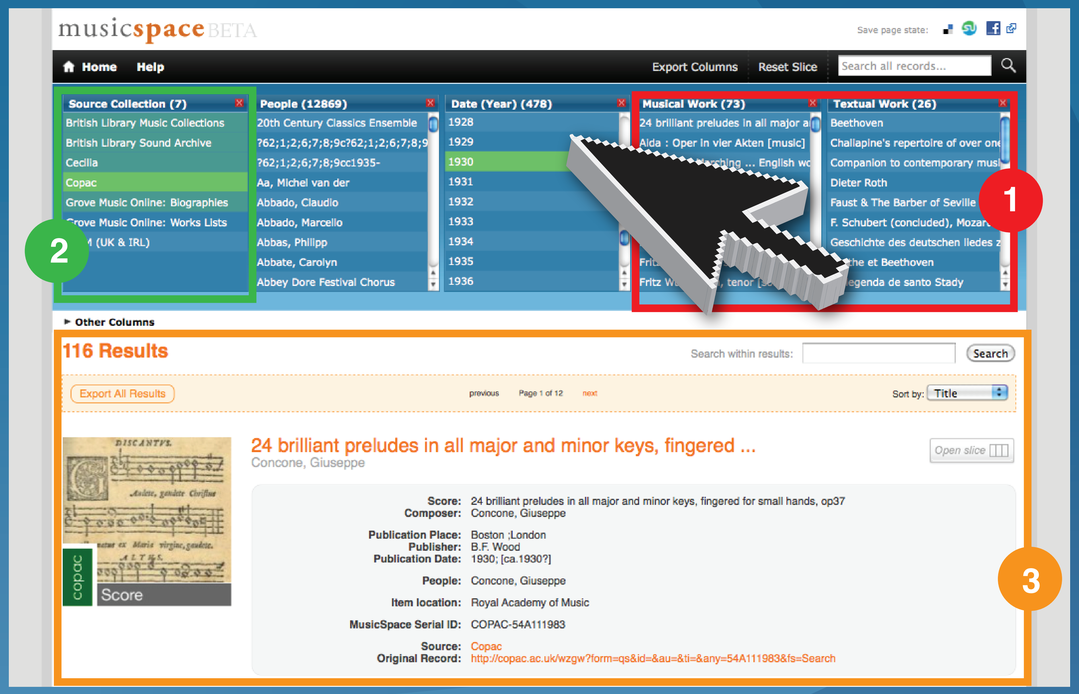
QWIC Unclogging data pipes for fast interactions

UNIVERSITY OF Southampton
Arts & Humanities Research Council



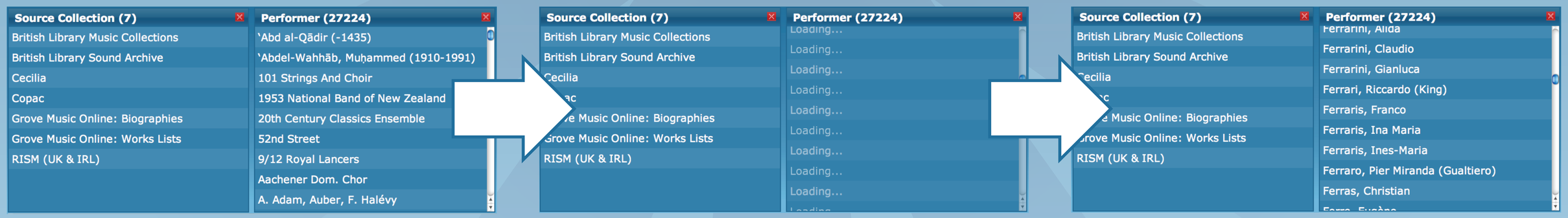
In rich exploratory interfaces, like faceted browsers, a single click can fire off a range of queries that each take time to process. For instance, in a column browser like mSpace (shown left), clicking a facet

(1) filters every facet in all columns to the right
(2) highlights all possible paths for that item in all columns to the left
(3) finds and displays all instance search results for that list.

When the data sets are large (100k records), traditional Web2.0 approaches of doing everything in the client break down, and performance grinds to a halt, keeping researchers from using effective data set sizes to evaluate their interaction in the field and longitudinally.

Quick Web Interface Control (QWIC) is a set of hard won, but simple-to-apply heuristics that can speed up any Web UI that does a lot over a lot to return a lot to the UI with one click

Example of QWIC in action: viewport performance: 200k records queried and returned in 80ms



Initial Data Loads In Column

User Scrolls & Viewport Requests More Data

Data Loads Into The Column

Use Real Separation for Real Performance

Moving the logic from the Client to the server and maintaining state in the URL rather than in a session means heavy lifting is done by the database, which is optimised for querying.

Use View ports rather than paging for processing long lists

Paging is a standard way to handle lists of result but that means a user clicking "next" and "previous" continuously. A viewport provides a fully scrollable list that loads items dynamically, so that the user gets a sense of the number of results, and retains context from the affordances of a list, rather than pages

Use Full Compound Indexes rather than skinning UI on existing schema

The temptation is to use an existing database schema directly to map to a rich data driven UI. Compound indexes of facets, normalised specifically for a UI, however, are a straightforward way to gain performance benefits in a database. Compound indexes for possible facets mean that the database does not need to scan its tables, so that much more data can be put into the UI.

D. A. Smith, J. Lambert, D. Bretherton, mc schraefel

http://musicspace.mspace.fm