David DeWitt, Mike Stonebraker, and Sarah Palin: You Were Right (by the way, I need a recommendation letter)

> Daniel Abadi Yale University October 27th, 2009

Well, Maybe, You Used to Be Right (but can I still have a recommendation letter?)

The Database Column

A multi-author blog on database technology and innovation.



Home				
			-	
	22 m 1	 	a 1	
		L.E.	8.3	_

About Contributors

Subscribe to feed

	12	100		
-	LA			

Search

About this Post

This page contains a single post by David DeWitt published on January 17, 2008 4:20 PM.

Relational databases for storing and querying RDF was the previous entry in this blog.

MapReduce II is the next entry in this blog.

Find recent content on the <u>main index</u> or look in the <u>archives</u> to find all content.

Blogs we read

DBMS2

MapReduce: A major step backwards

By David DeWitt on January 17, 2008 4:20 PM | Permalink | Comments (42) | TrackBacks (1)

[Note: Although the system attributes this post to a single author, it was written by David J. DeWitt and Michael Stonebraker]

On January 8, a Database Column reader asked for our views on new distributed database research efforts, and we'll begin here with our views on <u>MapReduce</u>.

As both educators and researchers, we are amazed at the hype that the MapReduce proponents have spread about how it represents a paradigm shift in the development of scalable, data-intensive applications. MapReduce may be a good idea for writing certain types of general-purpose computations, but to the database community, it is:

- 1. A giant step backward in the programming paradigm for large-scale data intensive applications
- 2. A sub-optimal implementation, in that it uses brute force instead of indexing
- 3. Not novel at all -- it represents a specific implementation of well known techniques developed nearly 25 years ago
- 4. Missing most of the features that are routinely included in current DBMS
- 5. Incompatible with all of the tools DBMS users have come to depend on

First, we will briefly discuss what MapReduce is; then we will go into more detail about our five reactions listed above.

What is MapReduce?

The basic idea of MapReduce is straightforward. It consists of two programs that the user writes called map and reduce

🛃 My Computer

4. MapReduce is missing features

All of the following features are routinely provided by modern DBMSs, and all are missing from MapReduce:

- · Bulk loader -- to transform input data in files into a desired format and load it into a DBMS
- Indexing -- as noted above
- Updates -- to change the data in the data base
- Transactions -- to support parallel update and recovery from failures during update
- · Integrity constraints -- to help keep garbage out of the data base
- · Referential integrity -- again, to help keep garbage out of the data base
- · Views -- so the schema can change without having to rewrite the application program

In summary, MapReduce provides only a sliver of the functionality found in modern DBMSs.

5. MapReduce is incompatible with the DBMS tools

A modern SQL DBMS has available all of the following classes of tools:

- · Report writers (e.g., Crystal reports) to prepare reports for human visualization
- · Business intelligence tools (e.g., Business Objects or Cognos) to enable ad-hoc querying of large data warehouses
- Data mining tools (e.g., Oracle Data Mining or IBM DB2 Intelligent Miner) to allow a user to discover structure in large data sets
- · Replication tools (e.g., Golden Gate) to allow a user to replicate data from on DBMS to another
- Database design tools (e.g., Embarcadero) to assist the user in constructing a data base.

MapReduce cannot use these tools and has none of its own. Until it becomes SQL-compatible or until someone writes all of these tools, MapReduce will remain very difficult to use in an end-to-end task.

1 TrackBacks

Listed below are links to blogs that reference this entry: MapReduce: A major step backwards.

TrackBack URL for this entry: http://www.databasecolumn.com/blog/mt-tb.cgi/26

» <u>This is one of the web's most interesting stories on Fri 18th Jan 2008</u> from purrl.net |** urls that purr **| These are the web's most talked about URLs on Fri 18th Jan 2008. The current winner is .. <u>Read More</u>

Tracked on January 18, 2008 12:09 AM

42 Comments

Ronn Brashear said:

As an MR advocate, I can agree with several of the above points. Certainly, MR development shouldn't ignore previous research, nor should it be constrained by it. MR is directed at a different problem from the modern DBMS.

