Distributed Computation of π with Apache Hadoop

Tsz-Wo Sze

Yahoo! Cloud Computing
Apache Hadoop PMC Member

Mapred'2010
Dec 1

Agenda

- Introduction
- A New World Record
- How to Compute The n^{th} Bits of π?
- Computing π with Hadoop

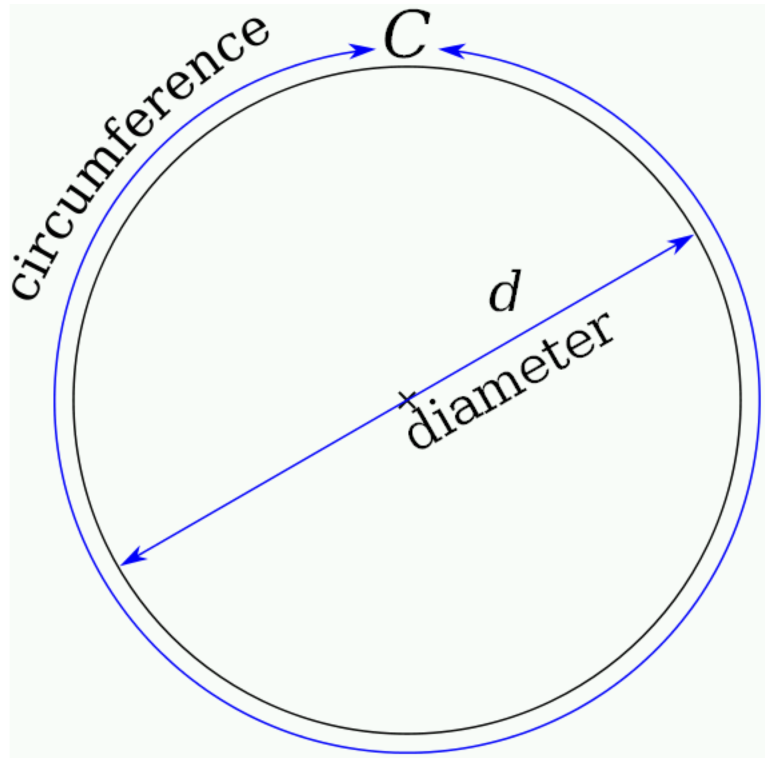
Agenda

- **Introduction**
- A New World Record
- How to Compute The n^{th} Bits of π?
- Computing π with Hadoop

What is π?

- π is a mathematical constant such that, for any circle,

$$\pi = \frac{\text{circumference}}{\text{diameter}} = \frac{C}{d}.$$

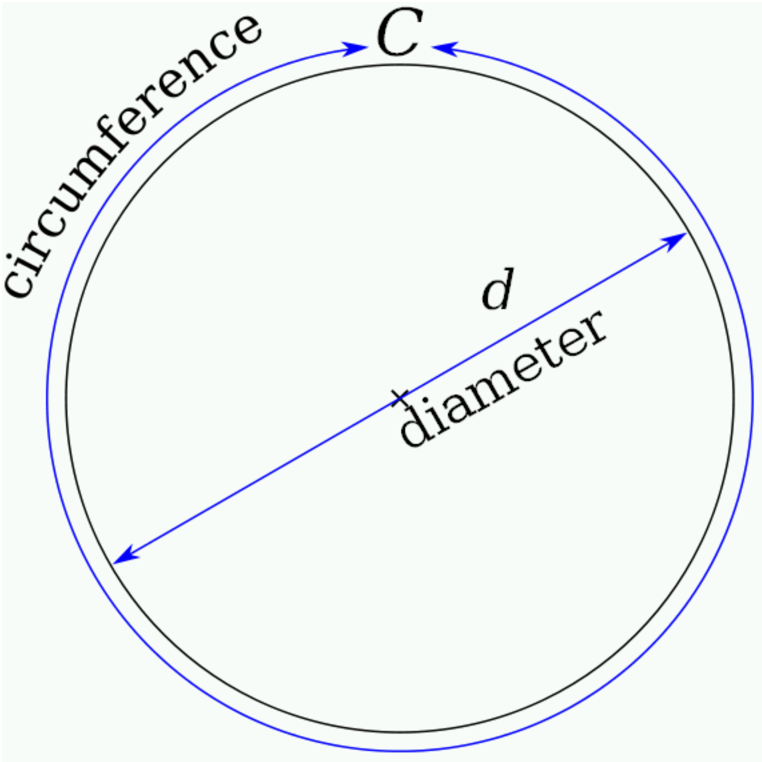


What is π?

