ScienceDirect's Advanced Recommender
a Fruitful Academic—Industrial Partnership

Martin Rajman
EPFL
Martin.Rajman@epfl.ch

Craig Scott
Elsevier
C.Scott@elsevier.com

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

ELSEVIER

Academic-Industrial partnership

Martin Rajman

- Executive Director of Nano-Tera.ch, a large Swiss Research Program funding collaborative multi-disciplinary projects for the engineering of complex systems in Health and the Environment.
- Senior researcher at EPF Lausanne, Switzerland (EPFL). His research interests include Artificial Intelligence, Computational Linguistics and Data-driven Probabilistic Machine Learning.
- Active in various large scale industry-research collaborations with majors economic players.

EPFL

EPFL
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

- EPFL is one of the two Swiss Federal Institutes of Technology and is located in Lausanne, Switzerland. It is considered to be among the worlds most prestigious universities in technology.

Academic-Industrial partnership

Craig Scott

- Senior Product Manager, Academic & Government Research Markets (ScienceDirect, Scopus, Scirus)
- Amsterdam, Netherlands
- Working on ‘big data’ experiments, A/B testing, search technology

Elsevier

- Science Technology & Medical division of Reed Elsevier
  Customers in >180 countries (1/3 North America, 1/3 Europe, 1/3 RoW)
  Serving >30 million scientists, students, health and information professionals

General Context

- Every day Elsevier records millions of data points on document downloads (or views) on about 12 million scientific publications
- This represents extremely valuable information about the involved scientific community(ies)
- This information is currently underexploited...

Aims

- Exploit the value present in huge amounts of usage/click stream data
- Leverage HPCC capabilities and LN RISK Private Cloud infrastructure
- Embed the use of performance metrics in product development decisions within an experimental, iterative and data driven approach

Concrete Goals

- Implementing an article recommender to improve the performance of the former “Related Articles” technology present in ScienceDirect article pages
- Produce comparative performance metrics via A/B testing techniques to validate product development choices

