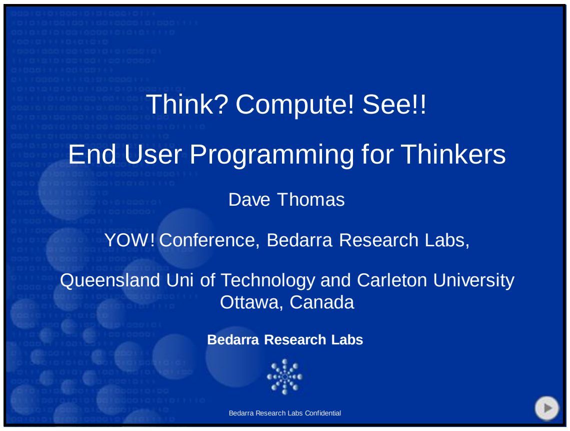


Think? Compute! See!!

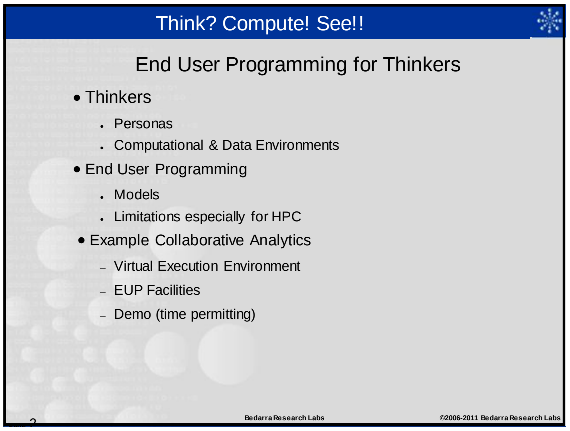
End User Programming for Thinkers

Dave Thomas

YOW! Conference, Bedarra Research Labs,

Queensland Uni of Technology and Carleton University
Ottawa, Canada

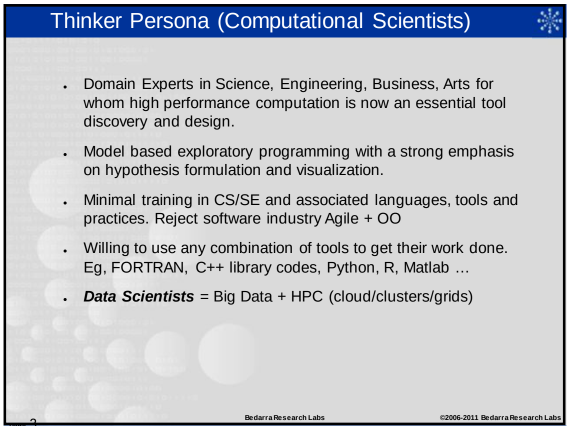
Bedarra Research Labs



Think? Compute! See!!

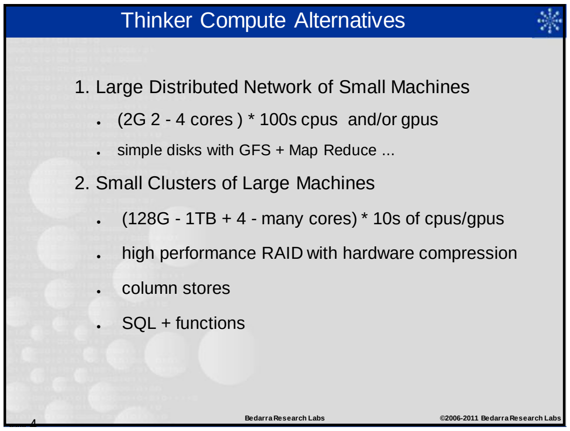
End User Programming for Thinkers

- Thinkers
  - Personas
  - Computational & Data Environments
- End User Programming
  - Models
  - Limitations especially for HPC
- Example Collaborative Analytics
  - Virtual Execution Environment
  - EUP Facilities
  - Demo (time permitting)



Thinker Persona (Computational Scientists)