For example, using MR to rapidly identify small subsets of data is a bad idea. However MR is a good large-data manipulation tool - something for which DBs are notoriously bad. A grid DB's indexing offers no advantage when computing page rank of the internet for example. Indices are pure overhead in that situation.

I am concerned the authors are suggesting that introducing MR into academia is a bad idea since *that is where most of the previous literature is well understood.* Some of the best improvements to MR lately have been based on distributing reductions ala Monet's continuous near-neighbor load distribution. To say MR doesn't have high level languages/tools/optimizations is short-sighted. Pig, Sawzall, and others functional languages are in development. Additional tools, research, and optimization will follow. Presenting MR as a research topic will enable that growth.

For engineers, the underlying issue is picking the right tool for the job. RDB versus Flat Files versus MQL versus MR smacks of the same "religious" debates between Java versus C++ versus Ruby versus assembly and is generally a waste of effort. A good engineer understands the specific problem space, examines the potential solutions, and picks the right tool for the job.

January 17, 2008 6:52 PM

ade said:

You seem to be under the impression that MapReduce is a database. It's merely a mechanism for using lots of machines to process very large data sets. You seem to be arguing that MapReduce would be better (for some value of better) if it were a data warehouse product along the lines of TeraData. Unfortunately the resulting tool would be less effective as a general purpose mechanism for processing very large data sets.

You seem to have made a category error in this article: <u>http://en.wikipedia.org/wiki/Category_mistake</u>

P

and wants to learn more about it. But it just feels to me like these DB's didn't really get a fair comparison here.

January 18, 2008 12:43 AM

Robert Weisman said:

MapReduce is not a DBMS. It is a framework for developing distributed systems. Ican't quite imagine why such a namework would nave indexes, or why the concepts of updates, integrity, transactions, views, or bulk loaders would even be meaningful. God forbid, one could even write a MapReduce to read from a DBMS. Some of these criticisms may be valid with Bigtable, but then Bigtable is probably the only DBMS designed to handle petabytes of data.

January 18, 2008 1:25 AM

Ilya Haykinson said:

I think your arguments are solid in that MapReduce represents a step backwards compared to a traditional DBMS. However, I believe that your overall point is severely off the mark, since I believe that your comparison is not entirely fair.

MapReduce is not a database framework. Instead, it's a computational framework. Inlike a database, it does not offer storage or oata, or transactions, or indeed any sort or a query tanguage. To see mapReduce as a database system is to seriously miscategorize its use.

Additionally, you write that there's a question of MapReduce's ability to scale. I think that Google's track record here is a great testament to the system's abilities: with hundreds of thousands of nodes, the system seems to be battle-tested.

Instead of looking at MapReduce you may want to look at <u>BigTable</u> -- Google's database technology. I think that it's a more fair comparison and probably deserves a thorough review.

January 18, 2008 2:33 AM

Joe Developer said:

Mapreduce's charm is, I gather, a combination of excellent fault tolerance and utter simplicity. There isn't a whiff of database about it, it's more like a simple pipe. And sometimes a pipe is all you need.

gasper_k said:

Hi,

MapReduce isn't meant to replace a RDB; It doesn't need indices, ordering, grouping and practically everything you wrote in itematical Entropy in the indices work by iterating over data in the given order and producing an output. Also, you can have a schema and data validation just as well, so again it doesn't fall short.

As for item 3, MapReduce not being novel is hardly an argument against it, is it?

January 18, 2008 12:21 PM

NAC said:

This is the basic problem with database people. They view everything as a database, and that everything must be done in/with a database.

MapReduce is for a different class of problems. eg if you were using MR to process large quantities of small image tiles, how would you forumulate that in teradata, what advantage does an RDBMS bring?

Relational databases are as much an inhibitor to modern application development as they are an asset. Hence the massive use of complex object relational mappers to work around their problems.

All technologies have strengths and weaknesses, MR and databases included. These are two different things, don't confuse them.

January 18, 2008 4:32 PM

Chris G said:

I think you could write an equivalent article:

"Airplanes: A major step backward"

You could say something like, "roads are good, everyone knows that, why would you throw them away?" You could talk about fuel consumption, possible crashes, etc.