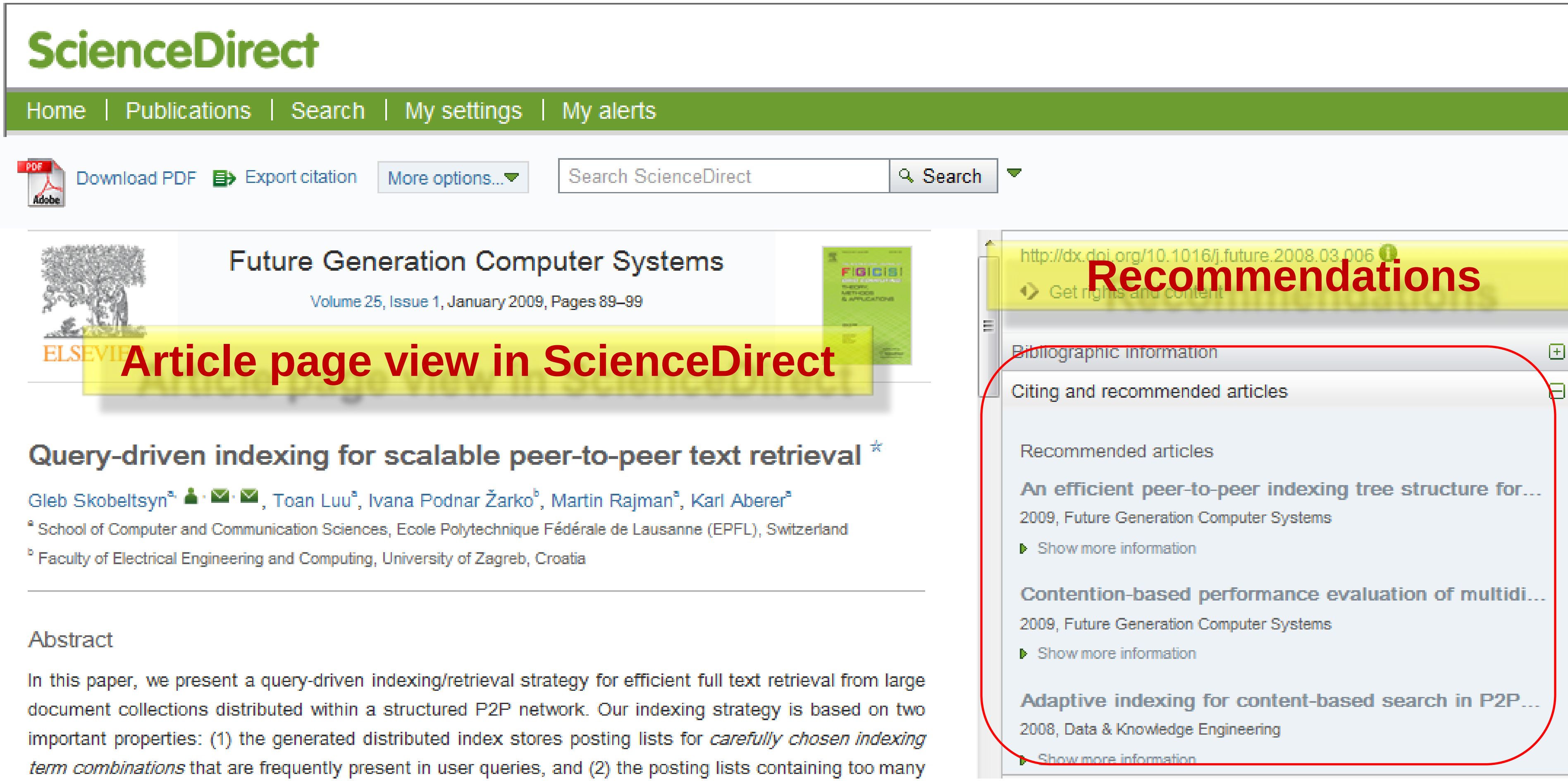
Why this goal?

- Incremental improvement of an existing technology:
  - less risk than the development of a brand new one
  - Shorter time-to-market
  - Limited risk to be faced with unexpected deployment/infrastructure problems
- Measurable success metric: FTA increase

Advanced Recommender ScienceDirect

- ScienceDirect is Elsevier’s full text platform hosting ~12M STM articles and books
- When researchers view articles on ScienceDirect, they’re also provided with links to other articles of interest
  – Previously, these were determined based on content similarity
  – The hypothesis was that exploiting co-downloads might provide better performance (collaborative filtering)

