The Efficiency of Spatial Indexing Methods Applied to Large Astronomical Databases

G. B. Berriman and J. C. Good

Caltech/IPAC, Mail Stop 100-22, Pasadena, CA 91125

B. Shiao and T. Donaldson

Space Telescope Science Institute, 3700 San Martin Drive, Baltimore, MD 21218.

Why Indexing is important?

Un-Indexed Database

INDEXED Database

Why Study Indexing Performance?

- Spatial indexing methods used in astronomy are based on quadrature methods.
  - Partitioning of the sky into cells.
  - Use cell numbers to create an index, usually a binary tree (B-tree).
  - Index written as database column.
- What determines the performance of a database:
  - Index used?
  - Index level?
  - Choice of hardware?

Compare performance of HTM and HEALPix indices on Solaris and Windows platforms on a common catalog, 2MASS All-Sky Point Source Catalog, installed in a PostgreSQL database.

Procedure

- **Create Catalog:** All 470 million records of the 2MASS All-Sky Catalog, stripped of all fields except 2MASS designation, RA and Dec.
  - Add to a source table spatial index value for some HTM/HEALPix level and x, y, z spherical 3-vector values.
  - Ingest the table into PostgreSQL.
  - Create a B-Tree index for the index column.

- **Timing the Queries:**
  - Use spatial constraint (e.g. location/radius) to construct
    - List of spatial bins touched by region;
    - Exact spatial filter based on 3-vector.
  - Augment each SQL query (for us no other relational constraints) with the above.
  - Run 1 million such queries with random sky positions and cone search radius (log scale between 1 arcsec and 1 degree).