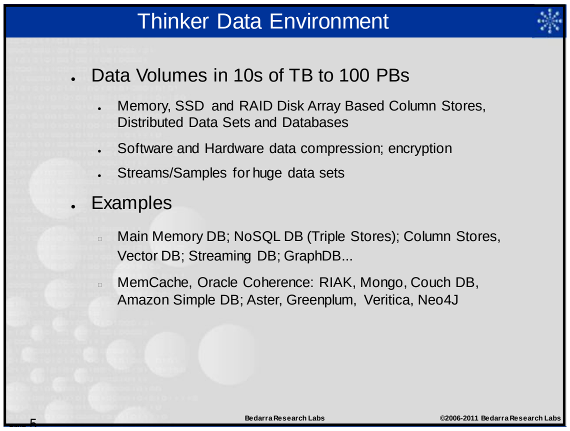
- Domain Experts in Science, Engineering, Business, Arts for whom high performance computation is now an essential tool discovery and design.
- Model based exploratory programming with a strong emphasis on hypothesis formulation and visualization.
- Minimal training in CS/SE and associated languages, tools and practices. Reject software industry Agile + OO
- Willing to use any combination of tools to get their work done. Eg, FORTRAN, C++ library codes, Python, R, Matlab …
- ***Data Scientists*** = Big Data + HPC (cloud/clusters/grids)



Thinker Compute Alternatives

1. Large Distributed Network of Small Machines
   - (2G 2 - 4 cores) * 100s cpus and/or gpus
   - simple disks with GFS + Map Reduce ...
2. Small Clusters of Large Machines
   - (128G - 1TB + 4 - many cores) * 10s of cpus/gpus
   - high performance RAID with hardware compression
   - column stores
   - SQL + functions

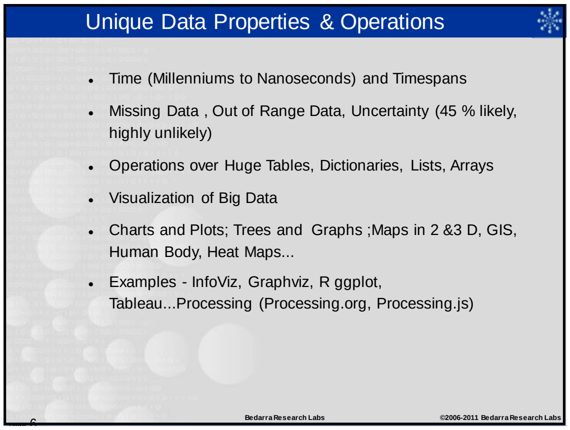
4 Bedarra Research Labs ©2006-2011 Bedarra Research Labs



Thinker Data Environment

- Data Volumes in 10s of TB to 100 PBs
  - Memory, SSD and RAID Disk Array Based Column Stores, Distributed Data Sets and Databases
  - Software and Hardware data compression; encryption
  - Streams/Samples for huge data sets
- Examples
  - Main Memory DB; NoSQL DB (Triple Stores); Column Stores, Vector DB; Streaming DB; GraphDB...
  - MemCache, Oracle Coherence: RIAK, Mongo, Couch DB, Amazon Simple DB; Aster, Greenplum, Veritica, Neo4J

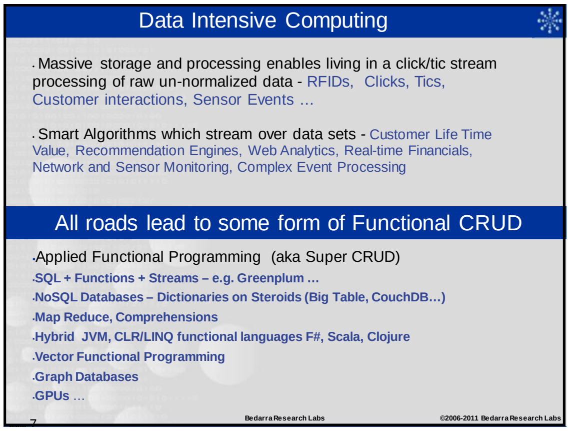
5 Bedarra Research Labs ©2006-2011 Bedarra Research Labs



