Solving Problems with High Performance Computing

LexisNexis®

Risk Solutions

What is the HPCC Platform?

Single Platform and supporting tools built specifically for

Data Intensive Computing and Big Data Delivery.

It’s been supporting enterprises for more than a decade.

It’s now open source and accessible to all.

HPCC Architecture

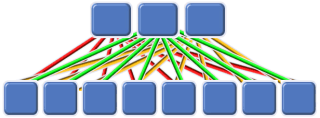
- HPCC is based on a distributed shared-nothing architecture, and uses commodity PC servers
- HPCC uses a consolidated central network switch for best for performance, but can use a decentralized network of satellite switches. Computing efficiencies in HPCC substantially reduce the number of required nodes and, hence, network ports compared to other distributed platforms like Hadoop.
- HPCC Distributed File System is record oriented (supporting fixed length, variable length field delimited and XML records)
- HPCC DFS is tightly coupled with the processing layer, exploiting data locality to a higher degree to minimize network transfers and maximize throughput
- HPCC applications are built using ECL, a declarative data flow oriented language supporting the automation of algorithm parallelization and work distribution

Three Main Components

❶ **HPCC Data Refinery (Thor)**

- Massively Parallel Extract Transform and Load (ETL) engine
  - Built from the ground up as a parallel data environment. Leverages inexpensive locally attached storage. Doesn't require a SAN infrastructure.
- Enables data integration on a scale not previously available:
  - Current LexisNexis person data build process generates 350 Billion intermediate results at peak
- Suitable for:
  - Massive joins/merges
  - Massive sorts & transformations
  - Any N^2 problem
  - *"identify and catalog all the DNA in the oceans"*

❷ **HPCC Data Delivery Engine (Roxie)**



- A massively parallel, high throughput, structured query response engine
- Ultra fast due to its read-only nature.
- Allows indices to be built onto data for efficient multi-user retrieval of data
- Suitable for

  Volumes of structured queries

  Full text ranked Boolean search

  "I want that fish there"

❸ **Enterprise Control Language (ECL)**

- An easy to use , data-centric programming language optimized for large-scale data management and query processing
- Highly efficient; Automatically distributes workload across all nodes.
  - Benchmark against SQL **(5 times more efficient)** for code generation
- Automatic parallelization and synchronization of sequential algorithms for parallel and distributed processing
- Large library of efficient modules to handle common data manipulation tasks

The HPCC Platform Delivers Benefits in Speed, Capabilities and Capacity

Speed

- Scales to extreme workloads quickly and easily
- **Increased development speeds - faster delivery w/ fewer resources**
- Improved developer productivity

Capabilities

- Enables massive joins, merges, sorts, transformations or tough O(n^2) problems
- Increases project responsiveness
- Accelerates creation of new research via rapid prototyping capabilities
- Offers a platform for collaboration and innovation leading to better results

Capacity

- Commodity hardware and fewer people can do much more in less time
- Uses IT resources efficiently via sharing and higher system utilization

Scalability: Little Degradation in Performance

Scalability

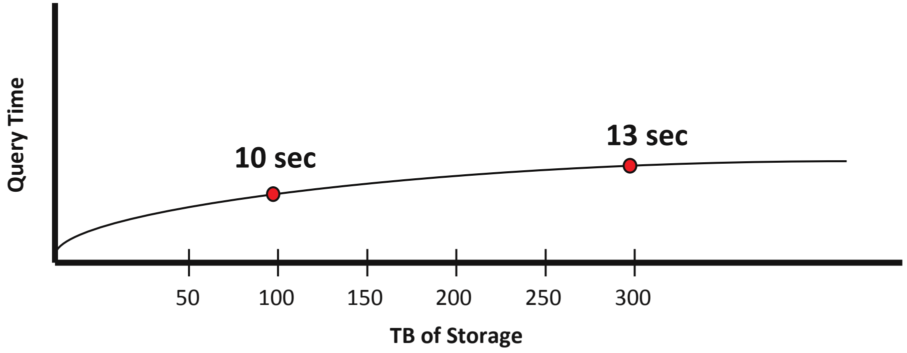
- Scales to support 1000+TB (up to petabytes) of data
- Purposely built system to do massive I/O
- Rapidly performs complex queries on structured and unstructured data to link to a variety of data sources
- Suitable for massive joins/mergers, beyond limits of relational DBs
- Scale increases with the addition of low-cost, commodity servers

HPCC – in production

- Current production systems range from 20 to 2000 nodes
- Currently supports over 150,000 customers, millions of end users
- Currently handling over 20 million transactions per day for our online and batch products. and innovation leading to better results

Complex query example demonstrates below

- Transaction latencies increase logarithmically while data sizes grow linearly



Enterprise Control Language (ECL)

Enterprise Control Language (ECL) is a programming language developed over 10 years by a group of ex-Borland compiler writers.

