The Aurora and Borealis Stream Processing Engines

Ugur Cetintemel Daniel Abadi Yanif Ahmad Hari Balakrishnan
Magdalena Balazinska Mitch Cherniack Jeong-Hyon Hwang
Wolfgang Lindner Samuel Madden Anurag Maskey Alexander Rasin
Esther Ryvkina Mike Stonebraker Nesime Tatbul Ying Xing
Stan Zdonik

Discussant presentation:
Craig Hawkins
craig_hawkins@brown.edu
March 02, 2015

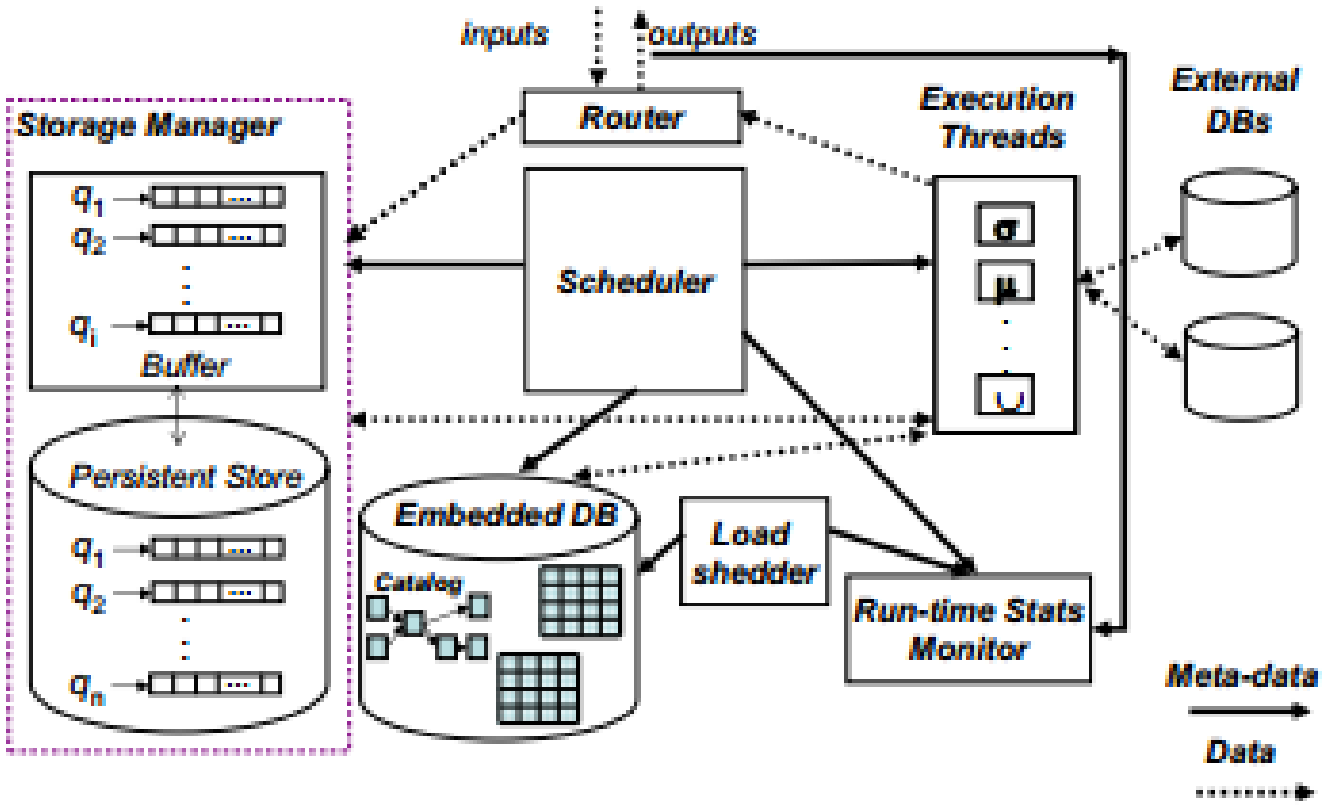


Fig. 5. Aurora Run-Time Architecture

Is this system provably correct?

