Big Data Challenges in a Product Group

Johannes Gehrke
Microsoft
Big Data

- 40% growth in data per year
- Cost of a disk drive to hold the world’s music: <$400

- Big Data: Data that is too large to store and analyze using traditional database systems.

- The three “V”s:
  - Volume
  - Velocity
  - Variety
Driving Factors: The Strategic Power of Data
Driving Factors: The Strategic Power of Data (Contd.)
Driving Factors: The Cloud
Driving Factors: The Cloud (Contd.)

- Example: Azure

- Scale:
  - 22 public regions
  - 250,000+ public facing websites
  - Millions of SQL databases
  - Tens of trillions of objects in Azure storage
  - Hundreds of millions of enterprise users
  - Tens of billions of authentications per week with Azure Active Directory

- Reach
  - Titanfall: 100,000 visual machines at launch day
  - Visual Studio Online: Released Nov 2013, now millions of active users

Source: Forbes.com
Driving Factors: The Office Cloud

• Let’s look at the cloud service behind the Outlook client
• Union of Exchange commercial service and Outlook.com
• Scale:
  • >2million commercial organizations
  • 500+ million accounts; 50+ million engaged users
  • 70+ billion messages per day
  • Over 100K servers
  • Exabytes of physical storage
  • Many 9s availability
Enterprise software used to be about making existing work more efficient. Now, the opportunity for software is to transform the work itself.
Disconnected set of tools for individuals

Hierarchical and siloed teams

Siloed Tools and Teams...
Office Graph
The intelligent cloud fabric delivering personalized experiences and insights
Home

27 of your colleagues viewed this recently

Northwind & Contoso Proposal

Amanda Cunningham Commented yesterday

Marketing - Professional Development

Dane Seale Commented today

Employee Handbook - Updates

Chris Magnuson

Ron Chitwood

Sarah Talley

Milton Dominquez

Linda Holman Commented today

Tablet Renewal Orders

Northwind Partnership

Future Scenarios Weekly Sync
Delve and Office Graph: Concepts

- Move enterprise search beyond IR technology (TF/IDF)
- Entity centric-notion of navigating content
- Data as a strategic asset
Microsoft Graph Developer API’s

https://graph.microsoft.com/

Insights from Office Graph

Other MS cloud services
Office Graph: Technology

• Confluence of:
  • Distributed systems
  • Database systems/Big data
  • Data science/ML

• This talk:
  • FARM [NSDI’14, SOSP’15]
  • Trill [VLDB’14]
  • Cloud abstractions for collaboration [SIGMOD’15]
This Talk

• Scalable transactions with FARM
• Collaboration through data-driven coordination
• MSR Impact in Big Data
Access Control in the Enterprise

- On the web, we have access to every document over which we search
  - At least at coarse granularity
- In the enterprise, (in general) every user sees a different subset of documents → The “personal view of the enterprise graph that a user sees is different for every user
What We Need: Low Latency at Scale

[Diagram showing latency vs scale with SQL Server and FARM highlighted]

- Latency:
  - ms
  - s
- Scale:
  - 1 machine
  - > 1K machines

MapReduce
FARM simplifies building low latency apps

- transactions and replication simplify programming
- hide failures, distribution, and concurrency
- **abstraction** that transactions run sequentially on a reliable server
Avoid compromises by harnessing new hardware

- strong consistency
  - strict serializability
- high availability
  - recovery from failure to peak performance in tens of milliseconds
- good performance
  - millions of transactions per second with sub millisecond latencies
FARM is enabled by three hardware trends

• cheap DRAM
  • currently <$5/GB
  • cost effective to keep data in DRAM

• non-volatile DRAM
  • DRAM + SSD + lithium ion batteries in power supply
  • 1.15x cost of DRAM

• fast commodity networks with RDMA
  • CX3: 40 Gb/s, 35 M msg/s, and 3 μs latency on Ethernet
  • CX4: 100 Gb/s, 150 M msg/s

existing software benefits little because of CPU bottlenecks
The hardware is not enough: throughput

- **new software**
- **today's software**

![Graph showing the throughput](graph.png)

- **Reads/s/server (millions)**
- **Bytes read (log)**

**Graph Description:**
- The graph compares the throughput of new software and today's software across various byte read values.
- The x-axis represents the number of bytes read (in log scale), while the y-axis shows the reads per second per server (in millions).
- The throughput decreases as the number of bytes read increases.
- New software consistently outperforms today's software across all byte read values.
The hardware is not enough: latency

![Graph showing latency vs. bytes read for new and today's software.](image)
Remote direct memory access (RDMA)

• One-sided reads and writes of remote memory
  • remote CPU not involved
  • kernel bypass
• FARM uses RDMA to reduce CPU overhead
  • one-sided reads of remote data
  • one sided writes into transaction logs
Why one-sided operations?