Designed specifically for algorithmic development on top of large data sets.

80% more efficient than C++, Java and SQL and 1/3 reduction in programmer time to maintain/enhance existing applications

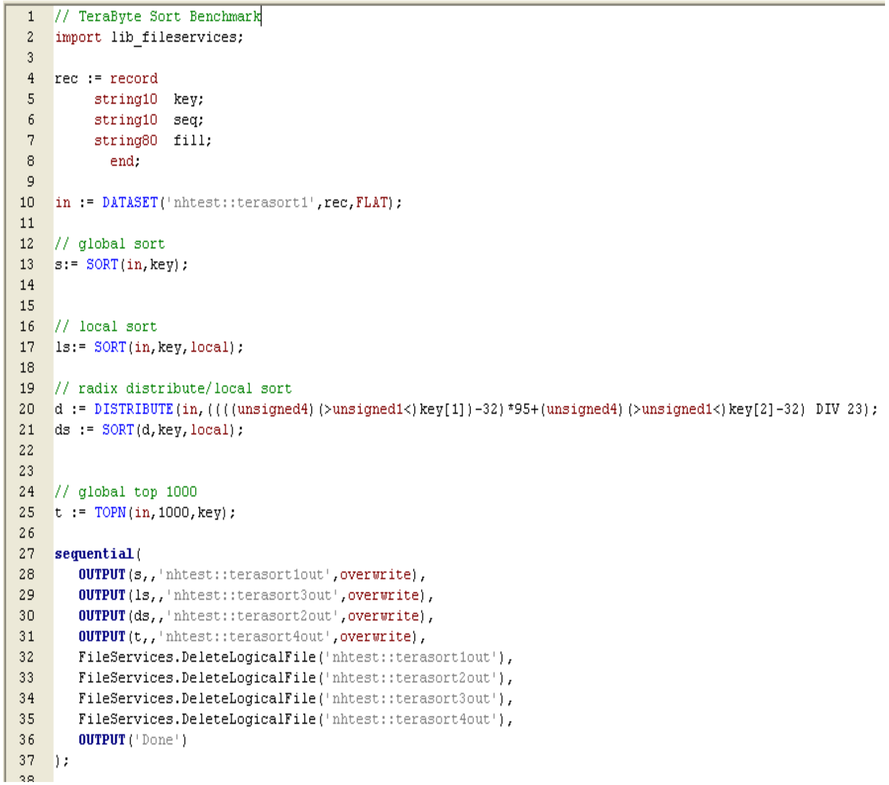
Functional language optimized for large-scale data processing at scale.

Abstracts the underlying platform configuration from the algorithm development allowing efficient, expressive data manipulation

Automatic parallelization and optimization of sequential algorithms for parallel and distributed processing (i.e., the programmer does not need to understand how to manage the parallel processing environment)

Large library of efficient modules to handle common data manipulation tasks

Inbuilt data primitives (sort, distribute, join) highly optimized for distributed data processing tasks



```
// TeraByte Sort Benchmark
import lib_fileservices;

rec := record
     string10  key;
     string10  seq;
     string80  fill;
       end;

in := DATASET('nhtest::terasort1',rec,FLAT);

// global sort
s:= SORT(in,key);


// local sort
ls:= SORT(in,key,local);

// radix distribute/local sort
d := DISTRIBUTE(in,((((unsigned4)(>unsigned1<)key[1])-32)*95+(unsigned4)(>unsigned1<)key[2]-32) DIV 23);
ds := SORT(d,key,local);


// global top 1000
t := TOPN(in,1000,key);

sequential(
   OUTPUT(s,,'nhtest::terasort1out',overwrite),
   OUTPUT(ls,,'nhtest::terasort3out',overwrite),
   OUTPUT(ds,,'nhtest::terasort2out',overwrite),
   OUTPUT(t,,'nhtest::terasort4out',overwrite),
   FileServices.DeleteLogicalFile('nhtest::terasort1out'),
   FileServices.DeleteLogicalFile('nhtest::terasort2out'),
   FileServices.DeleteLogicalFile('nhtest::terasort3out'),
   FileServices.DeleteLogicalFile('nhtest::terasort4out'),
   OUTPUT('Done')
);
```

