr | on 09 MAY 2018 | in Amazon EC2, News | Permalink | Share

Many of my colleagues are fortunate to be able to spend a good part of their day sitting down with and listening to our customers, doing their best to understand ways that we can better meet their business and technology needs. This information is treated with extreme care and is used to drive the roadmap for new services and new features.

AWS customers in the financial services industry (often abbreviated as FSI) are looking ahead to the Fundamental Review of Trading Book (FRTB) regulations that will come in to effect between 2019 and 2021. Among other things, these regulations mandate a new approach to the "value at risk" calculations that each financial institution must perform in the four hour time window after trading ends in New York and begins in Tokyo. Today, our customers report this mission-critical calculation consumes on the order of 200,000 vCPUs, growing to between 400K and 800K vCPUs in order to meet the FRTB regulations. While there's still some debate about the magnitude and frequency with which they'll need to run this expanded calculation, the overall direction is clear.

Building a Big Grid

In order to make sure that we are ready to help our FSI customers meet these new regulations, we worked with TIBCO to set up and run a proof of concept grid in the AWS Cloud. The periodic nature of the calculation, along with the amount of processing power and storage needed to run it to completion within four hours, make it a great fit for an environment where a vast amount of cost-effective compute power is available on an on-demand basis.

https://aws.amazon.com/blogs/aws/creating-a-1-3-million-vcpu-grid-on-aws-using-ec2-spot-instances-and-tibco-gridserver/

So what lies ahead?

More adoption of TPUs - tensor processing unit -

https://en.wikipedia.org/wiki/Tensor_processing_unit

Google Cloud Compute Platform - Provides 11.4 PFLOPS

Thank you.