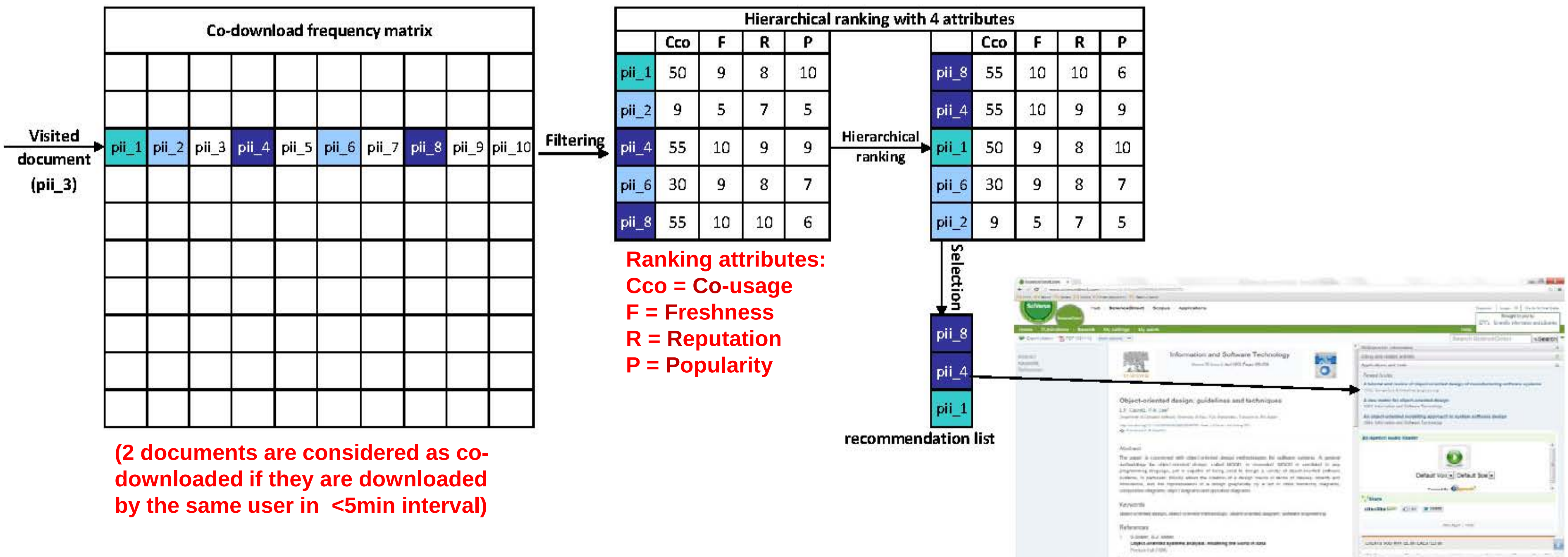
Advanced Recommender ScienceDirect



Advanced Recommender

ScienceDirect

- Two *main* steps to produce a recommendation list for a given document:
  1. Filtering of the most co-downloaded documents;
  2. Ranking of the filtered documents based on freshness, reputation and popularity
- Top 3 in the ranked list constitute the recommendation list for a given document
- Doc similarity used as tie-breaker or surrogate to gauge co-download for new docs



	Cco	F	R	P
pii_1	50	9	8	10
pii_2	9	5	7	5
pii_4	55	10	9	9
pii_6	30	9	8	7
pii_8	55	10	10	6

	Cco	F	R	P
pii_8	55	10	10	6
pii_4	55	10	9	9
pii_1	50	9	8	10
pii_6	30	9	8	7
pii_2	9	5	7	5

Ranking attributes:
Cco = Co-usage
F = Freshness
R = Reputation
P = Popularity

(2 documents are considered as co-downloaded if they are downloaded by the same user in <5min interval)

Tuning the recommender

- The recommender implements a parametric model
- Parameters are:
  – the size of the time window used for building the co-download matrices
  – Ordering of and weights assigned to the various scores in the aggregated ranking function used to select the most promising recommendation candidates

