

XQBench – A XQuery Benchmarking Service

Peter Fischer
ETH Zurich
<http://xqbench.org>

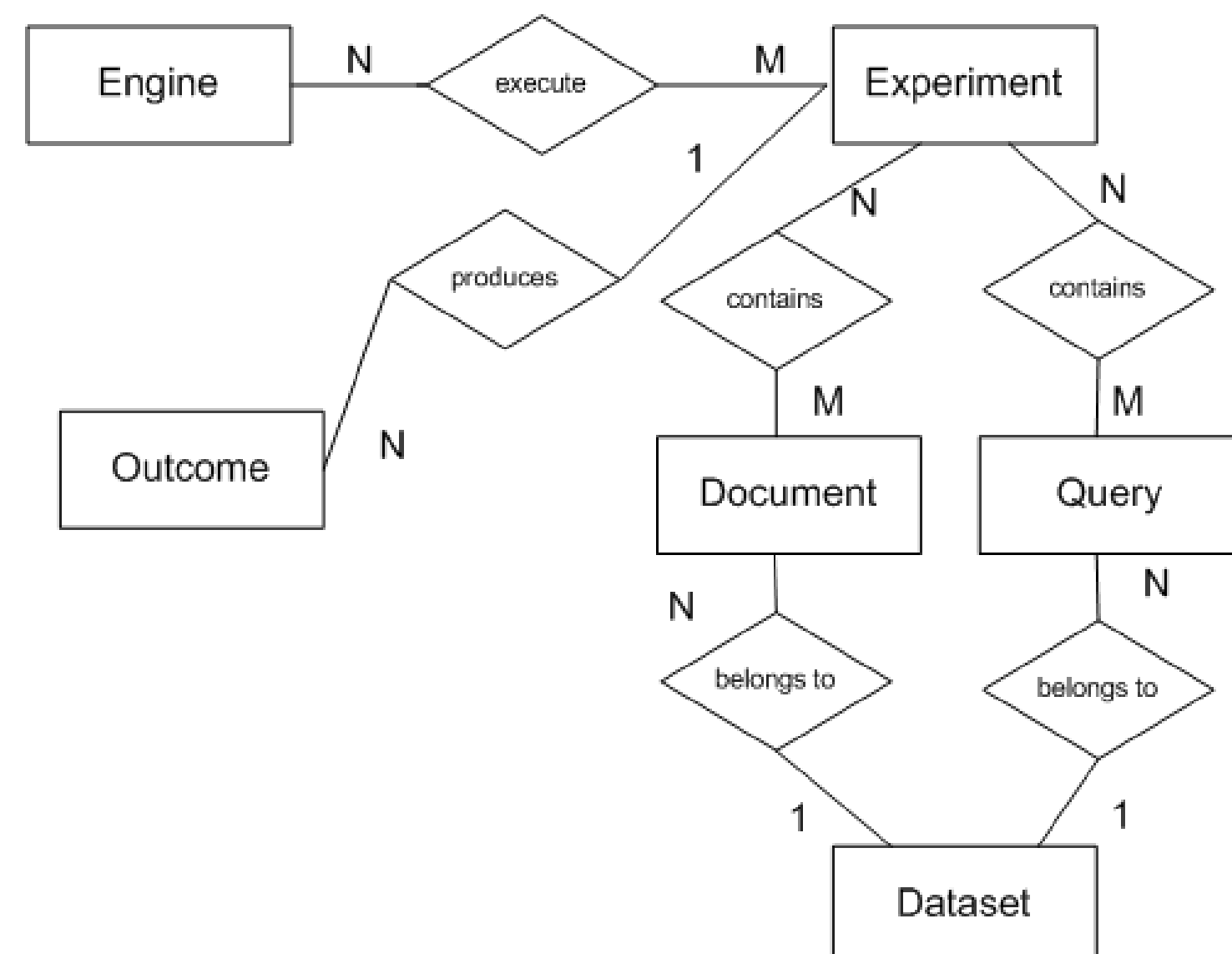
Motivation and Goal

- Help you decide which XML implementation meets your requirements
- A service that runs
 - standard XML benchmarks (XMark, TPoX, ...)
 - user-defined benchmarkson major XQuery Processors and Databases (Zorba, Saxon, eXist, etc.)
- You can use it out of the box
 - no installation of software
 - Web-based interface
 - automatic compilation of benchmark results
 - reference to previous

Benchmarking Model

- Model various use cases of XQuery:
 - Processors (cold start, repetitions)
 - Databases (ad-hoc loading, preloaded)
- Evaluate individual read queries on static datasets
- Measure overall time, details when possible
- Observe resource consumption and results

Data and Metadata Model



Outcome

- Measured Times
- Platform and Timing Information

Experiment

- Involved documents, queries, engines
- Grouping of runs by dataset and engine class

