Reality-based engineering

Putting data, query and analytic in the hands of the people who use them

L.G. Meredith

Excutive summary

Objective: to build a next generation Analytics-as-a-Service offering, bringing a new generation of analytics and process modeling capabilities “on line”.

Overview

- A simple business scenario
- Features of a solution that meets today's standards
- Technical details
- Background

The pitch -- business scenario

- Many business solutions and many kinds of domains, such as financials and logistics, require modeling
- Example: model the impact of oil price fluctuations on logistics costs, which impact my business' profits
    - Turns out that Vector Auto-regressive models are the best practice for this sort of financial modeling problem

Properties of VAR models for this business scenario

- You don't really want to know how VAR models are employed to solve this
- You want to treat it as a black box to wire into the more relevant calculations of profits based on possible changes to logistics costs
    - Exposes the need for *compositional* solutions -- wiring into larger calculations should be as simple as modifying the internals of the VAR model
- The model needs to be feed time-series data from recent samples of oil price fluctuations -- information that is available on the web

Pipes: editing pipe

http://localhost:3000/

Google

Apple (45) Amazon eBay Yahoo! News (36)

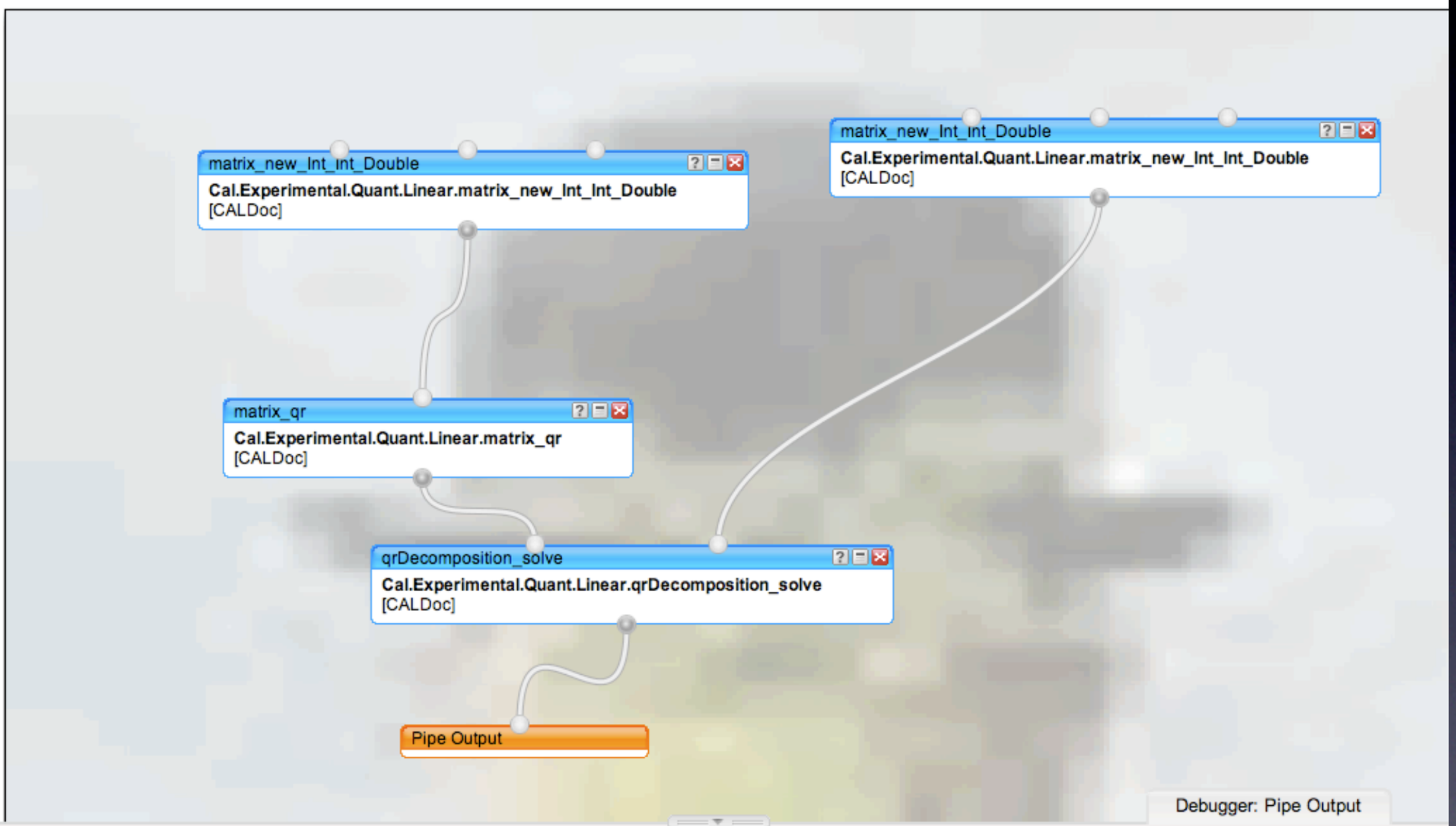
Pipes: editing pipe

untitled*

Layout | Expand All | Collapse All | Back to My Pipes | New | Save | Properties...