Unique Data Properties & Operations

- Time (Millenniums to Nanoseconds) and Timespans
- Missing Data , Out of Range Data, Uncertainty (45 % likely, highly unlikely)
- Operations over Huge Tables, Dictionaries, Lists, Arrays
- Visualization of Big Data
- Charts and Plots; Trees and Graphs ;Maps in 2 &3 D, GIS, Human Body, Heat Maps...
- Examples - InfoViz, Graphviz, R ggplot, Tableau...Processing (Processing.org, Processing.js)

6 Bedarra Research Labs ©2006-2011 Bedarra Research Labs



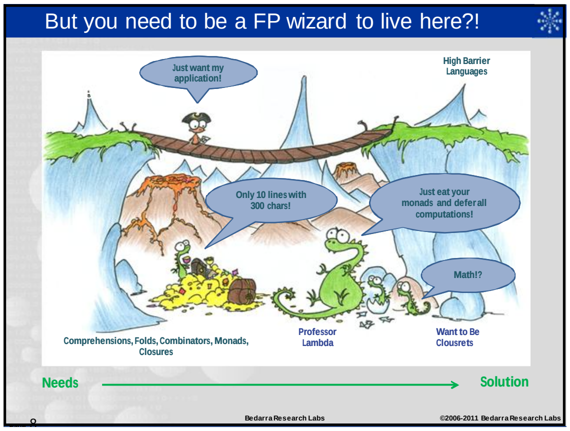
Data Intensive Computing

- Massive storage and processing enables living in a click/tic stream processing of raw un-normalized data - RFIDs, Clicks, Tics, Customer interactions, Sensor Events …

- Smart Algorithms which stream over data sets - Customer Life Time Value, Recommendation Engines, Web Analytics, Real-time Financials, Network and Sensor Monitoring, Complex Event Processing

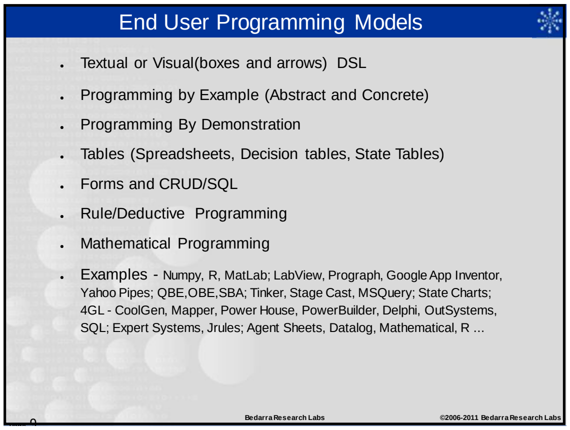
All roads lead to some form of Functional CRUD

- Applied Functional Programming (aka Super CRUD)
- SQL + Functions + Streams – e.g. Greenplum …
- NoSQL Databases – Dictionaries on Steroids (Big Table, CouchDB…)
- Map Reduce, Comprehensions
- Hybrid JVM, CLR/LINQ functional languages F#, Scala, Clojure
- Vector Functional Programming
- Graph Databases
- GPUs …



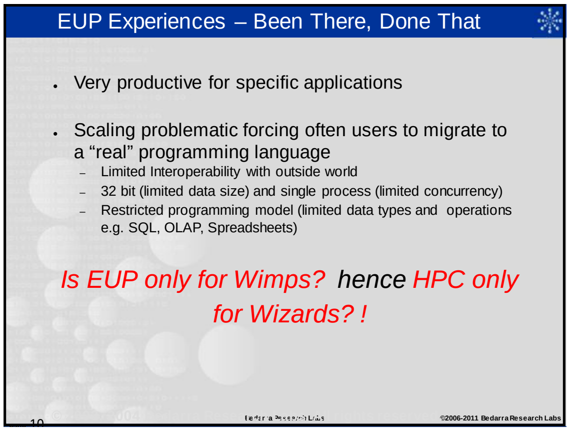
But you need to be a FP wizard to live here?!