Operations / µs

Series 1 vs. Series 2

Transfer size

Operations / µs vs. Transfer size (4x speedup)
FARM provides transactional memory
FARM exploits new hardware effectively

• New transaction, replication, and recovery protocols
• Three principles
  • Use RDMA to eliminate CPU overhead on the server side
  • Minimize RDMA operations to reduce CPU on the client side: Use as few messages as possible
  • Exploit parallelism effectively: Allow new transactions to be performed during recovery
FARM exploits new hardware effectively

• New transaction, replication, and recovery protocols
• Three principles
  • Use RDMA to eliminate CPU overhead on the server side
  • Minimize RDMA operations to reduce CPU on the client side: Use as few messages as possible
  • Exploit parallelism effectively: Allow new transactions to be performed during recovery
FARM API

storage and transactions

Transaction CreateTransaction()
ObjBuf *Transaction::Alloc(size, locality_hint)
ObjBuf *Transaction::Read(addr, size)
ObjBuf *Transaction::OpenForWrite(ObjBuf)
void Transaction::Free(ObjBuf)
void Transaction::Commit()
Scalability TATP

- **Hekaton (no fault tolerance)**
- **FARM (with fault tolerance)**

![Graph showing scalability TATP with markers for Hekaton and FARM, indicating a 20x increase in performance with FARM.](image-url)
Performance TATP

- Latency 99%
- Latency 50%

Latency $\mu$s

Throughput ops/$\mu$s
FARM: Summary

• Scaling out distributed transactions and graph operations with
  • strong consistency
  • high availability
  • high throughput
  • low latency
This Talk

• Scalable transactions with FARM
• Collaboration through data-driven coordination
• MSR Impact in Big Data
Ubiquitous Devices

2 Billion
5 Billion
7 Billion
50 Billion
10 Trillion
Driving Factors: Social Applications

web 2.0
Enabling Coordination

• Assume Tom and Meg want to coordinate itineraries
  • Fly on the same flight, in adjacent seats
  • Also stay in the same hotel if possible
Coordination: Enrollment

• Students want to enroll in classes with their friends
• Help with homework/moral support
• Already happens with out-of-band communication

• Interesting coordination scenarios:
• Negative constraints: Avoid the section my ex-* is in
• Strong mutual dependencies: I will take this tough class only if my smart friend Ashwin takes it too
Coordination: SIGMOD 2011

Room Sharing among attendees of the 2011 ACM SIGMOD Conference

The conference officers have set up a web page where interested attendees of the conference can register their interest in sharing rooms at the conference hotel. Through this service attendees can enter their details so that interested people can contact each other.

To register your interest, please submit your information at: http://bit.ly/sigm_share_room (URL shortener service forwarding to a Google Spreadsheets form). This service is provided solely as a convenience to participants that seek to share accommodation costs. Please contact directly participants that have expressed interest. The organizers will not be involved in the process nor are they responsible for possible abuse of the information you provide.
Sharing a room at the Conference Hotel?

This form allows people who want to stay at the conference hotel to express their interest in sharing rooms. Please fill out the following form, all people expressing interest in sharing a room can then contact each other by looking at the following page http://bit.ly/sigmod_share_room

* Required

**Name**

*Required Field*

**Email**

*Required Field*

**Period you wish to stay at the hotel**

*Optional Field*

Please add any constraints on sharing a room (gender, etc)