- π is a mathematical constant such that, for any circle,

$$\pi = \frac{\text{circumference}}{\text{diameter}} = \frac{C}{d}.$$

- We have $\pi = 3.244$

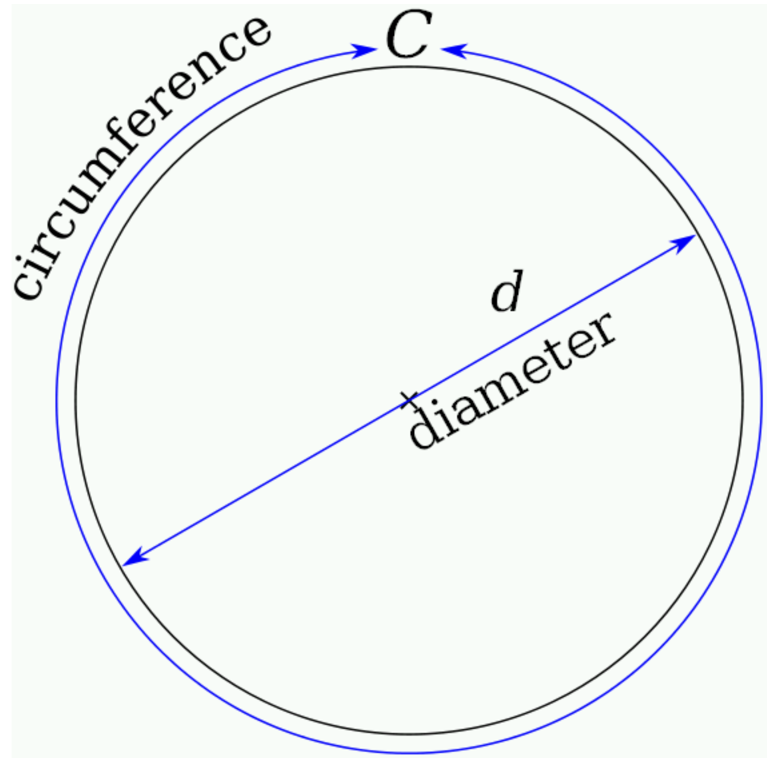


What is π?

- π is a mathematical constant such that, for any circle,

$$\pi = \frac{\text{circumference}}{\text{diameter}} = \frac{C}{d}.$$

- We have $\pi = 3.244$ (in **hexadecimal** ☺)



Decimal, Hexadecimal & Binary

- Representing π in different bases

$$
\begin{aligned}
\pi &= \texttt{3.1415926535 8979323846 2643383279} \ldots \\
&= \texttt{3.24}\underline{\texttt{3F}}\texttt{6A88 85A308D3 13198A2E} \ldots \\
&= \texttt{11.00100100 } \underline{\texttt{00111111}} \texttt{ 01101010} \ldots
\end{aligned}
$$

- **Bit position** is counted after the **radix point**.
- e.g., the eight bits starting at the ninth bit position are `00111111` in binary or `3F` in hexadecimal.

Two Types of Challenges

- Computing the first n decimal digits of π

$$\pi = 3.\underbrace{1415926535\ 8979323846\ 2643383279\ldots}_{n}$$

- Computing only the n^{th} bits of π

$$\pi = 11.00100100\ 00111111\ \underbrace{\overset{\overset{n}{\downarrow}}{0}1101010}_{\text{precision}}\ 10001000\ldots$$

We will focus on the second challenge in this talk.

Previous Results

- Fabrice Bellard (1997)
  - *Farthest bit position*: 1,000,000,000,151 $(= 10^{12} + 151)$
  - *Precision*: 152 bits
  - *Machines*: 20 workstations
  - *Duration*: 12 days
  - *CPU time*: 220 days
  - *Verification*: 180 days CPU time

Previous Results ’

- PiHex (2000)
  - *Farthest bit position*: 1,000,000,000,000,060 $(= 10^{15} + 60)$
  - *Precision*: 64 bits
  - *Machines*: Idle slices of 1734 machines
    An ‘average’ computer has a 450 MHz CPU
  - *Duration*: **736 days (>2 years)**
  - *CPU time*: 137 years
  - *Verification*: ???

It is not clear if they have verified their results.

Agenda

- Introduction
- **A New World Record**
- How to Compute The n^{th} Bits of π?
- Computing π with Hadoop

A New World Record

- Bit values (in hexadecimal)

  `0E6C1294 AED40403 F56D2D76 4026265B`

  `CA98511D 0FCFFAA1 0F4D28B1 BB5392B8`

A New World Record ’

- Bit values (in hexadecimal)

  `0E6C1294 AED40403 F56D2D76 4026265B`

  `CA98511D 0FCFFAA1 0F4D28B1 BB5392B8`

  (256 bits)

- ★ The first bit position: 1,999,999,999,999,997 ($= 2 \cdot 10^{15} - 3$)
- ★ The last bit position: 2,000,000,000,000,252 ($= 2 \cdot 10^{15} + 252$)
- ★ The two quadrillionth ($2 \cdot 10^{15}$th) bit is 0.

A New World Record ”

- Yahoo! Cloud Computing (July 2010)
  - *Farthest bit position*: 2,000,000,000,000,252
  - *Precision*: 256 bits
  - *Machines*: Idle slices of 1000-node clusters

    Each node has two quad-core 1.8-2.5 GHz CPUs
  - *Duration*: 23 days
  - *CPU time*: 503 years
  - *Verification*: 582 years CPU time