Just want my application!
High Barrier Languages
Only 10 lines with 300 chars!
Just eat your monads and defer all computations!
Math!?
Comprehensions, Folds, Combinators, Monads, Closures
Professor Lambda
Want to Be Clousrets
Needs
Solution



End User Programming Models

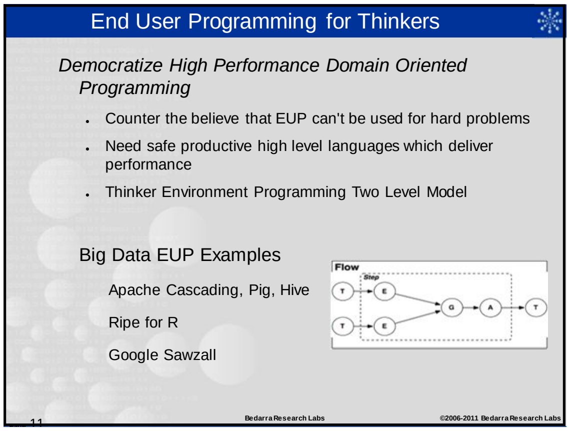
- Textual or Visual(boxes and arrows) DSL
- Programming by Example (Abstract and Concrete)
- Programming By Demonstration
- Tables (Spreadsheets, Decision tables, State Tables)
- Forms and CRUD/SQL
- Rule/Deductive Programming
- Mathematical Programming
- Examples - Numpy, R, MatLab; LabView, Prograph, Google App Inventor, Yahoo Pipes; QBE,OBE,SBA; Tinker, Stage Cast, MSQuery; State Charts; 4GL - CoolGen, Mapper, Power House, PowerBuilder, Delphi, OutSystems, SQL; Expert Systems, Jrules; Agent Sheets, Datalog, Mathematical, R …



EUP Experiences – Been There, Done That

- Very productive for specific applications
- Scaling problematic forcing often users to migrate to a “real” programming language
  - Limited Interoperability with outside world
  - 32 bit (limited data size) and single process (limited concurrency)
  - Restricted programming model (limited data types and operations e.g. SQL, OLAP, Spreadsheets)

Is EUP only for Wimps? hence HPC only for Wizards? !



End User Programming for Thinkers

Democratize High Performance Domain Oriented Programming

- Counter the believe that EUP can't be used for hard problems
- Need safe productive high level languages which deliver performance
- Thinker Environment Programming Two Level Model