(Meta)data for Documents

- Dataset
- Reference to File Location, Size
- Generator Information

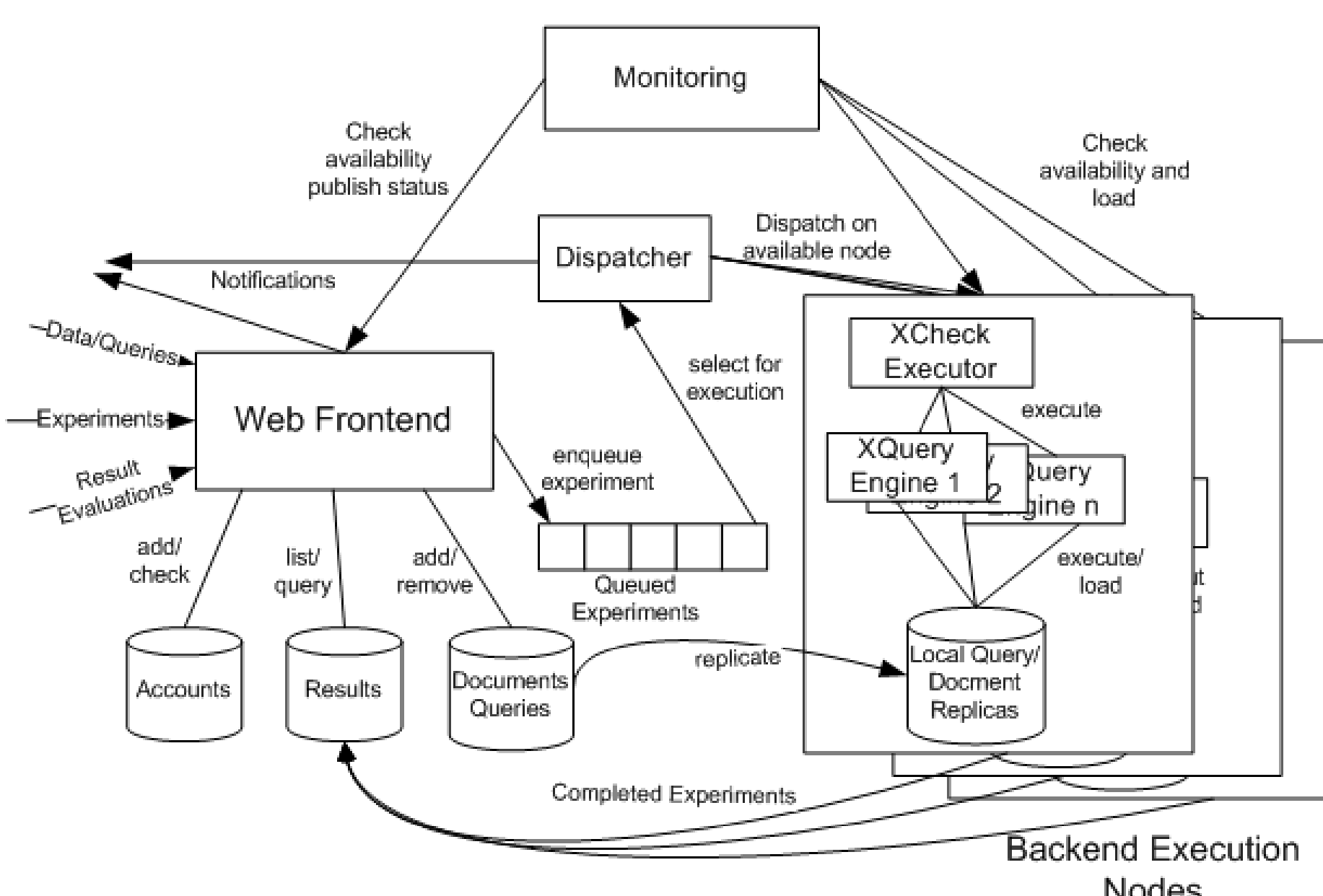
(Meta)data for Queries

- Language and Categories
- Query text
- Expected result on reference document
- Validation Query for Updates (not used yet)

Engine

- Classification (Processor, Database)
- Supported languages/features
- Install Path
- XCheck Adapter

Benchmark Service Architecture



Requirements

- Experiments must not influence each other,
- Frontend operations must not affect running experiments
- Service must be scalable over
 - data size
 - experiment size (docs, queries, engines)
 - experiment number

Frontend

- manage user registrations and permissions
- upload and manage datasets and queries
- submit experiments
- show, search and evaluate the outcomes
- Implemented as JSP+XQuery

Monitoring

- Monitor system parameters
 - CPU/Load
 - RAM/Swap
 - Disk/IO

Backend

- XQuery Engines and replica of data and queries
- Utilize XCheck as execution backend
- Wrap Engines into „adapters“, describing
 - Setup of engine (start, loading)
 - Query Execution
 - Teardown (discarding, stop)
 - Extraction of Timing
- Repeat Experiments N+1 times, average last N
- Enforce timeout to stop „runaway“ experiments
- Render results into HTML, XML, Graphs