Comparing with PiHex

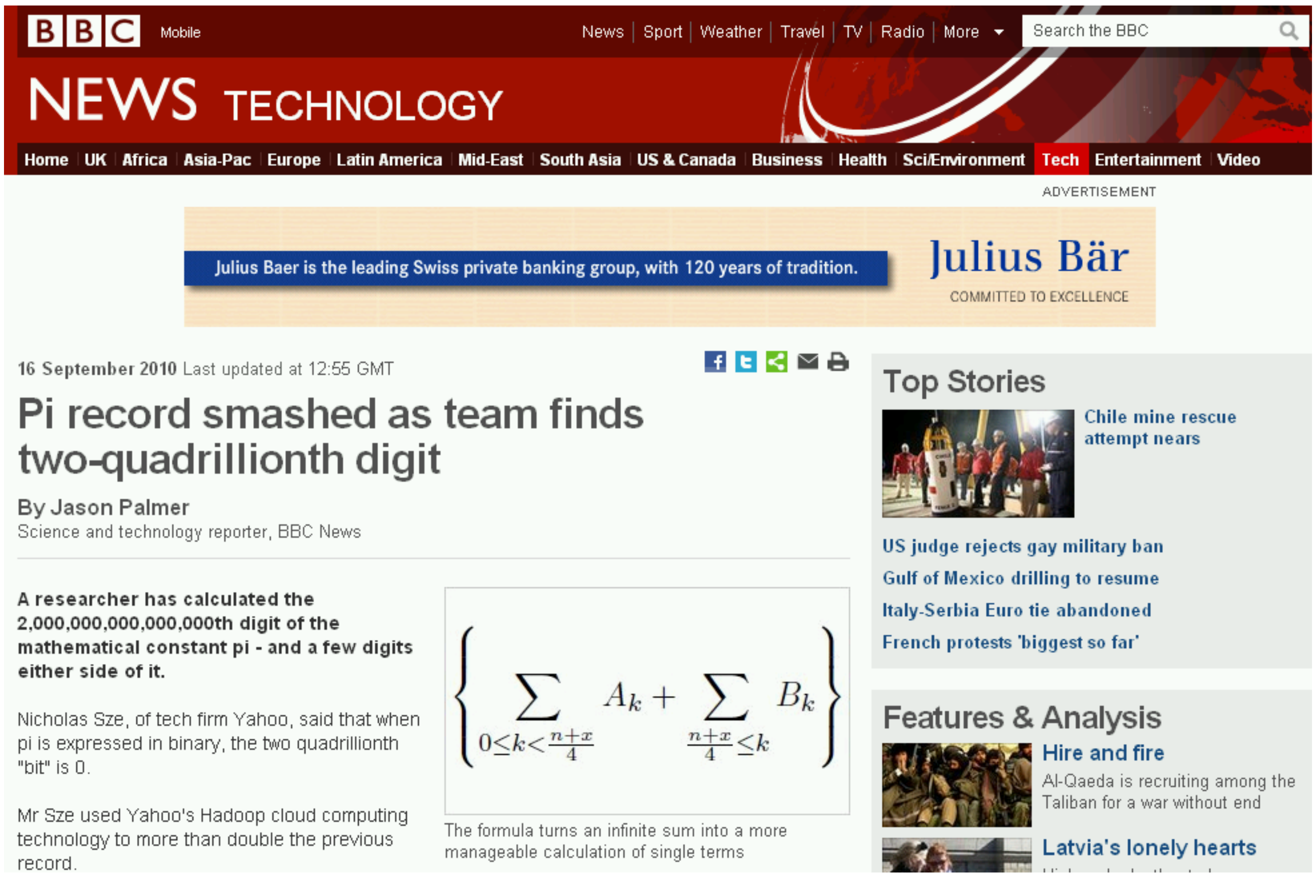
	PiHex	Our Computations	Ratio
Position:	around 10^{15}	around $2 \cdot 10^{15}$	1:2
Precision:	64 bits	256 bits	1:4
Duration:	736 days	23 days	32:1

Note that our hardware is 10 years more advanced than the ones used by PiHex.

BBC News (16 Sep 2010)

- Pi record smashed as team finds two-quadrillionth digit

http://www.bbc.co.uk/news/technology-11313194



BBC Mobile

News | Sport | Weather | Travel | TV | Radio | More

Search the BBC

NEWS TECHNOLOGY

Home | UK | Africa | Asia-Pac | Europe | Latin America | Mid-East | South Asia | US & Canada | Business | Health | Sci/Environment | Tech | Entertainment | Video

16 September 2010 Last updated at 12:55 GMT

Pi record smashed as team finds two-quadrillionth digit

By Jason Palmer

Science and technology reporter, BBC News

A researcher has calculated the 2,000,000,000,000,000th digit of the mathematical constant pi - and a few digits either side of it.

$$\left\{ \sum_{0 \le k < \frac{n+x}{4}} A_k + \sum_{\frac{n+x}{4} \le k} B_k \right\}$$

The formula turns an infinite sum into a more manageable calculation of single terms

Nicholas Sze, of tech firm Yahoo, said that when pi is expressed in binary, the two quadrillionth "bit" is 0.

Mr Sze used Yahoo's Hadoop cloud computing technology to more than double the previous record.

Top Stories

Chile mine rescue attempt nears

US judge rejects gay military ban

Gulf of Mexico drilling to resume

Italy-Serbia Euro tie abandoned

French protests 'biggest so far'