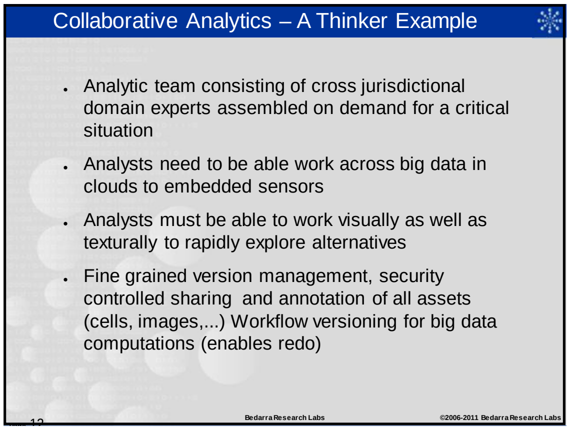
Big Data EUP Examples

Apache Cascading, Pig, Hive

Ripe for R

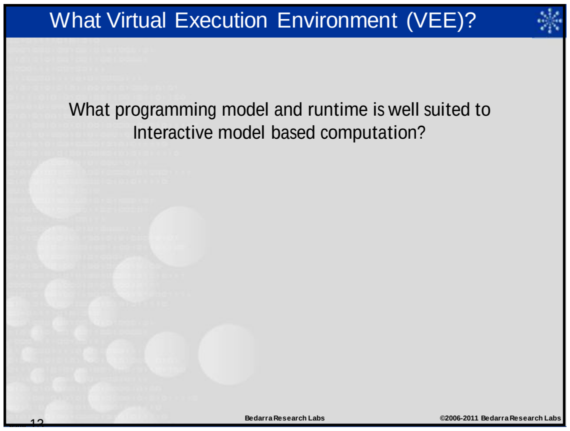
Google Sawzall

Flow
Step
T
E
T
E
G
A
T



Collaborative Analytics – A Thinker Example

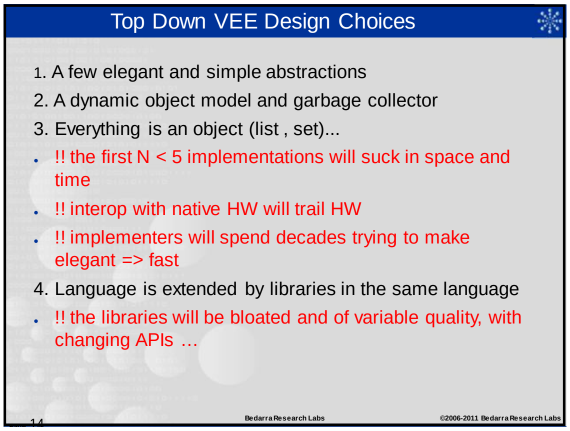
- Analytic team consisting of cross jurisdictional domain experts assembled on demand for a critical situation
- Analysts need to be able work across big data in clouds to embedded sensors
- Analysts must be able to work visually as well as texturally to rapidly explore alternatives
- Fine grained version management, security controlled sharing and annotation of all assets (cells, images,...) Workflow versioning for big data computations (enables redo)



What Virtual Execution Environment (VEE)?

What programming model and runtime is well suited to Interactive model based computation?

Bedarra Research Labs ©2006-2011 Bedarra Research Labs

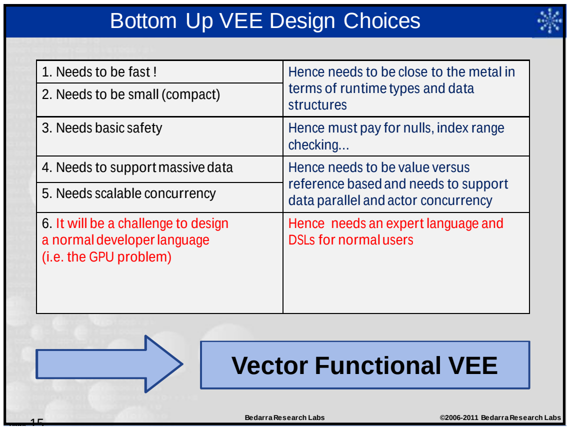


Top Down VEE Design Choices

1. A few elegant and simple abstractions
2. A dynamic object model and garbage collector
3. Everything is an object (list , set)...
- !! the first N < 5 implementations will suck in space and time
- !! interop with native HW will trail HW
- !! implementers will spend decades trying to make elegant => fast

4. Language is extended by libraries in the same language
- !! the libraries will be bloated and of variable quality, with changing APIs …

Bedarra Research Labs ©2006-2011 Bedarra Research Labs

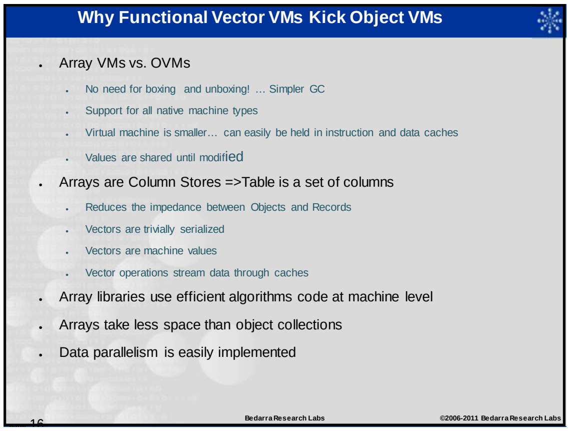


Bottom Up VEE Design Choices

1. Needs to be fast !	Hence needs to be close to the metal in terms of runtime types and data structures
2. Needs to be small (compact)	
3. Needs basic safety	Hence must pay for nulls, index range checking...
4. Needs to support massive data	Hence needs to be value versus reference based and needs to support data parallel and actor concurrency
5. Needs scalable concurrency	
6. It will be a challenge to design a normal developer language (i.e. the GPU problem)	Hence needs an expert language and DSLs for normal users