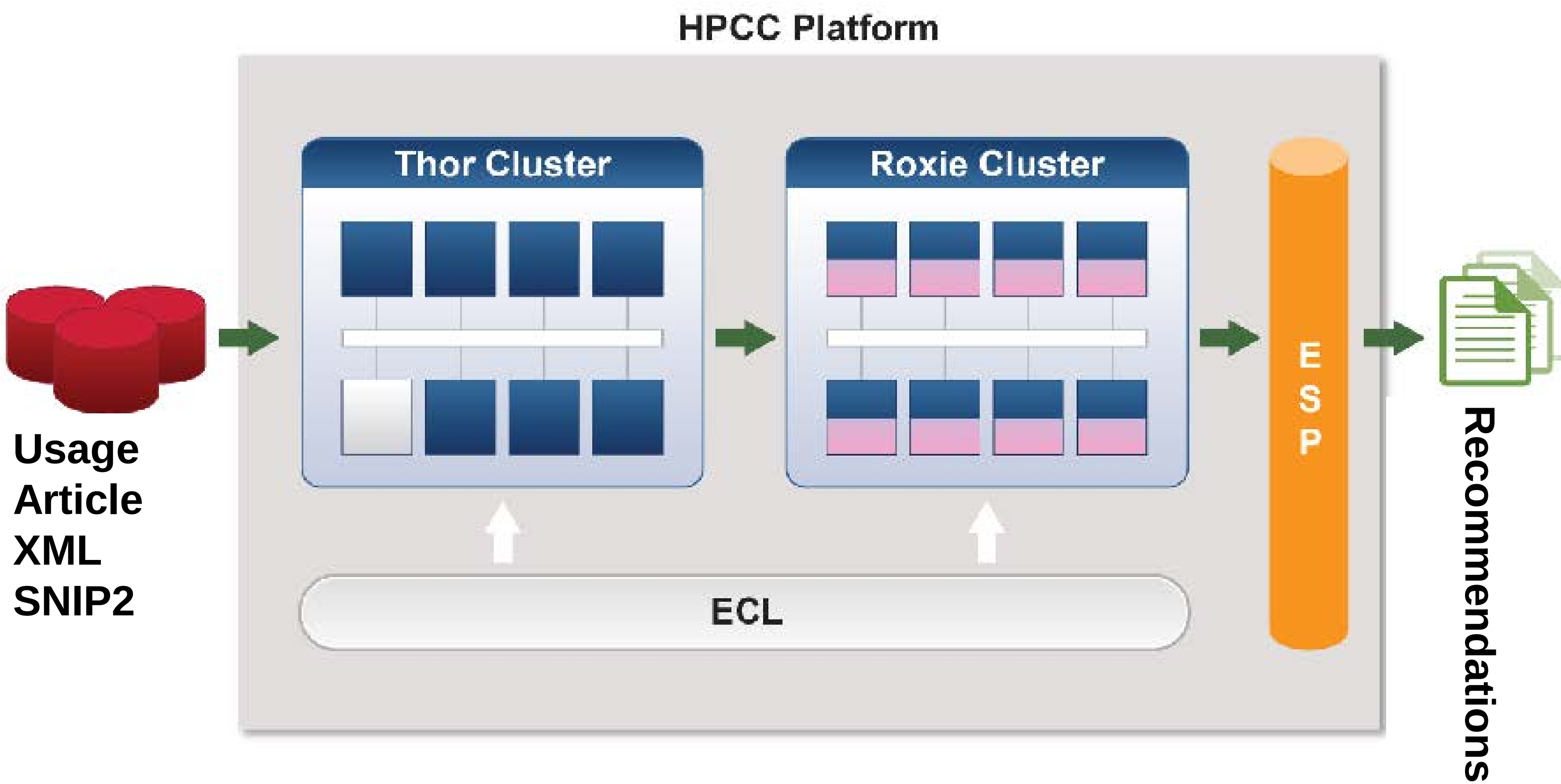
Tuning the recommender (2)

- One of the important goal of the collaboration was to find the best performing parameter settings
- To achieve this, a mix of informed guessing and machine learning on available data was performed... and the resulting parameter setting(s) were evaluated through A/B-testing

Developing the Recommender

- Key demand: efficient processing of dynamic, large-scale data
  - Large-scale matrix processing
  - Daily updates/fresh recommendations
  - High volumes of queries/displays
- HPCC (HPCC systems)
  - High Performance Clustered Computing
  - Open source big data platform
  - Commodity hardware
  - Private Cloud @ Lexis Nexis