*Optional Field*

Submit

Powered by Google Docs

Report Abuse - Terms of Service - Additional Terms
<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Name</th>
<th>email</th>
<th>Period you wish to stay at the hotel</th>
<th>Please add any constraints on sharing a room (gender, etc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5/10/2011 6:15:25</td>
<td>Ugo,</td>
<td><a href="mailto:unrep@asadmar.com">unrep@asadmar.com</a></td>
<td>12/06/2011-16/05/2011</td>
<td>Female</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5/10/2011 6:38:39</td>
<td>Xuan Tran Pham</td>
<td><a href="mailto:xuansp99@yahoo.com">xuansp99@yahoo.com</a></td>
<td>June 13 - June 17 (4 nights)</td>
<td>I will be interested to share a room (only females) during the conference. thanks!</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5/10/2011 16:38:53</td>
<td></td>
<td></td>
<td>12-17 June</td>
<td>Males only. I already have a room reservation -- looking to fill the other bed and split the cost.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5 nights June 12-16 (inclusive)</td>
<td></td>
</tr>
<tr>
<td>5/10/2011 18:34:49</td>
<td></td>
<td></td>
<td>13th-16th June</td>
<td>prefer females (i'm a girl)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>12-17 June</td>
<td></td>
</tr>
<tr>
<td>5/12/2011 13:03:30</td>
<td></td>
<td></td>
<td>Check-in 11, Check-out 15</td>
<td>n/a</td>
</tr>
<tr>
<td>5/13/2011 12:45:54</td>
<td></td>
<td></td>
<td>12th-17th</td>
<td>I'm easy going :)</td>
</tr>
<tr>
<td>5/14/2011 12:20:15</td>
<td></td>
<td></td>
<td>June 12 - June 17</td>
<td>male</td>
</tr>
<tr>
<td>6/4/2011 13:10:08</td>
<td>Wilco</td>
<td><a href="mailto:danjil@ias.gmu.edu">danjil@ias.gmu.edu</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Data-Driven Coordination

It is not just the applications that are data-driven, the coordination itself is data-driven!

Users want to agree on a choice of data values, not on the time of day of when they call each other to jointly enroll in a course

Today typically achieved with out-of-band communication
• Or through an ad-hoc solution for a given scenario...

We want to defer the coordination until a suitable value is there
Data-Driven Coordination

• Provide a declarative abstraction and mechanism to support coordination
  • Being declarative is fundamental principle in query and update languages
  • Coordination pertains to data, so should be expressed at the same level
  • Meg says: “Book me a ticket on the same flight as Tom”
  • System takes care of the actual coordination
Data-Driven Coordination

• Goal: Provide a declarative abstraction and mechanism to support coordination
  • Being declarative is fundamental principle in query and update languages
  • Coordination pertains to data, so should be expressed at the same level
  • Meg says: “Book me a ticket on the same flight as Tom”
  • System takes care of the actual coordination
Data-Driven Coordination

• Goal: Provide a declarative abstraction and mechanism to support coordination
  • Being declarative is fundamental principle in query and update languages
  • Coordination pertains to data, so should be expressed at the same level
  • Meg says: “Book me a ticket on the same flight as Tom”
  • System takes care of the actual coordination
Data-Driven Coordination

- Goal: Provide a declarative abstraction and mechanism to support coordination
  - Being declarative is fundamental principle in query and update languages
  - Coordination pertains to data, so should be expressed at the same level
  - Meg says: “Book me a ticket on the same flight as Tom”
  - System takes care of the actual coordination
ACID Transactions and Coordination

ACID Properties of a transaction
• Atomicity
• Consistency
• Isolation
• Durability

Coordination requires relaxing isolation
• For semantic reasons, not for performance (such as lower isolation levels, eventual consistency)
• We still want atomicity and durability

And the communication due to coordination should be “controlled”
• Keep the “residual" isolation
Entangled Queries

• Example scenario: Harry and Voldemort want to travel to NYC on the same flight

• In addition, Harry needs to travel on United
Harry’s Entangled Query

SELECT ‘Harry’, fno INTO ANSWER Reservation
WHERE
  fno IN (SELECT fno FROM Flights WHERE dest=‘JFK’)
  AND (‘Voldemort’, fno) IN ANSWER Reservation
CHOOSE 1
Harry’s Entangled Query (Contd.)

SELECT ‘Harry’, fno INTO ANSWER Reservation
WHERE
    fno IN (SELECT fno FROM Flights WHERE dest='JFK')
    AND (‘Voldemort’, fno) IN ANSWER Reservation
CHOOSE 1

- Voldemort’s answer must also be in the Reservation table
Voldemort’s Entangled Query

SELECT ‘Voldemort’, fno INTO ANSWER Reservation
WHERE
  fno IN (SELECT fno FROM Flights F, Airlines A
            WHERE F.dest='JFK' and F.fno = A.fno AND
              A.airline = ‘United’ )
  AND ('Harry', fno) IN ANSWER Reservation
CHOOSE 1

SELECT ‘Harry’, fno INTO ANSWER Reservation
WHERE
  fno IN (SELECT fno FROM Flights WHERE dest='JFK')
  AND ('Voldemort', fno) IN ANSWER Reservation
CHOOSE 1
Entangled Queries

• Harry and Voldemort want to travel to NYC on the same flight
• In addition, Harry needs to travel on United
What We Want: Transactions