For all valid inputs, does Aurora halt on the correct output?

SQuA1 ... has this been fully proven for correctness? "The Aurora Query Algebra" pg. 14

Ugur Centintemel: databases, systems (Brown)
Daniel Abadi: database systems (Yale)
Yanif Ahmad: data mgt. (Johns Hopkins)
Hari Balakrishnan: networks (M.I.T.)
Magdalena Balazinska: databases (U Washington)
Mitch Cherniack: databases, systems (Brandeis)
Jeong-Hyon Hwang: databases, dist. sys (SUNY Albany)
Wolfgang Lindner: databases, medical and distributed information systems, wireless sensor networks and mobile computing, information system security, algorithms, and e-business systems (M.I.T.)
Samuel Madden: databases, networks (M.I.T.)
Anurag Maskey: databases (Brandeis PhD candidate)
Alexander Rasin: databases (Brown)
Esther Ryvkina: databases (?)
Mike Stonebraker: databases (M.I.T.)
Nesime Tatbul: stream processing (M.I.T.)
Ying Xing: ?
Stan Zdonik: databases, systems (Brown)

How about one of these nice people?

1

2

3

5

4

1,2 3: cs.brown.edu
4. cs.dartmouth.edu
5. theory.stanford.edu

Maybe this guy too:

1

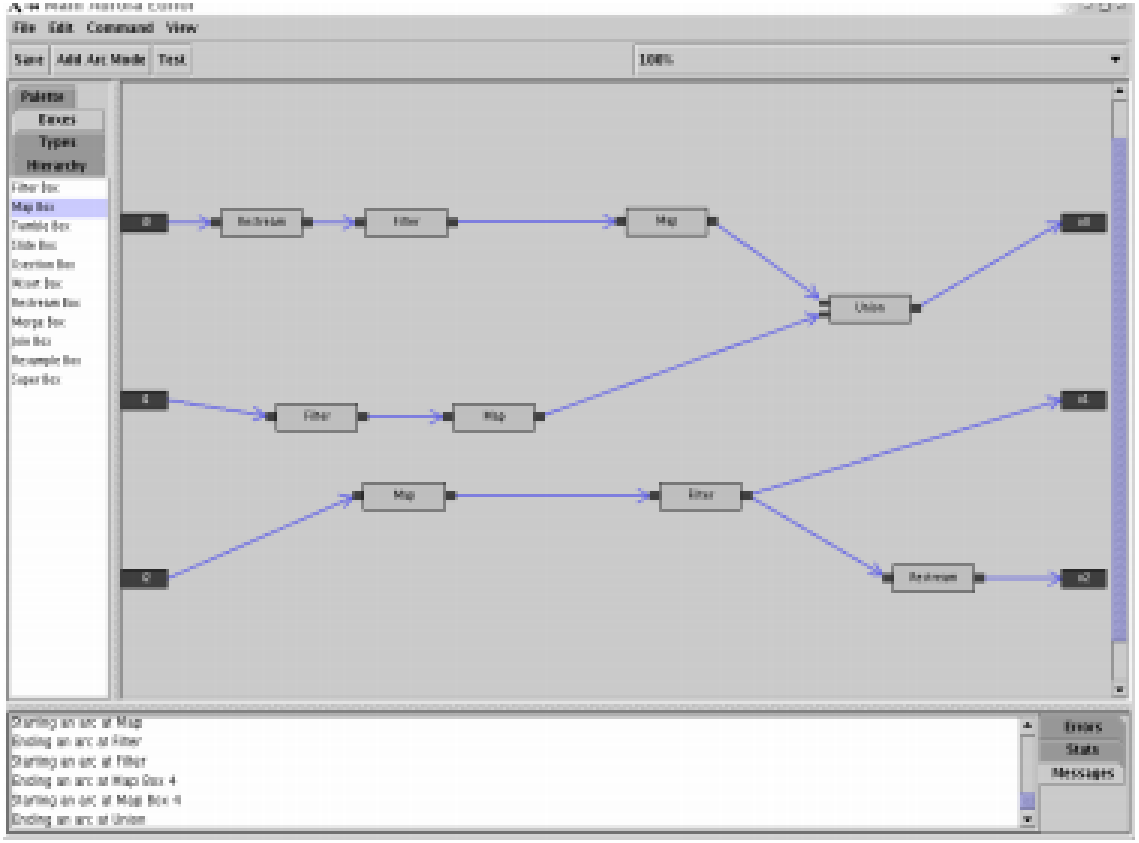


Fig. 1. Aurora Graphical User Interface

1: cs.brown.edu

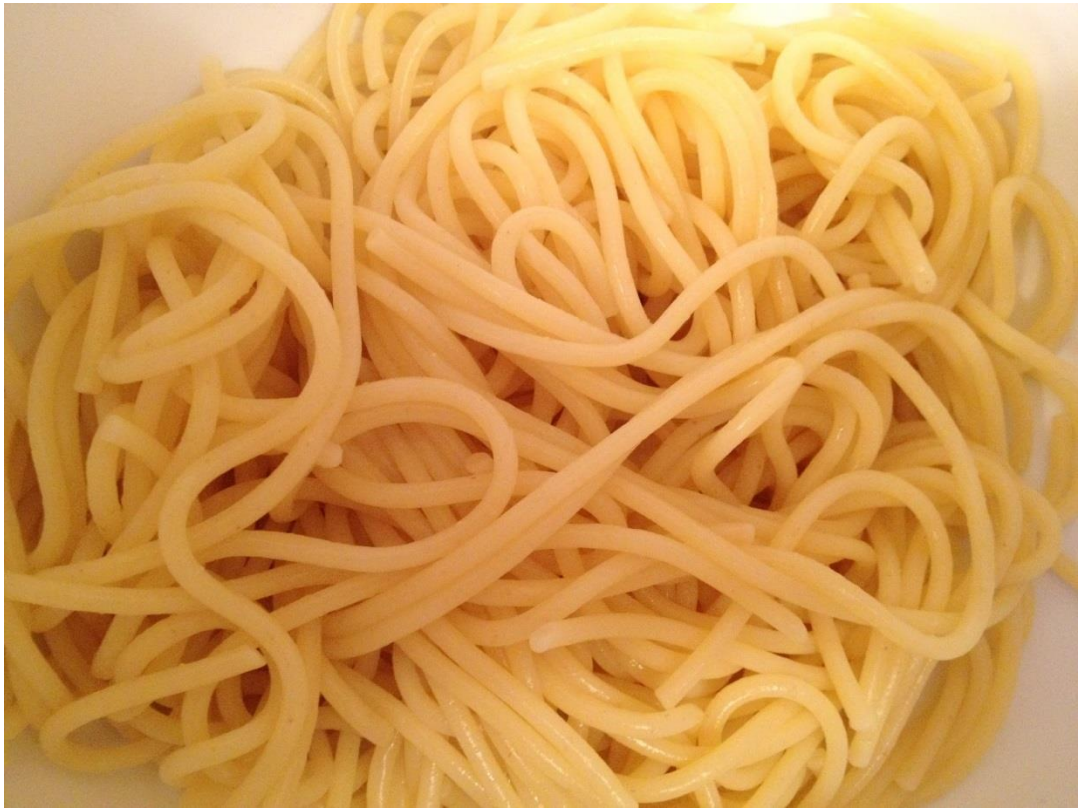
2: Aurora paper, 2007 Springer

Apple's top person:

1

1: wikipedia.org
SIr Jonathan Ive, holder of hundeds of design and utility patents.
getnetworth.com: est. net worth $130 million