Features & Analysis

Hire and fire

Al-Qaeda is recruiting among the Taliban for a war without end

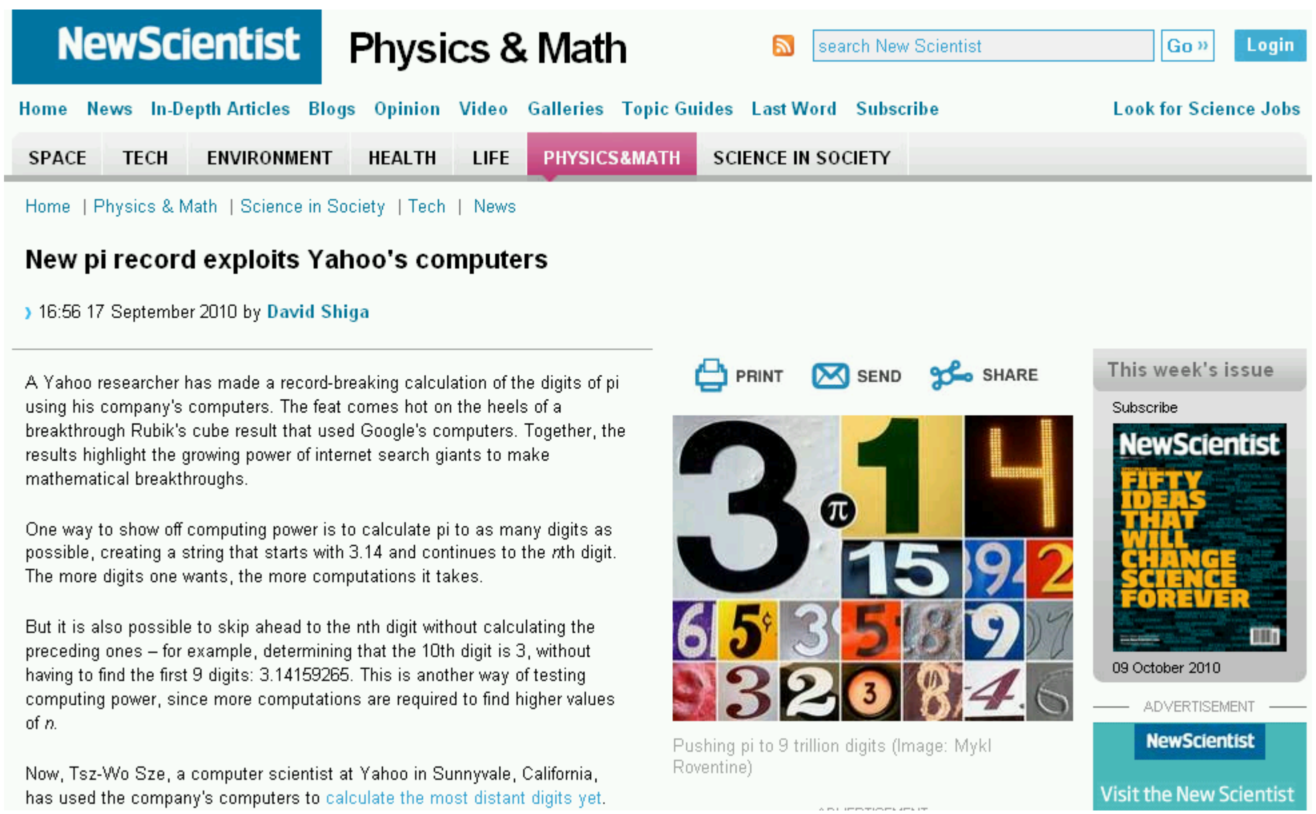
Latvia's lonely hearts

NewScientist (17 Sep 2010)

- New pi record exploits Yahoo's computers

  http://www.newscientist.com/article/dn19465-new-pi-record-exploits-yahoos-co

  html



NewScientist Physics & Math

search New Scientist Go » Login

Home News In-Depth Articles Blogs Opinion Video Galleries Topic Guides Last Word Subscribe Look for Science Jobs

SPACE TECH ENVIRONMENT HEALTH LIFE PHYSICS&MATH SCIENCE IN SOCIETY

Home | Physics & Math | Science in Society | Tech | News

New pi record exploits Yahoo's computers

› 16:56 17 September 2010 by David Shiga

A Yahoo researcher has made a record-breaking calculation of the digits of pi using his company's computers. The feat comes hot on the heels of a breakthrough Rubik's cube result that used Google's computers. Together, the results highlight the growing power of internet search giants to make mathematical breakthroughs.

One way to show off computing power is to calculate pi to as many digits as possible, creating a string that starts with 3.14 and continues to the *n*th digit. The more digits one wants, the more computations it takes.

But it is also possible to skip ahead to the nth digit without calculating the preceding ones – for example, determining that the 10th digit is 3, without having to find the first 9 digits: 3.14159265. This is another way of testing computing power, since more computations are required to find higher values of *n*.

Now, Tsz-Wo Sze, a computer scientist at Yahoo in Sunnyvale, California, has used the company's computers to calculate the most distant digits yet.

PRINT SEND SHARE

Pushing pi to 9 trillion digits (Image: Mykl Roventine)

This week's issue

Subscribe

NewScientist FIFTY IDEAS THAT WILL CHANGE SCIENCE FOREVER

09 October 2010

ADVERTISEMENT

NewScientist

Visit the New Scientist

Other News Coverage

- New Pi Record Exploits Yahoo's Computers

  http://cacm.acm.org/news/99207-new-pi-record-exploits-yahoos-computers

- The Yahoo! boffin scores pi's two quadrillionth bit

  http://www.theregister.co.uk/2010/09/16/pi_record_at_yahoo

- Pi calculation more than doubles old record

  http://www.radionz.co.nz/news/world/57128/pi-calculation-more-than-doubles-o

- Hadoop used to calculate Pi's two quadrillionth bit

  http://www.zdnet.co.uk/blogs/mapping-babel-10017967/hadoop-used-to-calculate

- engadget Yahoo! researcher breaks Pi record in finding the two-quadrillionth digit

  http://www.engadget.com/2010/09/17/yahoo-researcher-breaks-pi-record-in-find

- Slashdot Nicholas Sze of Yahoo Finds Two-Quadrillionth Digit of Pi

  http://science.slashdot.org/story/10/09/16/2155227/Nicholas-Sze-of-Yahoo-Fin

- gather The 2,000,000,000,000,000th digit of the mathematical constant pi discovered

  http://news.gather.com/viewArticle.action?articleId=281474978525563

- MAXIMUMPC Researcher Shatters Pi Record by Finding Two-Quadrillionth Digit

  http://www.maximumpc.com/article/news/researcher_shatters_pi_record_finding_two-quadrillionth_digit