HPCC Platform is now Open Source

HPCC Systems: http://hpccsystems.com

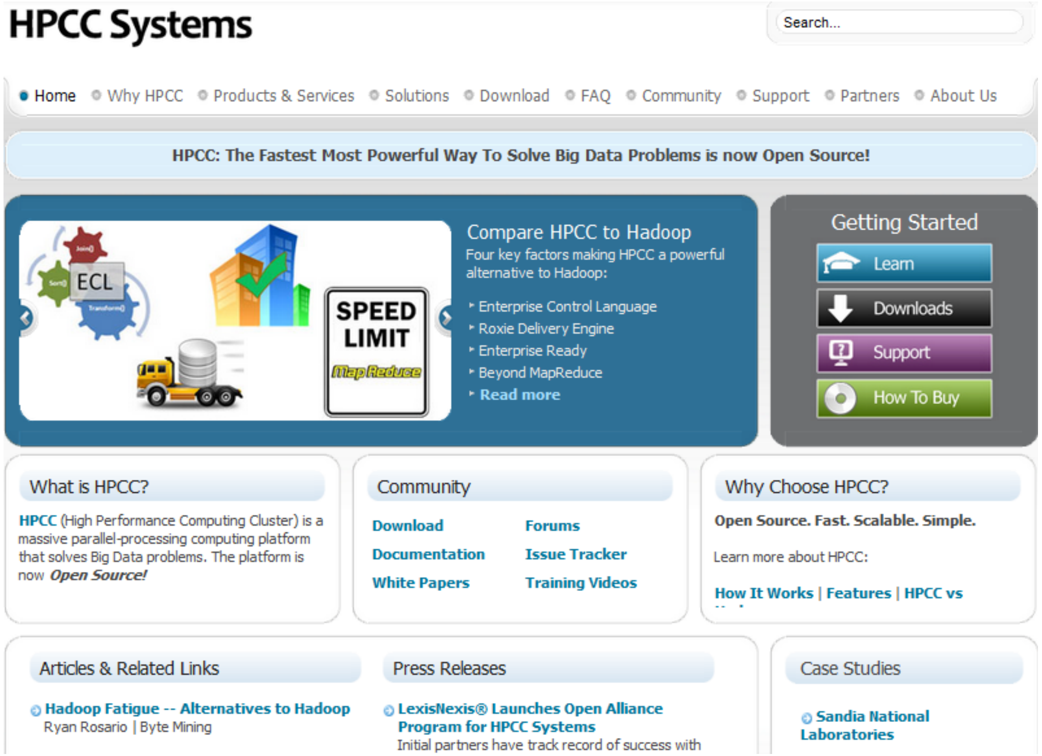
Comparison with Hadoop:

http://hpccsystems.com/Why-HPCC/HPCC-vs-Hadoop

Downloads: http://hpccsystems.com/download

Academic References:

http://hpccsystems.com/community/academic/references



HPCC: The Fastest Most Powerful Way To Solve Big Data Problems is now Open Source!

Compare HPCC to Hadoop

Four key factors making HPCC a powerful alternative to Hadoop:

- Enterprise Control Language
- Roxie Delivery Engine
- Enterprise Ready
- Beyond MapReduce
- Read more

What is HPCC?

HPCC (High Performance Computing Cluster) is a massive parallel-processing computing platform that solves Big Data problems. The platform is now *Open Source!*

Community

Download	Forums
Documentation	Issue Tracker
White Papers	Training Videos

Why Choose HPCC?

Open Source. Fast. Scalable. Simple.