For the knock out punch, give an example why cars are definitely better. Describe taking your kids to school in a plane would be ridiculous; a car is so much better.

I don't think anyone seriously suggests replacing all databases with MR. It'd be a terrible solution for a small database.

When Vertica is running Google, let me know. I'll get in line for the product.

January 18, 2008 5:05 PM

Greg Grasmehr said:

Interesting opinion and set of comments thus far; thanks to everyone who has provided input. Interesting reading to say the least.

January 18, 2008 5:19 PM

Stephan Wehner said:

Could one of the bonus points of MapReduce be that "It works" or "It doesn't cost much"?

Stephan

January 18 2008 6+24 DM

- L Re:This coming from the DB Community? by geekboy642 (Score:2) Friday January 18 2008, @09:33PM
 - L Re:This coming from the DB Community? by Qbertino (Score:2) Saturday January 19 2008, @07:10AM
 - L Re:This coming from the DB Community? by mdfst13 (Score:2) Saturday January 19 2008, @10:58PM
 - <u>1 reply</u> beneath your current threshold.

The only thing wrong with map-reduce... (Score:2)

by frank adrian314159 (469671) on Friday January 18 2008, @05:55PM (#22102138) Homepage

... is that they misspelled xapping [homeunix.net].

Stream processing. (Score:2)

by mypalmike (454265) on Friday January 18 2008, @06:06PM (#22102302) Homepage

The whole point of MapReduce is to take an unindexed stream of data and shrink it down based on some criteria where numerous records can be associated (Map) and aggregated (Reduce). It is a process. The *result* of the process is an indexed database, which is often inserted into a relational or time-series database.

It's an apples and oranges comparison, and the author's never eaten an orange.

reminiscent...(philosophical digression) (Score:1)

by cionslashdot (904508) on Friday January 18 2008, @06:43PM (#22102730)

All your comments bring back to my mind the criticisms of XML-based messaging technologies (SOAP, Web services). "A huge step backwards", "incompatible with existing technologies and approaches" (BNF, parsers, languages), "inefficient" (compared to binary formats), etc. Those complaints were right, but they fell on deaf ears, just as these will.... IT is driven by fads and the availability of high-productivity gizmos. Ironically, productivity often suffers in the long run, as people then have to deal with the mess that gets created using approaches that are fundamentally wrong.

Index Every Column? (Score:2)

by Tablizer (95088) on Friday January 18 2008. @07:29PM (#22103242) Homeosoe Journal

😂 Internet

🔁 100% 🛛 👻

Re:may be missing the (data)points (Score:4, Insightful)

by CajunArson (465943) on Friday January 18 2008, @04:02PM (#22100220) Journal

Are these guys just trying to stake a reputation based on being critical of Google? I tend to agree, I could probably write a nice article about how map-reduce would be a terrible system to use in making a 3D game. Could an article like that be technically the source would be an article than a logical non-sequice? Not timess Google and of the studen came out and claimed mapreduce is the new platform for all 3D game development (not likely).

Parent

Re:may be missing the (data)points by MajinBlayze (Score:1) Friday January 18 2008, @05:21PM

Re:may be missing the (data)points (Score:5, Funny)

by abscondment (672321) on Friday January 18 2008, @06:55PM (#22102862) Homepage

It's also terrible for painting.

- Since the bucket doesn't enforce any schema, you never know what color paint the bucket might hold. Heck, it could even be full of honey. You just can't know, and not being able to know is, well, like programming assembly.
- Buckets aren't indexed, so you're not able to find that one ounce of paint that you really want to use next. You've
 got to split up all of the paint into ounce cups each time and examine very cup. It's very intensive, and really
 slows down your painting. If you stored the paint in a B-tree of ounce cups, your search for the right ounce of
 paint would be much more efficient.
- Painting is so old. I mean, get with the program. Gold plate your house, or something newer (since newer is always better!). In fact, decades of research into titanium has determined that it'll hold up better to the elements, anyway, so you should just get titanium siding instead of painting.
- 4. Painting is an incomplete process. What if you want a window? Yeah, you can't paint a window for yourself, now can you? Did you need a jacuzzi? A fireplace? A new car? Sorry! Painting doesn't support those features yet. You'd better not paint at all if you want those things.
- 5. Painting, believe it or not, is incompatible with tennis. There's no racket, there's no court, and there's no ball. There's not even a net (unless you're working from a really tall building, in which case you might fall and so a net is often used). I mean, you don't even need to paint with another person. It's so... incompatible.