• Scenario
  • Harry and Voldemort want to fly to NYC on the same flight
  • If they can make a flight booking together, then they want to stay in the same hotel
  • This booking should be a transaction
Harry’s Entangled Transaction

BEGIN TRANSACTION WITH TIMEOUT 2 DAYS;

SELECT `Harry`, fno, fdate AS @ArrivalDay INTO ANSWER FlightReservation
WHERE fno, date IN (SELECT fno, fdate FROM Flights WHERE dest=`JFK`
AND (`Voldemort`, fno, fdate) IN ANSWER FlightReservation
CHOOSE 1;

-- (Code to perform flight booking)

SET @StayLength = `2011-10-30` - @ArrivalDay;

SELECT `Harry`, hid, @ArrivalDay, @StayLength INTO ANSWER HotelReservation
WHERE hid IN (SELECT hid FROM Hotels WHERE location=`NYC`
AND (`Voldemort`, hid, @ArrivalDay, @StayLength) IN ANSWER HotelReservation
CHOOSE 1;

-- (Code to perform hotel booking)

COMMIT;
Consistency

Recall ACID consistency:
• Every transaction, if executed by itself on a consistent database, will produce another consistent database.

What is the analogous property for entangled transactions?

BEGIN TRANSACTION WITH TIMEOUT 2 DAYS;
SELECT 'Harry', fno, fdate AS @ArrivalDay INTO ANSWER FlightReservation
WHERE fno, date IN (SELECT fno, fdate FROM Flights WHERE dest="JFK")
AND ("Voldemort", fno, fdate) IN ANSWER FlightReservation
CHOOSE 1;

-- (Code to perform flight booking omitted)
SET @StayLength = '2011-10-30' - @ArrivalDay;
SELECT 'Harry', hid, @ArrivalDay, @StayLength INTO ANSWER HotelReservation
WHERE hid IN (SELECT hid FROM Hotels WHERE location="NYC")
AND ("Voldemort", hid, @ArrivalDay, @StayLength) IN ANSWER HotelReservation
CHOOSE 1;

-- (Code to perform hotel booking omitted)
COMMIT;
Consistency

Entangled Query Oracle
- Process that executes alongside an entangled transaction
- For an entangled query, the oracle chooses a valid answer (=ground the query on the database) and returns it to any entangled query
- Has no direct effect on the database’s state

Oracle Consistency:
- Suppose an entangled transaction executes by itself on an initially consistent database, using an entangled query oracle to obtain answers to the entangled queries. Then the execution will produce another consistent database.
Isolation

Anomaly-based definition:
• No dirty reads, no unrepeatable reads, etc.

Two new anomalies
• Widowed Transactions
• Unrepeatable quasi-reads
New Anomaly 1: Widowed Transactions

Harry
Voldemort

Flight entangled queries
Ticket booking code
Hotel entangled queries
Room booking code
ABORT
New Anomaly 2: Unrepeatable Quasi-Reads

Entangled queries

Read Flights
Read Flights, Airlines
Entangled query evaluation

SELECT * FROM Airlines
WHERE airline = `United`

INSERT INTO Airlines
VALUES (125, `United`)
Eliminating These Anomalies

• How to avoid widowed transactions?
  • Group commit of all the transactions that are connected through entangled queries

• Unrepeatable quasi-reads
  • Appropriate locking of data structures
Putting Everything Together

• Traditional ACID Properties:
  • Atomicity
  • Consistency → Oracle Consistency
  • Isolation → Two new phenomena
  • Durability

• We can now define
  • Oracle-serializability: Serial schedule with a suitable oracle that provides answers to entangled queries
  • Entangled isolation: Schedule does not have any anomalies

• **Putting everything together**: Any schedule that is entangled-isolated is also oracle-serializable

• Integrates existing transactions and entangled transactions seamlessly
Putting Everything Together (Contd.)

• We are starting to understand the basics of low-level cloud abstractions, their performance and how new hardware impacts them
  • This is also a research pattern that we understand well

• But we are just starting to understand higher-level programming/ML abstractions
  • I talked about coordination
  • We see in general a need for ranking, recommendation, talking to devices, security, etc.
  • This is a much less understood research pattern
This Talk

• Scalable transactions with FARM
• Collaboration through data-driven coordination
• MSR Impact in Big Data
Trill: Fast Streaming Analytics Library

• **Performance**
  • 2-4 orders of magnitude faster than traditional SPEs
  • For relational queries, comparable to best DBMS
  • User-controlled latency specification
    • explicit latency vs. throughput tradeoff

• **Fabric & language integration**
  • Built as high-level language (HLL) library component
  • Works with arbitrary HLL data-types & libraries

• **Query model**
  • Extended LINQ syntax based on tempo-relational query model
  • Supports broad & rich analytics scenarios (relational, progressive, time-based)
Trill’s Use Cases

• Azure Stream Analytics Cloud service
• With Scope for Bing Ads
• With Orleans for Halo game monitoring & debugging
• ...