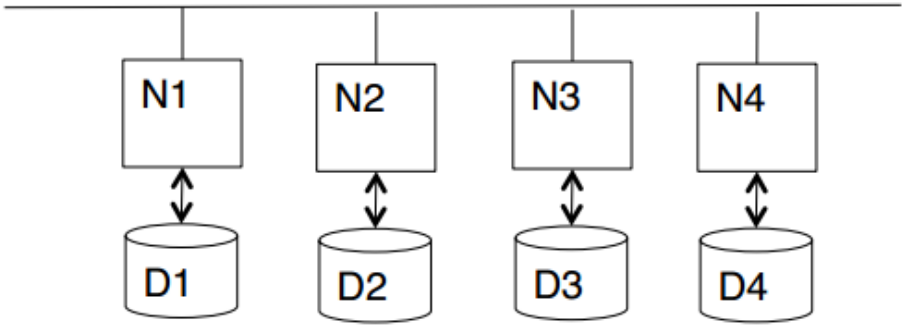
Spaghetti doesn't scale



1

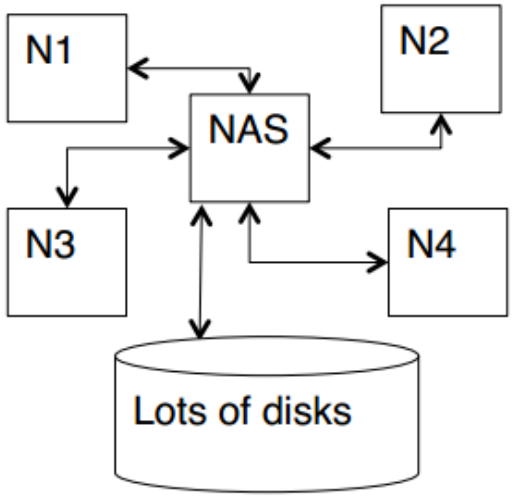
1. Microsoft clipart inside of PowerPoint

Shared Nothing



scales

Shared-Disk



does not scale

1

1. http://cs.brown.edu/courses/csci1270/files/lectures/L19_ParallelDBs_1.pdf
NAS: "network-attached storage" , RAID, etc.

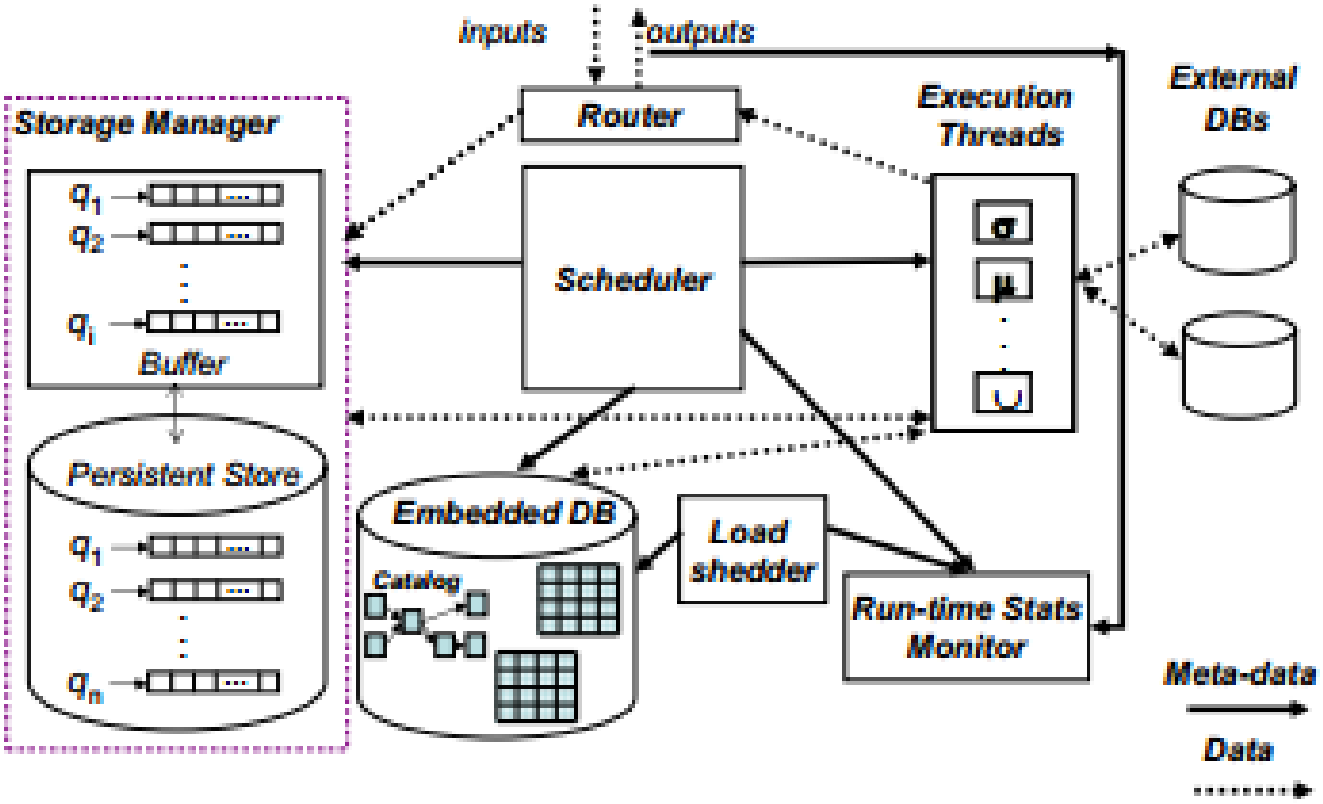
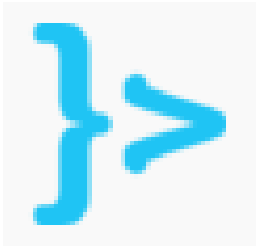