Vector Functional VEE

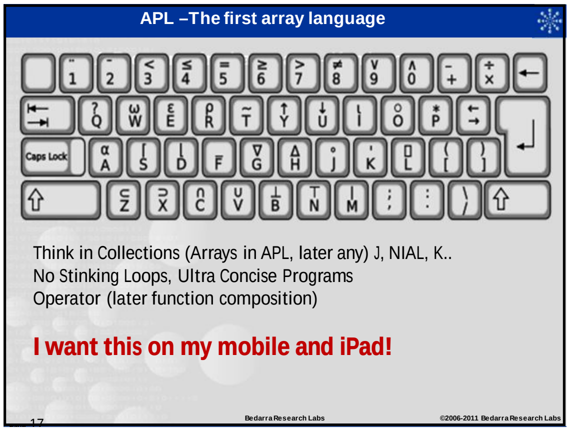
Bedarra Research Labs ©2006-2011 Bedarra Research Labs



Why Functional Vector VMs Kick Object VMs

- Array VMs vs. OVMs
  - No need for boxing and unboxing! … Simpler GC
  - Support for all native machine types
  - Virtual machine is smaller… can easily be held in instruction and data caches
  - Values are shared until modified
- Arrays are Column Stores =>Table is a set of columns
  - Reduces the impedance between Objects and Records
  - Vectors are trivially serialized
  - Vectors are machine values
  - Vector operations stream data through caches
- Array libraries use efficient algorithms code at machine level
- Arrays take less space than object collections
- Data parallelism is easily implemented

16 Bedarra Research Labs ©2006-2011 Bedarra Research Labs

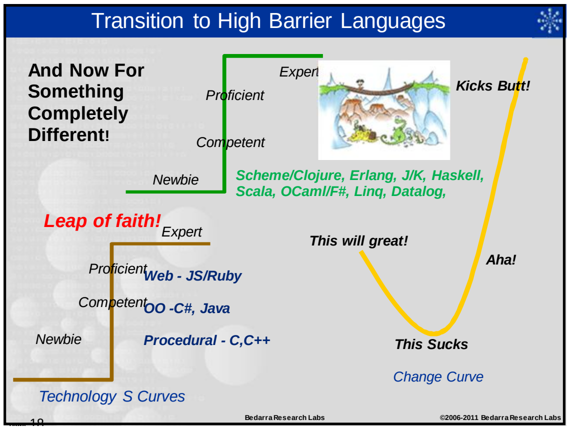


APL –The first array language

Think in Collections (Arrays in APL, later any) J, NIAL, K..
No Stinking Loops, Ultra Concise Programs
Operator (later function composition)

I want this on my mobile and iPad!

17 Bedarra Research Labs ©2006-2011 Bedarra Research Labs



Transition to High Barrier Languages

And Now For Something Completely Different!

Expert
Proficient
Competent
Newbie

Scheme/Clojure, Erlang, J/K, Haskell, Scala, OCaml/F#, Linq, Datalog,

Kicks Butt!

Leap of faith!

Expert
Proficient
Competent
Newbie

Web - JS/Ruby
OO -C#, Java
Procedural - C,C++

Technology S Curves

This will great!
Aha!
This Sucks

Change Curve

18 Bedarra Research Labs ©2006-2011 Bedarra Research Labs

CARE Analyst and Expert Programming Models

- 50+:1 of ratio of analysts to expert developers
- Experts surface new functionality to analysts as DSL library extensions

Analysts Application Programming Model

- Wide Spectrum DSLs (SQL, Sheets & Tables, Boxes and Arrows, ...)
- Narrow Domain Specific DSL (IP packets, finance, geographic, cultural ...)
- Leverage existing standards and user models
- Interoperable with R, MatLab, MS Office...