• Key enabler: performance + fabric & language integration + query model

Orleans

- Orleans distributed agent platform: [http://github.com/dotnet/orleans](http://github.com/dotnet/orleans)
  - All back end services: Players, games, statistics, regions, scoreboards, ...
  - 10s of services, 10s to 100s of machines each, 100Ks of requests per second
  - Bursty load (evenings, weekends) and peak load at product launch
- Orleans public preview, April 2014. Open source in GitHub, January 2015.
- Other Microsoft users: Skype, Azure, MSN, Game Studios (many games), ...
  - Examples: intelligent cache, telemetry, user profiles.
- Many 3rd parties: Honeywell, Mesh Systems (IoT), Gigya (identity mgmt), ...

References:
Bw-Tree/LLAMA

- Key-sequential index in SQL Server Hekaton
  - Lock-free for high concurrency, consistent with Hekaton’s overall non-blocking main memory architecture

- Indexing engine in Azure DocumentDB
  - Rich query processing over a schema-free JSON model, with *automatic indexing*
  - Sustained document ingestion at high rates

- Sorted key-value store in Bing ObjectStore
  - Support range queries
  - Optimized for flash SSDs

- [http://research.microsoft.com/deuteronomy/](http://research.microsoft.com/deuteronomy/)
<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>12</td>
<td>Flu</td>
</tr>
<tr>
<td>Bob</td>
<td>51</td>
<td>Diabetes</td>
</tr>
<tr>
<td>Chen</td>
<td>24</td>
<td>Flu</td>
</tr>
<tr>
<td>Dan</td>
<td>36</td>
<td>Cold</td>
</tr>
</tbody>
</table>

**Cipherbase**

Untrusted Machine

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>X%*!</td>
<td>)C</td>
<td>!x8J</td>
</tr>
<tr>
<td>~4Yz</td>
<td>##</td>
<td>)zFr#x</td>
</tr>
<tr>
<td>T$H2</td>
<td>!*</td>
<td>^@tG</td>
</tr>
<tr>
<td>&lt;*fB</td>
<td>@$</td>
<td>BxU3</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Secure Co-Processor

Disease = ‘Flu’


Arvind Arasu, Ken Eguro, Manas Joglekar, Raghav Kaushik, Donald Kossmann, Ravi Ramamurthy: Transaction processing on confidential data using cipherbase. ICDE 2015: 435-446

Arvind Arasu, Spyros Blanas, Ken Eguro, Raghav Kaushik, Donald Kossmann, Ravishankar Ramamurthy, Ramarathnam Venkatesan: Orthogonal Security with Cipherbase. CIDR 2013
Summary

• We are pushing the boundaries on scale, performance
  • Office Graph, Delve, Microsoft infrastructure

• We are looking for ways to make our developers and customers more efficient through high-level abstractions

• ASG/MSR have a great collaboration in many areas
  • Now only performance improvements, but new classes of problems
Thank You

johannes@microsoft.com
References

• FARM
  • Aleksandar Dragojevic, Dushyanth Narayanan, Miguel Castro, Orion Hodson: FaRM: Fast Remote Memory. NSDI 2014: 401-414
  • Aleksandar Dragojevic, Dushyanth Narayanan, Edmund B. Nightingale, Matthew Renzelmann, Alex Shamis, Anirudh Badam, Miguel Castro: No compromises: distributed transactions with consistency, availability, and performance. SOSP 2015: 54-70

• Declarative Data-Driven Coordination
  • Sudip Roy, Lucja Kot, Gabriel Bender, Bailu Ding, Hossein Hojjat, Christoph Koch, Nate Foster, Johannes Gehrke: The Homeostasis Protocol: Avoiding Transaction Coordination Through Program Analysis. SIGMOD Conference 2015: 1311-1326
  • Nitin Gupta, Lucja Kot, Sudip Roy, Gabriel Bender, Johannes Gehrke, Christoph Koch: Entangled queries: enabling declarative data-driven coordination. SIGMOD Conference 2011: 673-684