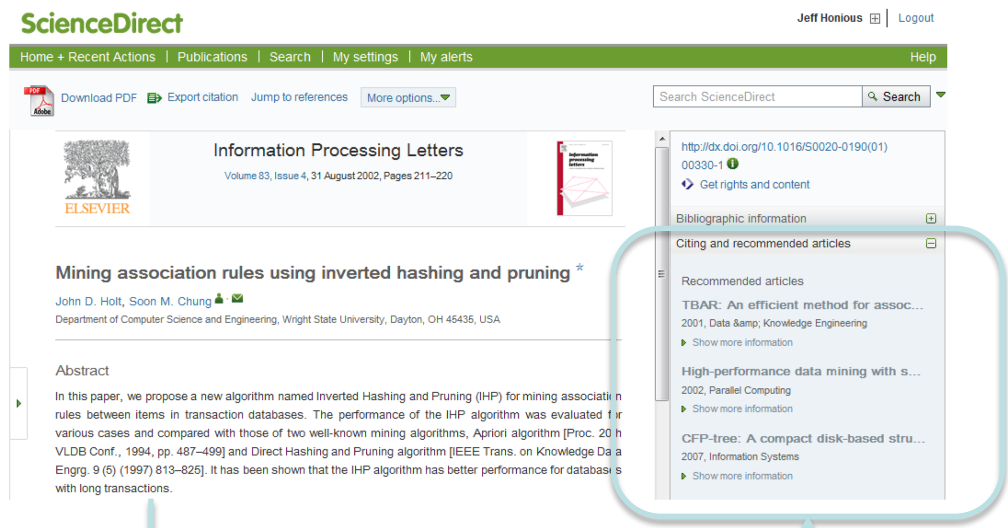
HPCC Architecture

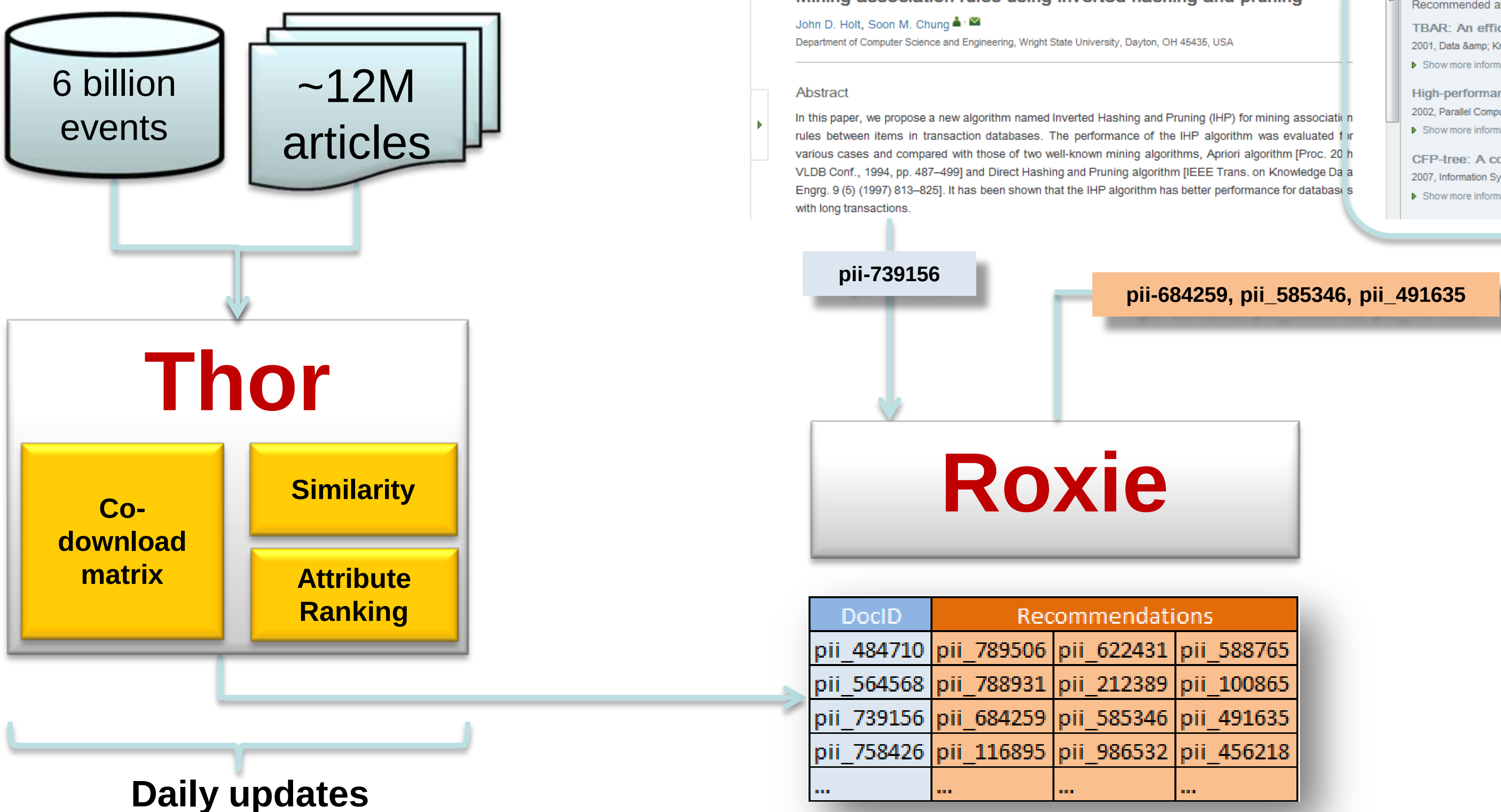


HPCC: Recommender

ScienceDirect

Multiple years of SD usage data/events
All SD XML Articles
Journal Rankings

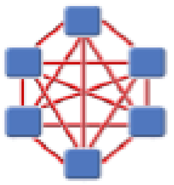




DocID	Recommendations		
pii_484710	pii_789506	pii_622431	pii_588765
pii_564568	pii_788931	pii_212389	pii_100865
pii_739156	pii_684259	pii_585346	pii_491635
pii_758426	pii_116895	pii_986532	pii_456218
...	...	...	...

HPCC Systems:

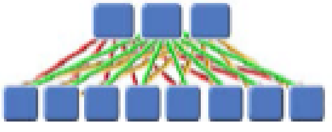
❶ **HPCC Systems Data Refinery (Thor)**