Fig. 5. Aurora Run-Time Architecture

not looking clean...

Frameworks



Languages

Easy (easier) to learn
Awkward to modify

Difficult to learn
Flexible

all images on this slide except django: wikipedia.org
django image from www.django.com

Frameworks are good for:

Standardization of code

People who need to work quickly

People who lack fully-formed coding skills

Examples in paper:
Financial Markets
Military
Highway Traffic Agencies

None of these entities are in a hurry
to roll out a product in 72 hours.

Every one of them can (and does)
hire professionally-skilled programmers.

That leaves code standardization as the key attractor.

Or does it?

"Overall, the entire application ended up consisting of 3400 lines of C++ code ... and a 53-operator Aurora query network".

1. Aurora paper, pg 12, discussing the environmental monitoring application build.
3400 lines of code, plus Aurora, to monitor 5 attributes of fish and their environment.
(breathing rate; temperature, pH, oxygenation, conductivty of water)

Aurora paper, pg 7: "We worked with a major financial services company on developing an Aurora application that detects feed problems and triggers the switch in real time.

Aurora paper, pg 12: "It seems likely that this application was developed at least as quickly in Aurora as it would have been with standard procedural programming." (environmental monitoring project)

How is this a savings in programmer time?

With user interfaces and software, there is a tradeoff between power and ease of use.

Aurora was struggling to find its voice in the coding ecosystem.

"Aurora's GUI for designing query networks ...proved invaluable"

"We felt the need for an API"

"Offer Aurora... as a library"

"Programmatic interfaces... are a good idea"

"XML adaptor required"

1. Aurora paper, pg 12 2. Aurora paper, pg 13 3. Aurora paper, pg 13 4. Aurora paper, pg 17
5. Aurora paper, pg 16

Where's the benchmark?

23 pages, and not a single performance metric to be found

1

1. www.wikipedia.org

streaming databases are not new... too mature to not have benchmarks

A camel is a horse designed by a commitee.[1]

2

1. source unknown
2. Microsoft PowerPoint clip art

critique on the writing quality of the paper

Authors Suppressed Due to Excessive Length[1]

Linear Road generic test described in detail. Performance with Aurora never detailed in the paper.

General waste of space describing external studies. Space could have been used to prove correctness and performance of system.

No summary or conclusion in paper.[2]

QoS mentioned multiple times before defined.[3]

Useless prognostications about the future.

Only a thin discussion of Borealis.

1. Aurora paper, pgs. 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22
2. Christian Mathiesen spotted this facet.
3. "quality of service"