- A bigger slice of pi

  http://radar.oreilly.com/2010/09/strata-week-grabbing-a-slice.html

- 2 Quadrillionth digit of PI is found: Scientist celebration in worldwide Pandemonium

  http://engforum.pravda.ru/showthread.php?296242-2-Quadrillionth-digit-of-PI-i

- And the number is...0

  http://www.hexus.net/content/item.php?item=26505

- Pi Record Smashed as Team Finds Two-Quadrillionth Digit

  http://hardocp.com/news/2010/09/16/pi_record_smashed_as_team_finds_twoquadril

  digit

- WebProNews Yahoo Engineer Calculates Two Quadrillionth Bit Of Pi

  `http://www.webpronews.com/topnews/2010/09/17/yahoo-engineer-calculates-two-qu`

- ReadWrite CLOUD A Cloud Computing Milestone: Yahoo! Reaches the 2 Quadrillionth Bit of Pi

  `http://www.readwriteweb.com/cloud/2010/09/a-cloud-computing-milestone-ya.`
  `php`

- Thaindian News Yahoo researcher Nicolas Sze determines the 2,000,000,000,000,000th digit of the mathematical constant pi

  `http://www.thaindian.com/newsportal/sci-tech/yahoo-researcher-nicolas-sze-det`
  `100430278.html`

- ...

Other Results

- We also have computed
  - the first billion bits, and
  - around the positions $n = 10^m$ for $m \leq 15$.

- The first billion (10^9) bits
  - Arbitrary precision arithmetic

Starting Bit Position	Precision (bits)	Time Used	CPU Time	Date Completed
1	800,001,000	10 days	19 years	June 23, 2010
800,000,001	200,001,000	3 days	8 years	June 22, 2010

Ten & Hundred Trillion

- $n = 10^{13}, 10^{14}$
  - It appears that both results are new.
  - $n = 10^{13}$
    - Verified with Alexander Yee
    - 5 trillion decimal digits (August 2010)
    - $\approx 1.66 \cdot 10^{13}$ bits
    - These two results agree ☺

One Quadrillion

- $n = 10^{15}$
  The result is similar to the one obtained by PiHex except:
  - the chosen starting positions are slightly different
  - our result has higher precision (228-bit vs 64-bit)

  The overlapped bits of these two results agree. ☺

Agenda

- Introduction
- A New World Record
- **How to Compute The n^{th} Bits of π?**
- Computing π with Hadoop

The BBP Formula

- Bailey, Borwein and Plouffe (1996)

$$\pi = \sum_{k=0}^{\infty} \frac{1}{2^{4k}} \left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right)$$

The above equation is called **the BBP formula.**

- This remarkable discovery leads to the first *digit-extraction algorithm* for π in base 2.
  - allow computing the n^{th} bits without computing the earlier bits

Another BBP-type Formula

- Bellard (1997)

$$\pi = \sum_{k=0}^{\infty} \frac{(-1)^k}{2^{10k}} \left(\frac{2^2}{10k+1} - \frac{1}{10k+3} - \frac{2^{-4}}{10k+5} \right.$$
$$\left. - \frac{2^{-4}}{10k+7} + \frac{2^{-6}}{10k+9} - \frac{2^{-1}}{4k+1} - \frac{2^{-6}}{4k+3} \right)$$

- 43% faster than the BBP formula

Computing The $(n+1)^{\text{th}}$ Bits of π

- In order to obtain the $(n+1)^{\text{th}}$ bits,
  - multiply π by 2^n, and
  - take the fraction part,

$$\{2^n\pi\}, \qquad \text{where } \{x\} \stackrel{\text{def}}{=} x - \lfloor x \rfloor.$$

For examples,

$$\{3.14\} = 0.14 \qquad\qquad \text{(fraction part)}$$

$$\lfloor 3.14 \rfloor = 3 \qquad\qquad \text{(integer part)}$$

Example

- Suppose $n+1=9$.

$$
\begin{aligned}
\pi &= 11.00100100\ \underline{\overset{\overset{9}{\downarrow}}{0}0111111\cdots} \\
\{2^n\pi\} &= \{2^8\pi\} \\
&= \{11\ 00100100.\underline{00111111\cdots}\} \\
&= .\underline{00111111\cdots}
\end{aligned}
$$

The BBP Algorithm

- Using BBP formula

$$\pi = \sum_{k=0}^{\infty} \frac{1}{2^{4k}} \left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right),$$

we have

$$\{2^n \pi\} = \left\{ \sum_{k=0}^{\infty} \frac{2^{n+2-4k}}{8k+1} - \sum_{k=0}^{\infty} \frac{2^{n-1-4k}}{2k+1} - \sum_{k=0}^{\infty} \frac{2^{n-4k}}{8k+5} - \sum_{k=0}^{\infty} \frac{2^{n-1-4k}}{4k+3} \right\}.$$