Learn more about HPCC:

How It Works | Features | HPCC vs

Articles & Related Links

Hadoop Fatigue -- Alternatives to Hadoop
Ryan Rosario | Byte Mining

Press Releases

LexisNexis® Launches Open Alliance Program for HPCC Systems
Initial partners have track record of success with

Case Studies

Sandia National Laboratories

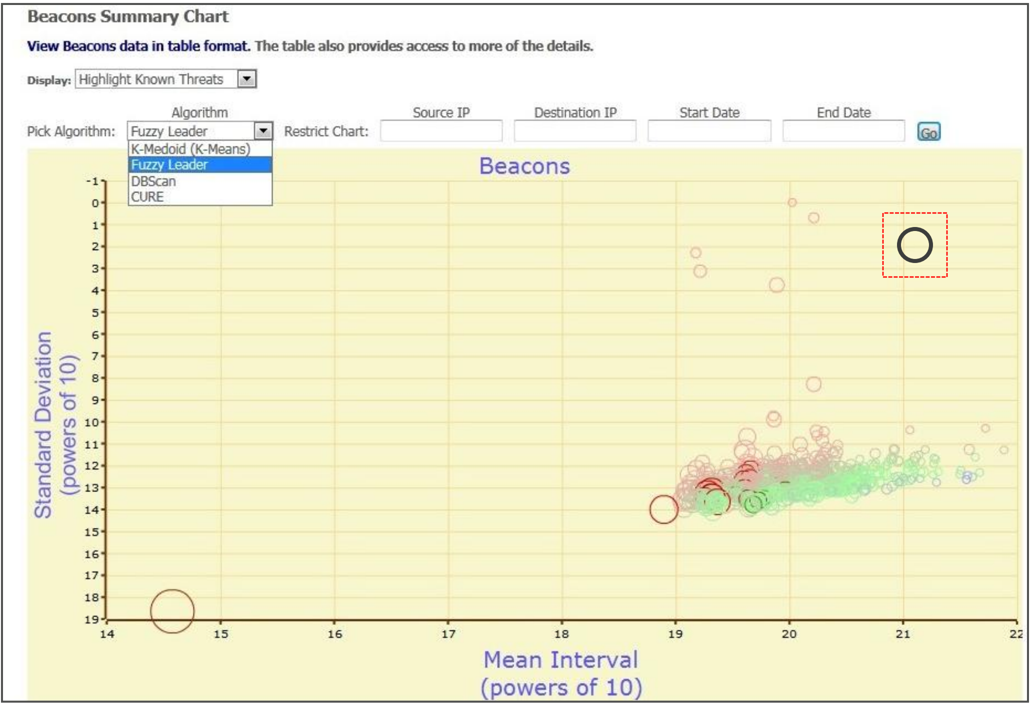
HPCC Systems Case Studies

Case Example 1: Network Traffic Analysis in Seconds

Scenario

Conventional network sensor and monitoring solutions are constrained by inability to quickly ingest massive data volumes for analysis

- 15 minutes of network traffic can generate 4 Terabytes of data, which can take 6 hours to process
- 90 days of network traffic can add up to 300+ Terabytes



Beacons Summary Chart

View Beacons data in table format. The table also provides access to more of the details.

Display: Highlight Known Threats

Pick Algorithm: Fuzzy Leader

Restrict Chart: Source IP, Destination IP, Start Date, End Date

Task

Drill into all the data to see if any US government systems have communicated with any suspect systems of foreign organizations in the last 6 months

- In this scenario, we look specifically for traffic occurring at unusual hours of the day

Result

In seconds, the HPCC sorted through months of network traffic to identify patterns and suspicious behavior

Case Example 2: ChoicePoint Migration

Scenario

Reed Elsevier buys ChoicePoint for $4.1 billion and integrates it into LexisNexis Risk Solutions

- Spun off from Equifax, Inc. in 1997, ChoicePoint grew quickly through acquisition.
- ChoicePoint kept its business decentralized and individual units operated different – via duplicative, technology platforms.
- Its largest division, Insurance, operated on an expensive and outdated IBM mainframe-based architecture, while its public records data center used an old, inflexible relational database built off an Oracle/Webmethods, Sun Microsystems-based architecture.
- ChoicePoint struggled under high information technology (IT) management costs and its growth was constricted due to its inability to adapt its technology and develop new solutions to meet evolving business needs.

Task

Migrate ChoicePoint, including the Insurance business to LexisNexis HPCC platform without disrupting business

Result

- LexisNexis anticipates 70% reduction in new product time-to-market as a result of moving ChoicePoint to HPCC platform.
- 2 years into the integration, LexisNexis is on track to generate substantial savings within the first 5 years.
- LexisNexis has seen an immediate 8-10% improvement in hit rates as a function of better matching and linking technology, which is a derivative of the HPCC technology
- The response time of interactive transactions for certain products has improved by a factor of 200% on average, simply as a function of moving to the HPCC.
- One HPCC (one infrastructure) translates into less people maintaining systems.