Cal.Experimental.Quant.Linear

- qrDecomposition_wait_Long_Int
- qrDecomposition_wait_Long
- qrDecomposition_wait
- qrDecomposition_toString
- qrDecomposition_solve
- qrDecomposition_notifyAll
- qrDecomposition_notify
- qrDecomposition_isFullRank
- qrDecomposition_hashCode
- qrDecomposition_getR
- qrDecomposition_getQ
- qrDecomposition_getH
- qrDecomposition_getClass
- qrDecomposition_equals
- qrDecomposition_new
- maths_wait_Long_Int
- maths_wait_Long
- maths_wait
- maths_toString
- maths_notifyAll
- maths_notify
- maths_hypot
- maths_hashCode
- maths_getClass



Debugger: Pipe Output

Features of a solution that makes sense in today's market

- "Cloud-y", as with Google spreadsheets
    - Users can get to their models from anywhere
    - Users can share their models with other users
    - Users can feed their models with data available on the web
- Better than Google spreadsheets
    - Users can remotely provision infrastructure necessary to run their models

Features of a solution that makes sense in today's market

- Compositional, similar to Yahoo! Pipes we have
    - UI elements for "wiring" models up from existing models and pieces/parts of models
    - Business logic to support "wiring" as well as execution and analysis
    - Persistence layer to support storing and retrieving models
- Better than Yahoo! Pipes these elements also support
    - Manipulating partially completed executions, because models sometimes take a long time to run

Features of a solution that makes sense in today's market

- (Unduplicatably) rich set of existing models -- like CRAN
  - A library of existing well understood, high-value models
  - That keeps growing because users contribute back to it
- Better than CRAN, supporting analysis of ***models***, on the basis of
  - Behavior
  - Performance characteristics
  - Provenance
  - Allows users to contribute "provenanced" data sets, as well

Features of a solution that makes sense in today's market

- "Crowd-y", like SoftwareAG's AlignSpace, supporting "in situ" social networking technologies
    - Shared, collaborative model-making "space"
    - Structured conversations
- Better than AlignSpace
    - Collaborative decision making support
    - Agile methods

Challenges

- Given a really rich library of models, how will less experienced users find the ones they need?
- Given a really rich collection of data sets, how will less experienced users find the ones they need?

Challenges - finding models

- What are the factors that go into selecting a model?
  - Function/behavior
  - Provenance
  - Licensing
  - Scalability/performance
  - Security

Challenges - finding models

- What are the factors that go into selecting a data set?
  - Availability
  - Provenance
  - Cleanining/formatting/massaging

Proposal

- Build a best of breed PaaS
  - Take the best of Bespin, Pipes and GoogleAppEngine
  - Add functional programming model with full support for SELECT-FROM-WHERE
    - *Import rich model library*
- Include collaborative technologies *in context*
  - People want to and need make decisions in context
  - This creates a sticky experience
- Use this as the basis to introduce new search capability

commodity

still a commodity

not yet commodity

Proposal

- The Magritte PoC (as well as what we can surmise from release cycles of Pipes, Bespin, etc) showed that you can get to an alpha public release in a year with only a two person team,
    - approximately Beta public = 4 person years, but 18 calendar months
- The real high-cost, but exceptionally sticky value-add proposal is the rich model repository
    - You need to import 70% CRAN into a newer execution semantic
    - This can be done in partnership
    - cost is 24 calendar months, 4 FTE

Proposal - revenue model

- Tiered service
- Just as with S3/EC2 users pay on the basis of resource consumption which comes down to a cost for
  - bandwidth+storage+compute time
  - plus additional service fees which can also include consulting on everything from model design to optimization
- App store
  - Users can distribute components, providing an additional revenue stream

Market analysis

A view from the clouds: cloud-service ecosystem

Observations

- It is no longer possible to do software in isolation
- No one writes software de novo
- No one writes standalone, single-threaded applications anymore
- Cloud enables mobile; mobile enables cloud
- These facts are just as true of the scientific and HPC-computing communities as they are of ordinary software development