Drop The Integer Part Earlier

$$\{2^n\pi\} = \left\{ \left\{ \sum_{k=0}^{\infty} \frac{2^{n+2-4k}}{8k+1} \right\} - \left\{ \sum_{k=0}^{\infty} \frac{2^{n-1-4k}}{2k+1} \right\} \right.$$
$$\left. - \left\{ \sum_{k=0}^{\infty} \frac{2^{n-4k}}{8k+5} \right\} - \left\{ \sum_{k=0}^{\infty} \frac{2^{n-1-4k}}{4k+3} \right\} \right\}$$

Drop The Integer Part Earlier ’

$$\{2^n\pi\} = \left\{ \left\{ \sum_{k=0}^{\infty} \left\{ \frac{2^{n+2-4k}}{8k+1} \right\} \right\} - \left\{ \sum_{k=0}^{\infty} \left\{ \frac{2^{n-1-4k}}{2k+1} \right\} \right\} \right.$$

$$\left. - \left\{ \sum_{k=0}^{\infty} \left\{ \frac{2^{n-4k}}{8k+5} \right\} \right\} - \left\{ \sum_{k=0}^{\infty} \left\{ \frac{2^{n-1-4k}}{4k+3} \right\} \right\} \right\}$$

Split The Summations

- For each sum, write

$$\left\{\sum_{k=0}^{\infty}\left\{\frac{2^{n+x-4k}}{yk+z}\right\}\right\} = \left\{\sum_{\substack{n+x-4k>0\\k\geq 0}} A_k + \sum_{\substack{n+x-4k\leq 0\\k\geq 0}} B_k\right\},$$

where

$$A_k \stackrel{\mathsf{def}}{=} \frac{2^{n+x-4k} \bmod (yk+z)}{yk+z},$$

$$B_k \stackrel{\mathsf{def}}{=} \frac{1}{2^{4k-n-x}(yk+z)}.$$

Split The Summations '

- For each sum, write

$$\left\{\sum_{k=0}^{\infty}\left\{\frac{2^{n+x-4k}}{yk+z}\right\}\right\} = \left\{\sum_{\substack{n+x-4k>0\\ k\geq 0}} A_k + \sum_{\substack{n+x-4k\leq 0\\ k\geq 0}} B_k\right\},$$

where

$$A_k \overset{\text{def}}{=} \frac{2^{n+x-4k} \bmod (yk+z)}{yk+z},$$

$$B_k \overset{\text{def}}{=} \frac{1}{2^{4k-n-x}(yk+z)}.$$

Split The Summations ”

- For each sum, write

$$\left\{\sum_{k=0}^{\infty}\left\{\frac{2^{n+x-4k}}{yk+z}\right\}\right\} = \left\{\sum_{\substack{n+x-4k>0\\k\ge 0}} A_k + \sum_{\substack{n+x-4k\le 0\\k\ge 0}} B_k\right\},$$

where

$$A_k \stackrel{\mathsf{def}}{=} \frac{2^{n+x-4k} \bmod (yk+z)}{yk+z},$$

$$B_k \stackrel{\mathsf{def}}{=} \frac{1}{2^{4k-n-x}(yk+z)}.$$

Evaluating The Summations

- The first sum

$$\left\{ \sum_{0 \le k < \frac{n+x}{4}} A_k \right\} = \left\{ \sum_{0 \le k < \frac{n+x}{4}} \frac{2^{n+x-4k} \bmod (yk+z)}{yk+z} \right\}$$

  - Number of terms: **linear to** n
  - Integer operations: **mod-powering**
  - Floating point operations: **division with a fixed precision**

Evaluating The Summations ’

- The second sum

$$\left\{\sum_{\frac{n+x}{4}\le k} B_k\right\} = \left\{\sum_{\frac{n+x}{4}\le k} \frac{1}{2^{4k-n-x}(yk+z)}\right\}$$

  - Number of terms: **linear to the precision**
  - Integer operations: **shifting**
  - Floating point operations: **reciprocal computation with a lower precision**

Algorithm Characteristics

- For **position** n and **precision** p,
  - *Running time*: $O(p(n^{1+\epsilon}+p))$ for any $\epsilon > 0$
    - ★ p small: essentially linear in n, $O(n^{1+\epsilon})$
    - ★ n small: quadratic in p, $O(p^2)$
  - *Space*: $O(p + \log n)$
  - *Embarrassingly parallel*:
    - ★ The summations can be easily split into many smaller summations.
    - ★ Easy to compute in parallel

Parameters

- Usually, we have
  - **large position** n (e.g. $2 \cdot 10^{15}$)
  - **small precision** p (e.g. 288)

Again,

- ★ running time is essentially linear, $O(n^{1+\epsilon})$;
- ★ space is only $O(\log n)$.

Errors

- Possible errors
  - *Rounding errors*: losing precision
  - *Hardware errors*: rare but hard to be detected
- For the new world record,
  - Two computations at two different positions
  - Only the bits covered by both computations are considered as valid results.