Parent

🕄 100% 🛛 👻

	 Instance Rj consists of the files Fi, j, 1 ≤ i ≤ N After being collected by the map-reduc reduce instance are grouped on their keys (by sorting or hashing) and feed to the reduce prog To draw an analogy to SQL, map is like the group-by clause of an aggregate query. Reduce is (e.g., average) that is computed over all the rows with the same group-by attribute. Stonebraker and DeWitt point out that MapReduce is frequently touted as a replacement for traditiona such as relational databases. The advantage of MapReduce compared with other data management s fault tolerance, but MapReduce ignores the most important lessons learned from hard-won experienc The database community has learned the following three lessons from the 40 years that have IMS in 1968: Schemas are good. Schemas are good. Schemas are good. High-level access languages are good. MapReduce has learned none of these lessons and represents a throw back to the 1960s, befor The DBMS community learned the importance of schemas, whereby the fields and their data to importantly, the run-time system of the DBMS can ensure that input records obey this schema. application from adding "garbage" to a data set. MapReduce has no such functionality, and the out of its data sets. A corrupted MapReduce dataset can actually silently break all the MapReduc dataset 	e framework, the input records to ram. analogous to the aggregate functi al data management techniques an systems is that provides a high de- e, according to the authors: unfolded since IBM first released ore modern DBMSs were invented. ypes are recorded in storage. Mor This is the best way to keep an re are no controls to keep garbag uce applications that use that that many MapReduce implementa	ion nd tools, gree of e je ations		
Brian Slesinsky	Re: MapReduce: A Major Step Backwards	Posted: Jan 21, 2008 3:53 PM	🚱 Reph		
Posts: 37 Nickname: skybrian Registered: Sep, 2003	I think they're just confused. Of course doing a query using an index is far faster when one exists, but the index has to be built first. MapReduce is a fine tool for building custom indexes.				
Morel Xavier	Re: MapReduce: A Major Step Backwards	Posted: Jan 22, 2008 2:28 AM	ᢙ Reply		
Posts: 61 Nickname: masklinn Registered: Sep, 2005	The part Stonebraker and DeWitt seem to have missed, which was discussed in depth in responses to their articles and which you also seem to have missed, is that MapReduce is <i>not</i> a database technology and it's <i>not</i> supposed to be or replace one.				
	Thus, all criticisms of mapreduce based on it not being a good enough database are, well, pretty strange.				
		😜 Internet	🕄 100% 🔻		

seed media group

^

Socie	February 6, 2009 Contemporary Arts Center thefeastconference.com	
Good Mat Finding the fun in good math Squashing bad math and the Latest Posts Archives	h, Bad Math Tools who promote it About RSS Contact	ScienceBlogs™ - Last 24 Hours - Syndication Feeds (RSS)
Search Google ^m Custom Search Search	« Before Groups from Categories: a Category Refresher Main Capturing More Symmetry using Categories: Groupoids » Databases are hammers; MapReduce is a screwdriver.	Email Subscriptions The SB Weekly Recap BLOGS IN THE NETWORK
Profile	Category: goodmath > Programming: Technology Posted on: January 22, 2008 10:34 AM, by Mark C. Chu-Carroll A bunch of people have sent me links to <u>an article about MapReduce.</u> I've hesitated	
Hank Charles Hank (CO) is	to write about it, because the currently hyped <u>MapReduce</u> stuff was developed, and extensively used by Google, my employer. But the article is really annoying, and deserves a response. So I'm going to be absolutely clear. I am <i>not</i> commenting on this in my capacity as a Google employee. (In fact, I've never actually used MapReduce at work!) This is strictly on my own time, and it's purely my own	Your INNER FISH A JOURNEY INTO 3.3-BILLION-YEAR HISTORY BY 788 HUMAN BODY Will change
Mark Chu-Carroll (aka MarkCC) is a PhD Computer Scientist, who works for Google as a Software Engineer. My professional interests center on programming languages and tools, and how to improve the languages and tools that are used for building	opinion. If it's the dumbest thing you've ever read, that's my fault, not Google's. If it's the most brilliant thing you've ever read, that's my credit, not Google's. I wasn't asked to write this by Google, and I didn't ask their permission, either. This is just me, the annoying geek behind this blog, writing solely on my own behalf, speaking for me and but me. Get it?	NEIL SHUBIN NEIL SHUBIN TOP FIVE / MOST GERMAN
complex software systems.	for no one but me. Got it?	I - LL I