Observations

- In fact, because of the scaling requirements most offerings are facing, the lines between ordinary software development and HPC-computing are becoming blurred
- These trends re-enforce each other
- Therefore, failing some catastrophic events, we can predict the shape of a broad range of "engineering" environments
    - from software development ...
    - ... to scientific computing

Predictions

- They will provide a browser-based cloud-level development and deployment experience
- They will support social engagement and collaboration at every phase of the life-cycle
- They will provide provenance of data-sets as well as components/models/analytics

Verification

- Cloud-based engineering environments, PaaS are springing up all over the place
    - Yahoo Pipes, GoogleAppEngine, Mozilla Bespin, Popfly, ..., Stax, Wavemaker, ...,
    - AlignSpace
    - ... but also Cellucidate, KEGG, ...
- These fit into a larger ecosystem of S3/EC2 cloud services
- They also fit into a larger ecosystem of cloud-level dependency management such as maven or ivy

Mining the trends of data-mining

Observations

- Text-mining is now self-limiting; companies large and small are pushing solutions that increase the reach and applicability of simple text-mining, slowing the rate of development of richer mining capabilities
- Internet available text is dwarfed by the other searchable data sources on the internet
  - audio/visual
  - scientific
  - financial
  - code

Observations

- In the Scientific and High-Performance computing communities this has had a second-order effect
- to climb above the torrent of data these industries build models
- components/models/analytics are being developed at a rate proportional to the rate of growth of data capture and demand for data analysis

Predictions

- We should see an accumulation of these components/ models/analytics
- But, successful PaaS movement will inexorably lead to accelerated growth of models
- This will lead to a necessity for search of components/ models/analytics on a basis deeper than text

Verification

- Cloud-level repositories include
    - SourceForge, GoogleCodeBase, GitHub, RubyForge, ..., and eventually a MS offering
- But also
    - CRAN, KEGG, BioCyc, Reactome, ...
- We are already seeing search
    - Java.licio.us, Merobase, Hoogle
    - The so-called “semantic” web has at its heart an acknowledgement of a need to mine at a deeper level
- This is just the “free” stuff and doesn’t include all the stuff sitting behind the corporate firewall

Convergence of reality-based policies with reality-based business offerings

Observations

- The market is shifting because there are very immediate concerns
  - Energy
  - Environment
  - Health care
  - Social engagement on policy
  - Education

why isn't financial crisis on this list?

- Businesses will track these shifts and solutions vendors to these businesses need to track these shifts

Observations

- The solutions to these reality-based concerns involve technologies that fit hand-in-glove with the mining of richer kinds of data
  - Climate data
  - Geologic data
  - Biological data
  - Simulations from mechanical and chemical engineering...

Predictions

- We will begin to see a convergence of technological trends: the need for mining richer data will line up with and enable the ability to mine components/models/ analytics
- People will look for common patterns and abstractions across these different classes of search problems

Verification

- At the research level this has already happened
  - Oleg's LogicT
  - Monadic constraint-based programming
- The rest of this deck is aimed to show that it is happening at the industry level and the specific characteristics of the technology

Impact of programming model changes

Observations

But must provide one more aspect of context

- Programming model is under siege
- Concurrency from above
- Concurrency from below
- Stable, performant virtual machine technology

Predictions

- We should see a range of new programming models being offered with a few dominant players
- We should see an explosion of new programming language proposals to match the programming model proposals
  - Just like there is essentially one model behind Java, C#, ...
  - We will see multiple different languages realizing new models

Verification

- Programming models on the table
  - Functional
  - Message-passing-based concurrency
  - Exotic concurrency
- Language proposals
  - Scala, F#, CAL, Haskell, OCaml, ..., Clojure, Fortress
  - Erlang
  - Resurgence of interest in Lisp, Scheme

Verification

- Dragged into the limelight along with these are
  - Ruby
  - R
- These have functional characteristics, but essentially single-threaded semantics
- Dynamic semantics essentially bad solutions to reasonable problems
  - Will always have worse performance
  - Will not have scalable development characteristics
  - Will not align well with search capabilities

Reasonable search and seizure

Executive summary




- Industry has begun to recognize that
  - An Entity, whether you call it a POJO or a Row or a Record or ... is best described by functional types
  - The SELECT-FROM-WHERE construct is much richer than originally understood

W3C World Wide Web Consortium

Google

Microsoft

ORACLE

Executive summary

- This has implications in terms of
    - The range of stores you can support
    - The range of control regimes you can support
    - The configurability, reliability, availability and deployment options you can support

Questions?