Starting Bit Position	Precision (bits)	Time Used	CPU Time	Date Completed
1,999,999,999,999,993	288	23 days	582 years	July 29, 2010
1,999,999,999,999,997	288	23 days	503 years	July 25, 2010

Agenda

- Introduction
- A New World Record
- How to Compute The n^{th} Bits of π?
- **Computing π with Hadoop**

MapReduce Summation

- The BBP algorithm basically evaluates the sum

$$S = \sum_{i \in I} T_i$$

  - each term T_i is simple

    $\left[\text{We have } A_k = \frac{2^{n+x-4k} \bmod (yk+z)}{yk+z} \text{ and } B_k = \frac{1}{2^{4k-n-x}(yk+z)}\right]$

  - I is a large index set

    $\left[\text{For position } n = 10^{15}, \text{ we have } |I| \approx 7 \cdot 10^{14} \text{ using Bellard's formula.}\right]$

A Straightforward Approach

- Partition the index set I into m pairwise disjoint subsets $I_1, \cdots, I_m$
- Then, compute the summation by a job with
  - m maps: each map evaluates

$$\sigma_j \stackrel{\text{def}}{=} \sum_{i \in I_j} T_i$$

  - Single reduce: compute the final sum

$$S = \sum_{1 \le j \le m} \sigma_j$$

Two Problems

- Multiple maps but one reduce
  - Fail to utilize reduce slots
- The job may run for a long time.
  - Need to persist the intermediate results

Starting Bit Position	Precision (bits)	Time Used	CPU Time	Date Completed
99,999,999,999,997	1024	4 days	37 years	June 11, 2010
100,000,000,000,001	1024	5 days	40 years	June 7, 2010
999,999,999,999,993	288	13 days	248 years	July 2, 2010
1,000,000,000,000,001	256	25 days	283 years	July 6, 2010
1,999,999,999,999,993	288	23 days	582 years	July 29, 2010
1,999,999,999,999,997	288	23 days	503 years	July 25, 2010

Multi-level Partitioning

- Partition the sum into many small jobs

Final Sum:	$S = \sum_{1\le j\le m} \Sigma_j$
Jobs:	$\Sigma_j = \sum_{1\le k\le m_j} \sigma_{j,k}$
Tasks:	$\sigma_{j,k} = \sum_{1\le t\le m_{j,k}} s_{j,k,t}$
Threads:	$s_{j,k,t} = \sum_{i\in I_{j,k,t}} T_i$

- Write the intermediate results into HDFS

Map-side & Reduce-side Computations

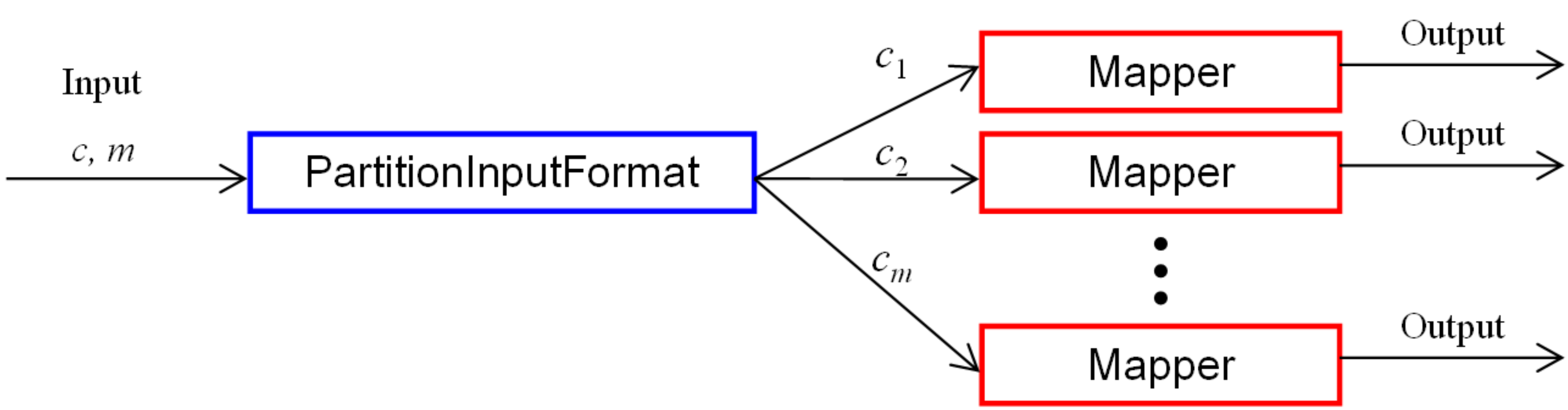
- Developed a *generic framework* to execute tasks on either the map-side or the reduce-side.

- Applications only have to define two functions:

  - partition(c, m): partition the computation c into m parts $c_1, \ldots, c_m$
  - compute(c): execute the computation c

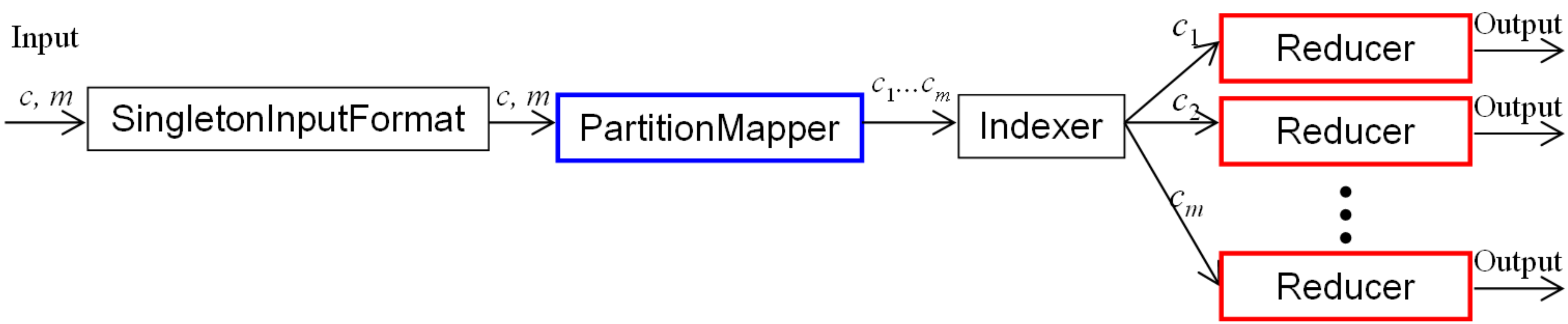
Map-side Job

- Contains **multiple mappers** and **zero reducers**
  - A PartitionInputFormat partitions c is into m parts
  - Each part is executed by a mapper