Data and Control Flow

- Queue experiments until a suitable execution backend becomes available.
- Pull Experiment into Backend node, synchronize documents and queries
- Propagate results towards repository

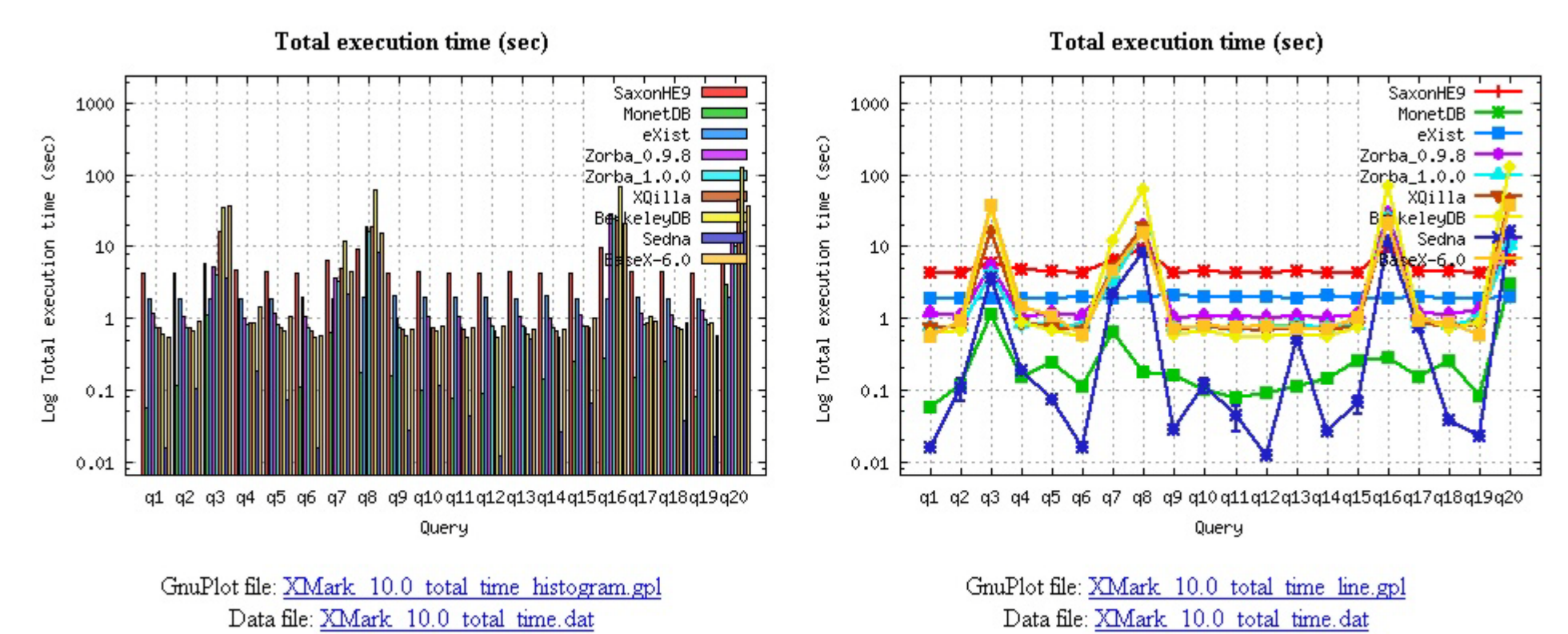
Usage

Public and Private Data/Queries/Experiments

- Everybody can run and search standard benchmarks
- Registered users can add own data and queries
- Private data and outcomes only accessible to owner
- Private data and outcomes can be made public

Result Presentation and Search

- For every outcome, multiple views:
 - Experiment description
 - HTML (textual) for
 - XML for further querying
 - Plots (PNG, PDF) in various dimensions
- Search Options
 - Complete List
 - Directed GUI search for documents, queries, engines
 - General XQuery for complex constraints



Status

Platform

- ETH provides frontend and several backend nodes
- Available at „beta“ stage
- Helper infrastructure
 - Bugtracker
 - Wiki
 - Mailing List
- Components are/will be open source

Supported XQuery Implementations

- Open-Source platforms: BaseX, BerkeleyDB, eXist, MonetDB, Saxon B, Sedna, XQuilla, Zorba
- Commercial platforms will need vendor agreement

Preinstalled Benchmarks

- XMark, TPoX, MeMBer (data only)
- BooX (new app benchmark covering all XQuery)

Future Work

- Provide broader coverage of benchmarks and engines
- Exploit and collect metadata more extensively
- Extend benchmarking model for updates, state propagation, pipelines etc
- Foster community involvement