- 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 Systems 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)**

3,980 Lines of ECL

↓

482,410 Lines of C++

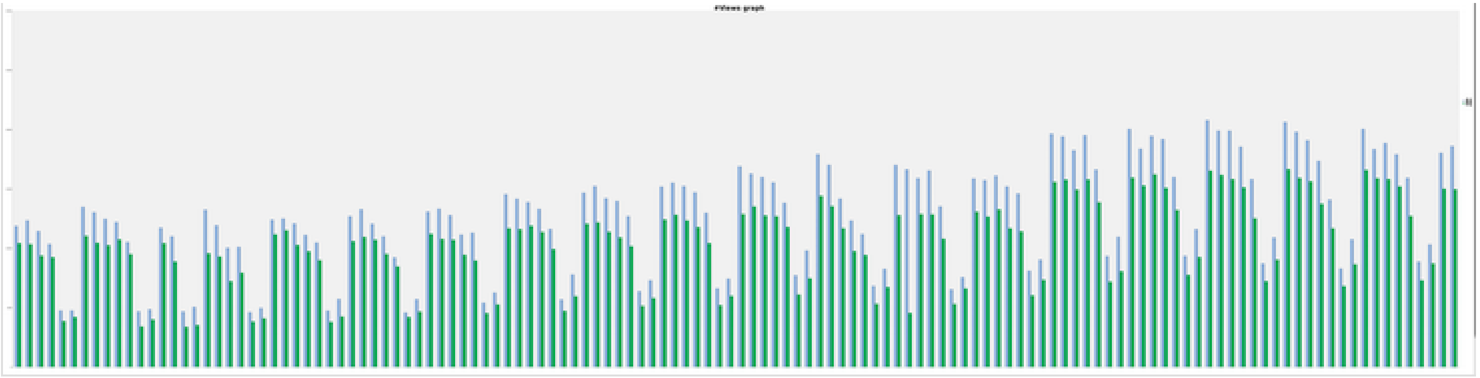
- An easy to use , data-centric programming language optimized for large-scale data management and query processing
- Highly efficient; Automatically distributes workload across all nodes.
  - Industry analysts estimate 80% more efficient than C++, Java and SQL and 1/3 reduction in programmer time to maintain/enhance existing applications
  - 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