Case Example 3 Suspicious Equity Stripping Cluster

Scenario

Is mortgage fraud just an activity of isolated individuals, or an industry?

Task

Rank the nature, connectedness, and proximity of suspicious property transactions for every identity in the U.S.

Result

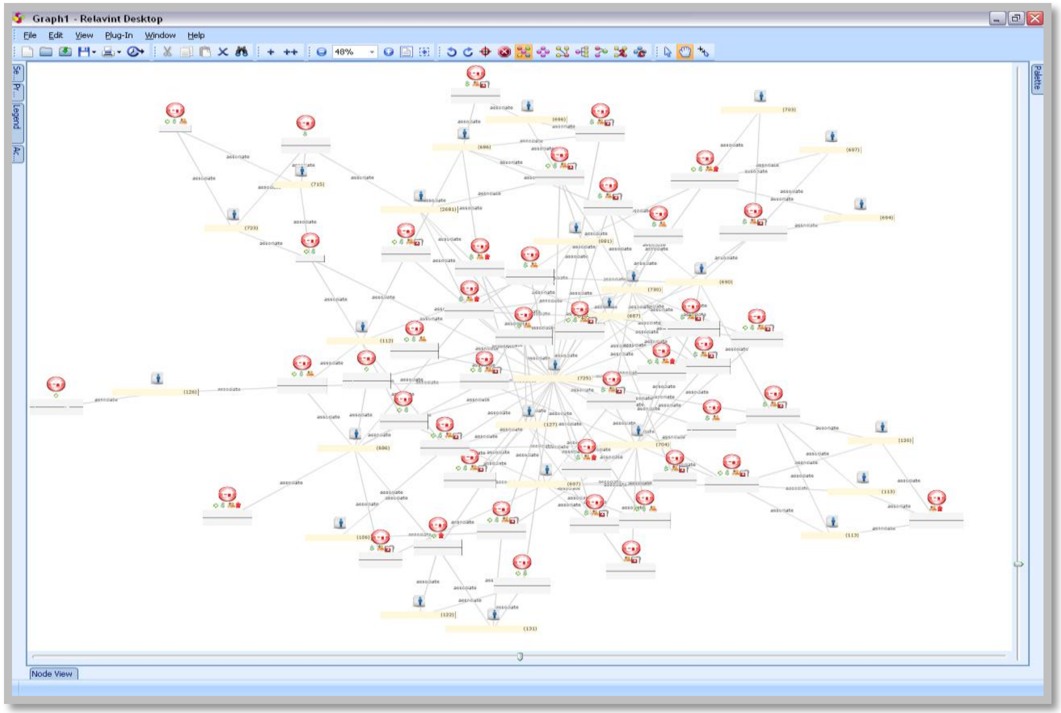
Florida Proof of Concept

Highest ranked influencers

- Identified known ringleaders in flipping and equity stripping schemes.
- Typically not connected directly to suspicious transactions.

Clusters with high levels of potential collusion.

Clusters offloading property, generating defaults



Case Example 4: Insurance Scoring, From 100 Days to 30 Minutes

Scenario

One of the top 3 insurance providers using Oracle analytics products on multiple statistical model platforms with disparate technologies

- Insurer issues request to re-run all past reports for a customer: 11.5M reports since 2004
- Using their current technology infrastructure it would take 100 days to run these 11.5M reports

Task

Migration of 75 different models, 74,000 lines of code and approximately 700 definitions to the HPCC

- Models were migrated in 1 man month.
- Using a small development system (and only one developer), we ran 11.5 million reports in 66 minutes
- Performance on a production-size system: 30 minutes
- Testing demonstrated our ability to work in batch or online