8

😜 Internet

🔍 100% 📼

CODE CODEX

JUST LIKE JIM ROCKFORD. EXCEPT NOT AS COOL. AND A PROGRAMMER. ALSO, REAL.

ABOUT ME



JAKOB HOMAN WA, UNITED STATES

I'm a senior in the Computer and Software

Systems program at the University of Washington, Bothell. I'm interested in Java, C#, Ruby and currently exploring distributed computing and computational linguistics. Right now I'm working on creating a 64-node Hadoop cluster at our university and developing applications using the MapReduce framework. Please feel free to email me.

TUESDAY, JANUARY 22, 2008

Yep, MapReduce isn't a relational database. So what?

About a week ago, MapReduce (and by extension, Hadoop) was broad-sided by some database guys in a rather hyperbolic post: MapReduce: A Major Step Backwards. The MapReduce community seemed quite taken aback, mostly because the attack came out of nowhere and amounted mainly to "MapReduce isn't a relational database, therefore it sucks." I'm not sure how one follows the other, but ok, whatever normalizes your data.

Even though our Hadoop project is over (I'll get to posting the final matrix multiplication code to Google Code soon), and I don't know when I'll get the opportunity to do more MapReducing, I've grown quite fond of this particular parallel processing paradigm that could. Luckily, Mark Chu-Carroll over at ScienceBlogs, has written a very well though-out and funny rebuttal to the rather odd charge of one tool not being some other tool.

Incidentally, if you're not reading Mark's blog every day, you should be. He'll make your brain bigger.

CODED BY JAKOB HOMAN WHO WAS STILL AWAKE AT 8-14 PM VIEW MY COMPLETE PROFILE Digg submit LABELS: HADOOP THE PAST ... 2008 (26) September (1) **o SNARKY REPLIES:** Nut graf fail Post a Comment August (3) May (3) LINKS TO THIS POST ► April (4) Croate a Link P 😜 Internet 100%

Arnon Rotem-Gal-Oz's Cirrus Minor

The DBMS vs. Map/Reduce -

January 19, 2008 (a) 09:39 PM

is that really a competition?



I quess they might have meant to talk about another Google tool called



NAVIGATION

Home

- Papers, Articles & Presentations
- SPAMMED Architecture Framework

SOA Patterns

About Me

FEATURED PRESENTATIONS & PAPERS

REST introduction (ppt) SOA Pattern Presentation (pdf) Fallacies of Distributed Computing (pdf) Getting SPAMMED for architecture (pdf) OO Primer (ppt) Use Case Methodology for large systems (pdf) Software Architecture (ppt) Service Oriented Architecture - Intro (ppt) What is SOA anyway? (pdf) More...

SOA PATTERNS BOOK



😜 Internet

追 Error on page.



For my 8 hour trip to Kalamazoo last week, I printed some Google white papers for some "light reading". One of these was <u>MapReduce: Simplified Data Processing on Large Clusters</u>, which was recently updated. I read the original version last year and wanted to catch up. From the paper:

66

Programmers find the system easy to use: more than ten thousand distinct MapReduce programs have been implemented internally at Google over the past four years, and an average of one hundred thousand MapReduce jobs are executed on Google's clusters every day, processing a total of more than twenty petabytes of data per day.