Developing the prototype

- Minimum Viable Product (MVP) approach:
  - Done outside main product release schedule
  - Using only a small but representative set of traffic volume
  - Essentially a “pseudo-production” infrastructure, using cloud computing
  - Rapid (start to finish 6 month project)
- Delivered:
  - Data flows to HPCC
  - UI component to display recommendations
  - A/B testing
  - Logging
- The prototype components had to be implemented with minimal modification to the existing infrastructure

Evaluating the prototype

- Another important constraint imposed on the collaboration is that the produced prototype can be objectively evaluated
- The targeted performance metric is the CTR within the Recommender Box
- Due to the presence of important seasonal variations in the FTA counts, an A/B testing approach must be used



Development Cycles

During the pilot…

- EPFL cycled through 9 *variants* while testing for best dimension combinations & weights
- A/B testing was used on 9 variants, each measured during a fixed usage window
- Compared to existing related articles

Current Recommender: Main facts

- Deployed in Aug 2013
- Continuously A/B-tested since then (>130 days)
- Daily synthetic A-/Btest report over mail
- Detailed A/B-test report available on the Web
- More than 50% better (relative difference with 90% certitude) than the “related articles” baseline

Next Steps

- Currently the recommender provides identical recommendations for all users; the goal is to identify user communities for which tailored recommendations could be generated
- Currently the recommender is used in a data PULL scenario; the goal is to exploit recommender techniques in a data PUSH scenario, e.g. mail ALERTS
- Currently the recommender is focused on identifying documents ‘interesting’ for recommendation; the goal is to quantify document ‘quality’ for automatic recommendations
- The recommender is the first example of the use of large-scale computing to combine multiple sources and types of data (usage, full text, SNIP2); the goal is to extend this into a broader data management platform capable of providing an experimental framework for further data-driven products/services

Acknowledgments

- EPFL:
  - Thierry Delbecque
  - Manh Hung Nguyen
- Elsevier:
  - Olivier Dumon
  - Chris Shillum
  - STEPS (Girish Manchaiah, Aymen Barri, Melissa Ringgold et al.)
  - HPCC (John Holt, Jeff Honious, Jon Burger et al.)

Thanks and any questions

ScienceDirect

- Martin.rajman@epfl.ch
- C.scott@elsevier.com

[redacted] 5 Oct

Whoever built the @sciencedirect "other articles you might be interested in" engine. I owe you many **pints**. It's absolutely awesome.

Collapse · Reply · Retweet · Favorite · More

9 RETWEETS · 6 FAVORITES

ScienceDirect article suggestions

ScienceDirect

Recommended articles

People who downloaded this article also downloaded these articles. Learn more

Python in scientific computing: Applications to Bose–Einstein condensates
Computer Physics Communications, Volume 177, Issues 1–2, July 2007, Pages 45
Jon K. Nilsen
Show abstract | PDF (160 K)

MontePython: Implementing Quantum Monte Carlo using Python
Computer Physics Communications, Volume 177, Issue 10, 15 November 2007, Pages 799–814
Jon Kristian Nilsen
Show abstract | PDF (631 K)

Using B SP and Python to simplify parallel programming
Future Generation Computer Systems, Volume 22, Issues 1–2, January 2006, Pages 123–157
Konrad Hinsen , Hans Petter Langtangen , Ola Skavhaug , Åsmund Ødegård
Show abstract | PDF (325 K)

View more recommended articles