Result

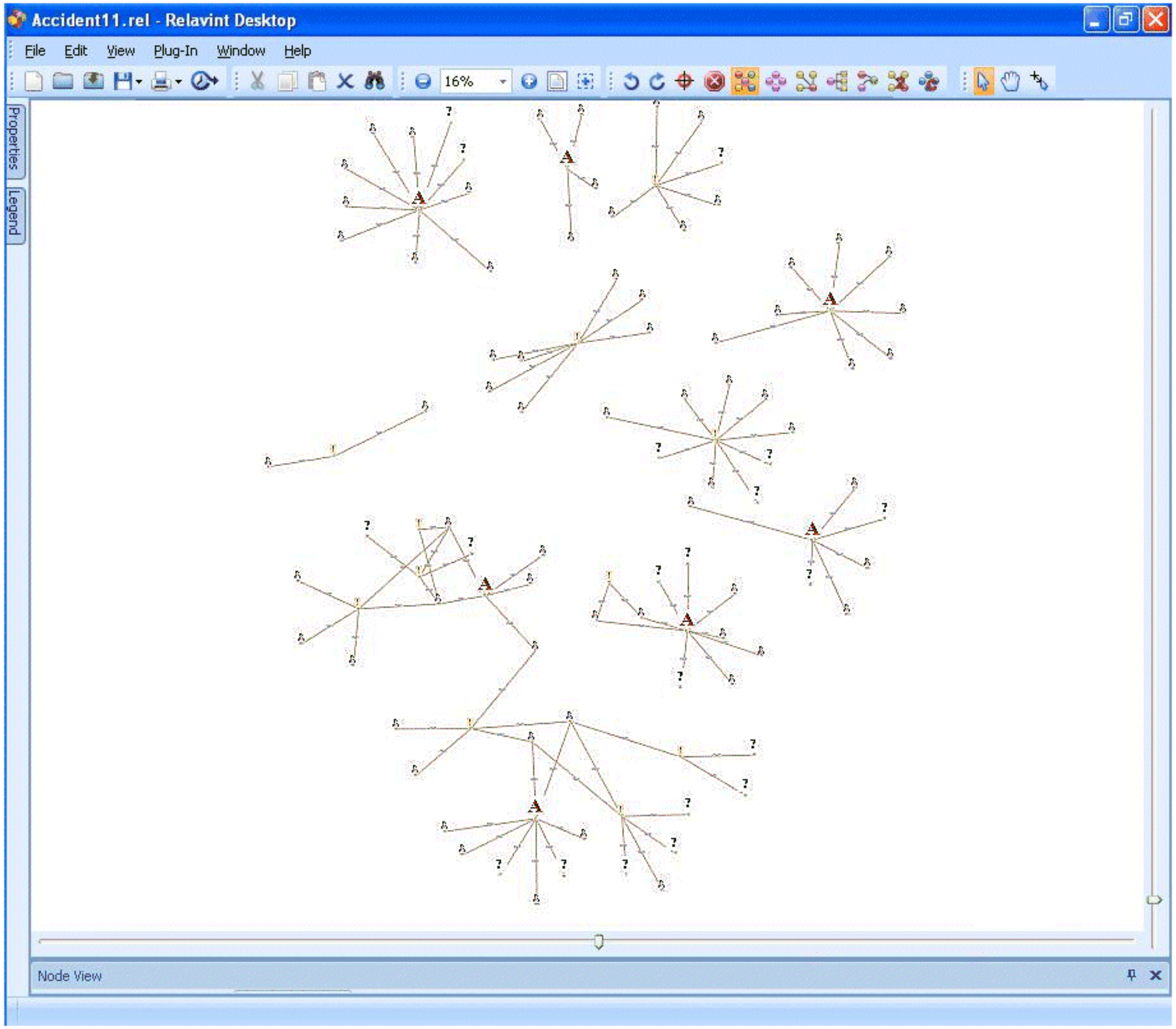
- Reduced manual work, increases reliability, and created capability to do new scores
- Decreased development time from 1 year to several weeks; decreased run time from 100 days to 30 minutes
- One HPCC (one infrastructure) translates into less people maintaining systems.

Case Example 5: Detecting Insurance Collusion in Louisiana

Scenario

This view of carrier data shows seven known fraud claims and an additional linked claim.

The Insurance company data **only finds a connection between two of the seven claims**, and only identified one other claim as being weakly connected.