Meh. 20 petabytes? You should see the JAR files in *my* app, I tell ya...

Seriously, I did not read that article and think..."hmm...that's kind of like a database." I cannot imagine anyone thinking that. Nor is MapReduce an index. You can use it to "create" an index, for example. But still...MapReduce does not compete with a database in any way. It is entirely different, for an entirely different kind of problem. Yet, we have the DeWitt/Stonebraker article, described next.

More Interesting Reading

😜 Internet

P

+ Comments on Joel's "How Hard Could It Be? Five Easy Ways to Fail"

Doing it wrong: getters and setters -

Relational Database Experts Jump The MapReduce Shark

January 17th, 2008 | Databases, Programming

In this article relational database experts David DeWitt and Michael

systems (RDBMSs) and find MapReduce wanting. They make some strong points in favor or relational databases, but the comparison is not appropriate. When I finished reading the article I was thinking that the

or why programmers might be excited about non-RDBMS ways to manage data.

The article makes five points:

1. MapReduce is a step backwards in database access

They're right about that, but MapReduce is not a database system. In the **the Wikipedia article describing MapReduce** that the authors link to from the article the word "database" isn't mentioned, and only one of the footnotes points to a paper about MapReduce applications in relational data processing. MapReduce is not a data storage or management system — it's an algorithmic technique for the distributed processing of large amounts of data. Google's web crawler is a real-life example. I'm not

sure what MapReduce has to do with schemas or separating the data from the application but apparently MapReduce is just Codasyl all over again. The authors know relational databases but I wonder if they know how useful a hash map of key:value pairs can be.

MapReduce has the same relationship to RDBMSs as my motorcycle has to a snowplow — it's a step backwards in snowplow technology if you look at it that way.



Search

To search, type and hit enter

Recent Posts

- Tips for successful freelancing
- Database Thaw Reheated
- Video Professor announces Haskell, Scheme, and Smalltalk training
- Comment correcting software closes \$25m funding
- Miasma: a new framework for web applications
- Doing it wrong: getters and setters
- → Relational Database Experts Jump The MapReduce Shark
- → Comments on Joel's "How Hard Could It Be? Five Easy Ways to Fail"
- Why Programmers Don't Like Relational Databases
- Thirteen Patterns Of
 - Programmer Interviews
- → Abject-Oriented Databases
- The Agile Method and Other Fairy Tales: QED
- → Keep Legacy Databases Away From Your Code
- Jack Of All Languages, Master

MapReduce has the same relationship to RDBMSs as my motorcycle has to a sno...

#4 My daily readings 01/18/2008 « Strange Kite on 01.18.08 at 4:39 am

[...] My daily readings 01/18/2008 Typical Programmer - Relational Database Experts Jump The MapReduce Shark [...]

#5 **Ben** on 01.18.08 at 5:09 am

Excellent and well written article. I don't know how the "experts" got it so wrong!

#6 Franchu on 01.18.08 at 5:27 am

When I was reading the original post I was wondering if the authors understanding was totally disconnected from what MapReduce is, or whether I was missing something.

Reading your post was very gratifying to see that I was not the only one thinking what you brilliantly expressed.

#7 James Urquhart on 01.18.08 at 6:00 am

When reading the article a few minutes ago, a lot of it came across as utter hogwash. Especially when they mixed MapReduce up with being a database.

Just another case of writing without thought i guess.

#15 **JOE** on 01.18.08 at 10:48 am

Other articles I'm inspired to write based on the referenced article:

I tried using MapReduce to create a website, and it sucks compared to markup with CSS. It doesn't have any concept of how to style a website. MapReduce is a serious step backward in terms of web design.

I also tried to have MapReduce babysit my kids, and I came back half an hour later to find that it was just sitting there crunching data, and wasn't watching them at all. This thing can't do anything

Also, compared to a standard hammer, this MapReduce things is really crappy at pounding nails into things.

#16 test 01/18/2008 « Strange Kite on 01.18.08 at 10:50 am

[...] Typical Programmer - Relational Database Experts Jump The MapReduce Shark [...]