New Functionality

Expert Programming Model

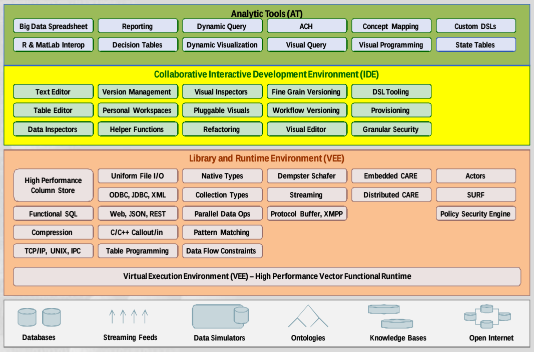
- Wide Spectrum Functional Vector Language
- Full Interoperable with current technologies: ODBC, Java, C#, C++, Web

CARE Core Platform

,column store, core, Core Libraries, platform interop ....

- Virtual Execution Environment (VEE) – High Performance Vector Functional Runtime

CARE - An Exploratory Colaborative Analytics Environment



Analytic Tools (AT)

Big Data Spreadsheet, Reporting, Dynamic Query, ACH, Concept Mapping, Custom DSLs
R & MatLab Interop, Decision Tables, Dynamic Visualization, Visual Query, Visual Programming, State Tables

Collaborative Interactive Development Environment (IDE)

Text Editor, Version Management, Visual Inspectors, Fine Grain Versioning, DSL Tooling
Table Editor, Personal Workspaces, Pluggable Visuals, Workflow Versioning, Provisioning
Data Inspectors, Helper Functions, Refactoring, Visual Editor, Granular Security

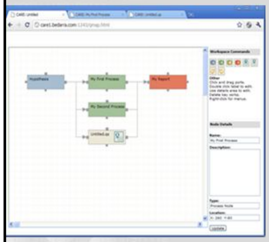
Library and Runtime Environment (VEE)

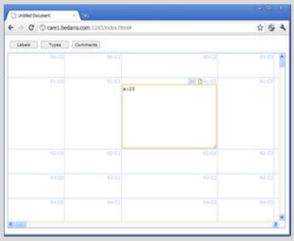
High Performance Column Store, Uniform File I/O, Native Types, Dempster Schafer, Embedded CARE, Actors
ODBC, JDBC, XML, Collection Types, Streaming, Distributed CARE, SURF
Functional SQL, Web, JSON, REST, Parallel Data Ops, Protocol Buffer, XMPP, Policy Security Engine
Compression, C/C++ Callout/in, Pattern Matching
TCP/IP, UNIX, IPC, Table Programming, Data Flow Constraints

Virtual Execution Environment (VEE) – High Performance Vector Functional Runtime

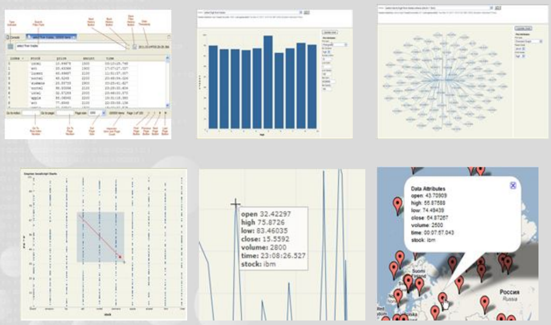
Databases, Streaming Feeds, Data Simulators, Ontologies, Knowledge Bases, Open Internet

Analyst Visual IDE Concept Maps and Sheets





Analyst IDE - Visual Inspection and Refinement



open 32.42297
high 75.8726
low: 83.46035
close: 15.5592
volume: 2800
time: 23:08:26.527
stock: ibm

Data Attributes
open: 43.70909
high: 55.87588
low: 74.49439
close: 64.87267
volume: 2500
time: 00:07:57.043
stock: ibm

Bedarra Research Labs

Demo

Bedarra Research Labs

Thanks!

Bedarra Research Labs