Accident11.rel - Relavint Desktop
File Edit View Plug-In Window Help
16%
Properties
Legend
Node View

Case Example 5: Detecting Insurance Collusion in Louisiana

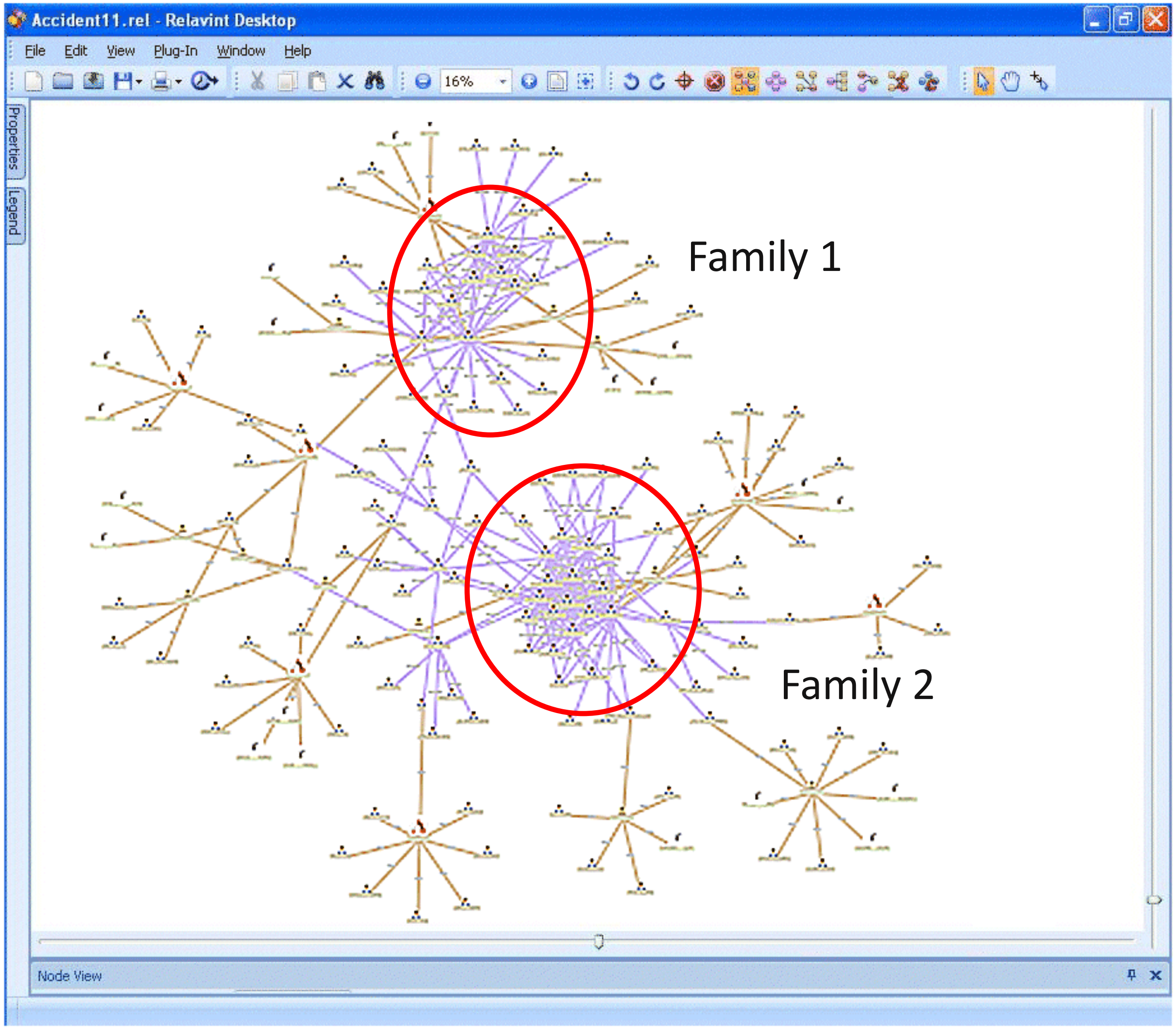
Task

After adding the Link ID to the carrier Data, LexisNexis HPCC technology then added 2 additional degrees of relative

Result

The results showed **two family groups interconnected on all of these seven claims.**

The links were much stronger than the carrier data previously supported.



Family 1

Family 2

On the Horizon

- Just Announced Amazon Support
  - Additional Amazon integration to come
- Eclipse Plug-in – Additional support for the Mac coming
- Machine Learning Libraries
  - Currently building / testing
  - Coded as a consistent library
  - Runs across multiple nodes
  - Coded in a high level data language

How we can help you

The Platform

- One proven open-sourced fully-integrated platform designed for data intensive computing and data delivery
    - Enables complex data intensive algorithms
    - One tool from ingest to delivery
    - Faster solution development and time to publish

HPCC Systems

- Training and other support-in-kind for certain academic projects (just ask!)
- Classroom materials and collaboration with academic community
- Student Internships