#17 Nathan Fiedler on 01.18.08 at 3:08 pm

Very pleased to see, based on the comments on the original article, and your blog, that _no one_ was fooled by these supposed experts. They seem to lack even a basic understanding of how MR is used. I feel they must have read the white paper and thought they knew more than anyone else who had read the same paper. Thank you for the rebuttal.



DBMS MUSINGS

THURSDAY, OCTOBER 22, 2009

Analysis of the "MapReduce Online" paper

I recently came across a paper entitled "MapReduce Online" written by Tyson Condie, Neil Conway, Peter Alvaro, Joe Hellerstein, Khaled Elmeleegy, and Russell Sears at Berkeley (University of California). Since I'm very interested in Hadooprelated research (see my group's work on HadoopDB) and this Berkeley group have historically produced reliably good research papers, I immediately downloaded the paper and read it carefully. The paper demonstrates the impact of several improvements the group made to Hadoop for interactive queries, and since Hadoop is becoming increasingly popular, I expect this paper to have wide interest. Therefore, I think it might be useful to post a summary of the paper and some analysis on this blog. If you have also read this paper, I encourage discussion in the comment thread of this post.

Overview

The authors argue that since MapReduce's (and therefore Hadoop's) roots are in batch processing, design decisions were made that are cause problems as Hadoop gets used more and more for interactive query processing. The main problem the paper addresses is how data is transferred between operators ---both between 'Map' and 'Reduce' operators within a job, and also across multiple MapReduce jobs. This main problem has two subproblems: DANIEL ABADI



ABOUT ME DANIEL ABADI NEW HAVEN, CT. UNITED STATES

Daniel Abadi is an Assistant Professor of Computer Science at Yale University. His research interests are in database system architecture and implementation, cloud computing, and the Semantic Web. He has been a recipient of a Churchill Scholarship, an NSF CAREER Award, the 2008 SIGMOD Jim

æ

😜 Internet

X

🖓 🔻 🔍 100% ·

Fast Forward Two Years

Case study: Facebook

- Wanted to create an enterprise data warehouse
- Brought in all the usual suspects EDW vendors for extensive POCs
- In the end, they went with Hadoop!
- Added their own SQL-layer on top of Hadoop (Hive)
- Recent statistics
 - 2.5 petabyte enterprise data warehouse
 - Adding 15TB of new data a day
- Entire companies (e.g. Stampede) are being started that specialize in using Hadoop to create data warehouses

DeWitt And Stonebraker Were Right!!!!

It's not apples and oranges
A comparison is not only possible, it's necessary

Well, They Used to be Right

Now Stonebraker is saying that Hadoop and DBMSs can coexist in the enterprise
Even Mike Olson (founder of Cloudera!) says that Hadoop need not encroach on the traditional DBMS space
I disagree ---- Hadoop will increasingly be used for apps that used to be the realm of DBMSs

People Who Choose Not to Use Database Systems Aren't Dumb

There must be a reason DBMSs are too expensive Free / open source databases like MySQL/PostgreSQL/Ingres don't scale out of the box Proprietary solutions price by the TB or per CPU - DBMSs are too hard to use – DBMSs don't scale • Yes, they should scale in theory. But in practice they don't scale. Even the expensive solutions.

Scalability Problems With Today's Parallel Databases



HadoopDB

 Project at Yale whose goal is to turn Hadoop into a parallel DBMS

- Basic idea: supplement HDFS with a single-node free/open-source DBMS on each node
- Store structured data in the database instances
- Leverages 2 decades of research in DBMS literature

E.g. indexing, compression, direct operation on compressed data, I/O sharing, column-wise storage
 See VLDB 2009 paper

HadoopDB Architecture



HadoopDB Performance



HadoopDB Scalability



Conclusions

- Hadoop will be used more and more for parallel data analysis
- It's performance will be worse than parallel databases
 - But it's free, easy to use, and more scalable
- HadoopDB working on improving performance of Hadoop on structured data analysis
 - Facebook, Yahoo, and Berkeley also working on Hadoop performance