Reduce-side Job

- Contains **a mapper** and **multiple reducers**
  - A SingletonInputFormat launches a PartitionMapper
  - An Indexer launches m reducers.



Utilizing The Idle Slices

- Monitor cluster status
  - Submit a map-side (or reduce-side) job if there are sufficient available map (or reduce) slots.
- Small jobs
  - Hold resource only for a short period of time
- Interruptible and resumable
  - can be interrupted at any time by simply killing the running jobs

Running The Jobs

Running Jobs

Jobid	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total	Reduces Completed	Job Scheduling Information
job_201006091641_93488	NORMAL	tsz	1,999,999,999,999,996-288/P20_3.job9332	100.00%	1	1	99.99%	100	97	0 running map tasks using 0 map slots. 0 additional slots reserved. 3 running reduce tasks using 3 reduce slots. 0 additional slots reserved.
job_201006091641_93491	NORMAL	tsz	1,999,999,999,999,996-288/P20_3.job9335	100.00%	1	1	99.99%	100	96	0 running map tasks using 0 map slots. 0 additional slots reserved. 4 running reduce tasks using 4 reduce slots. 0 additional slots reserved.
job_201006091641_93492	NORMAL	tsz	1,999,999,999,999,996-288/P20_3.job9336	100.00%	1	1	99.99%	100	92	0 running map tasks using 0 map slots. 0 additional slots reserved. 8 running reduce tasks using 8 reduce slots. 0 additional slots reserved.
job_201006091641_93494	NORMAL	tsz	1,999,999,999,999,996-288/P20_3.job9337	99.99%	200	199	0.00%	0	0	1 running map tasks using 1 map slots. 0 additional slots reserved. 0 running reduce tasks using 0 reduce slots. 0 additional slots reserved.
job_201006091641_93495	NORMAL	tsz	1,999,999,999,999,996-288/P20_3.job9338	99.99%	200	179	0.00%	0	0	21 running map tasks using 21 map slots. 0 additional slots reserved. 0 running reduce tasks using 0 reduce slots. 0 additional slots reserved.
job_201006091641_93497	NORMAL	tsz	1,999,999,999,999,996-288/P20_3.job9339	100.00%	1	1	99.99%	100	36	0 running map tasks using 0 map slots. 0 additional slots reserved. 64 running reduce tasks using 64 reduce slots. 0 additional slots reserved.
job_201006091641_93499	NORMAL	tsz	1,999,999,999,999,996-288/P20_3.job9340	100.00%	1	1	99.99%	100	30	0 running map tasks using 0 map slots. 0 additional slots reserved. 70 running reduce tasks using 70 reduce slots. 0 additional slots reserved.
job_201006091641_93500	NORMAL	tsz	1,999,999,999,999,996-288/P20_3.job9341	100.00%	1	1	99.99%	100	52	0 running map tasks using 0 map slots. 0 additional slots reserved. 48 running reduce tasks using 48 reduce slots. 0 additional slots reserved.
job_201006091641_93501	NORMAL	tsz	1,999,999,999,999,996-288/P20_3.job9342	100.00%	1	1	99.99%	100	23	0 running map tasks using 0 map slots. 0 additional slots reserved. 77 running reduce tasks using 77 reduce slots. 0 additional slots reserved.
job_201006091641_93502	NORMAL	tsz	1,999,999,999,999,996-288/P20_3.job9343	100.00%	1	1	99.99%	100	45	0 running map tasks using 0 map slots. 0 additional slots reserved. 55 running reduce tasks using 55 reduce slots. 0 additional slots reserved.
job_201006091641_93503	NORMAL	tsz	1,999,999,999,999,996-288/P20_3.job9344	100.00%	1	1	99.99%	100	28	0 running map tasks using 0 map slots. 0 additional slots reserved. 72 running reduce tasks using 72 reduce slots. 0 additional slots reserved.
job_201006091641_93505	NORMAL	tsz	1,999,999,999,999,996-288/P20_3.job9345	100.00%	1	1	99.99%	100	3	0 running map tasks using 0 map slots. 0 additional slots reserved. 97 running reduce tasks using 97 reduce slots. 0 additional slots reserved.
job_201006091641_93508	NORMAL	tsz	1,999,999,999,999,996-288/P20_3.job9346	99.99%	200	117	0.00%	0	0	83 running map tasks using 83 map slots. 0 additional slots reserved. 0 running reduce tasks using 0 reduce slots. 0 additional slots reserved.

The World Record Computation

- 35,000 MapReduce jobs, each job either has:
  - 200 map tasks with one thread each, or
  - 100 reduce tasks with two threads each.
- Each thread computes 200,000,000 terms
  - ∼45 minutes.
- Submit up to 60 concurrent jobs
- The entire computation took:
  - 23 days of real time and 503 CPU years

Thank you!