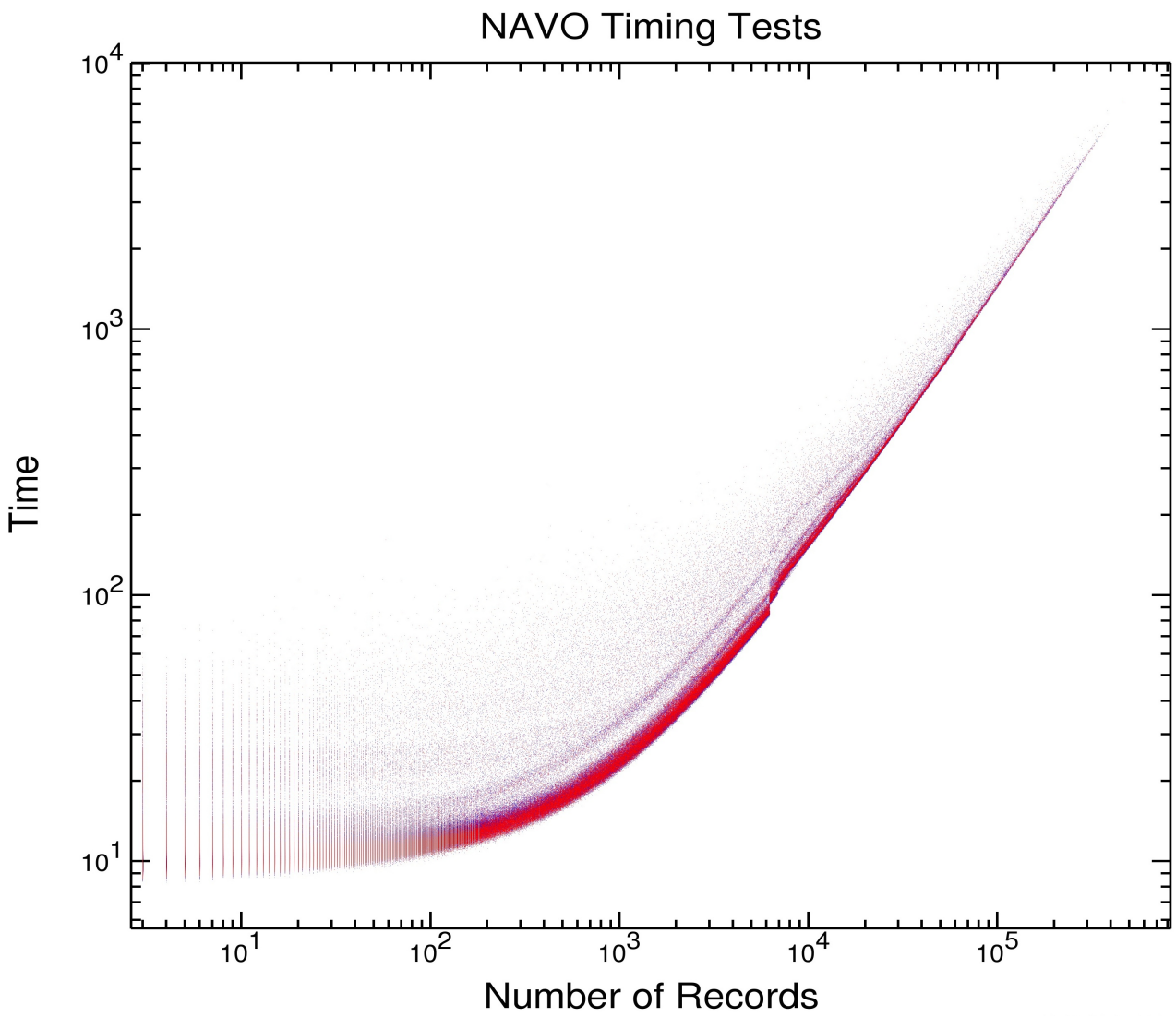
Platforms

	Database	OS/Compiler	Server
STScI	PostgreSQL 9.5.	Win 2012 server OS. Cygwin 2.5.2 DLL with gcc 5.4.0.	2 processors @3.46 GHZ each with 6 cores each. 12 logical processors. Total of 24 processors.
IPAC	PostgreSQL 9.3.5	Solaris 10	Xeon 2.27 GHz. 2 processors, 4 cores.

Software Package

- Developed software package in ANSI-C for creating database indices and constructing queries:
  - HTM and HPX libraries;
  - Utility for adding HTM/HPX indices to a table of locations;
  - Utility for turning a region specification into SQL fragments for inclusion in a query;
  - Utilities for running a single cone search and large-scale timing tests against PostgreSQL code, with Makefile and README;
  - Utility for generating uniformly spaced random locations on the sky (intended for inclusion in a test program).
- Self-contained, but the PostgreSQL search utilities require the PostgreSQL libraries.
- We plan to release the code with an Open Source (BSD 3-clause) license in Fall 2017.

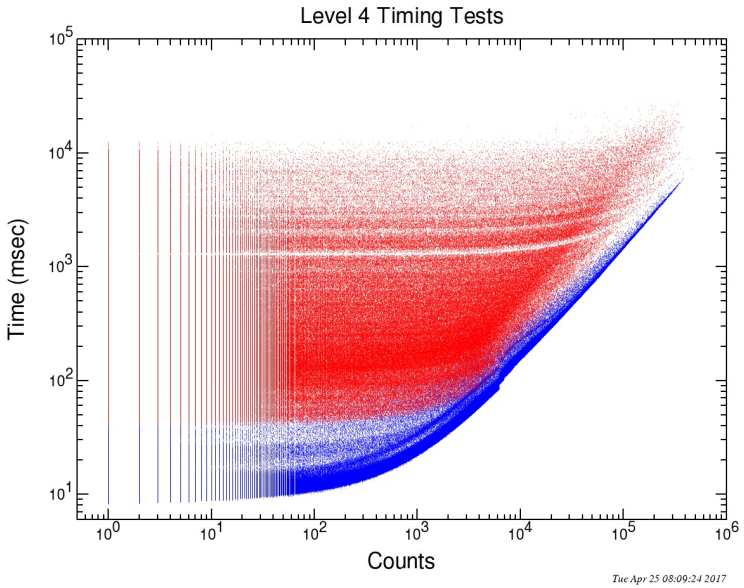
Solaris Level 14

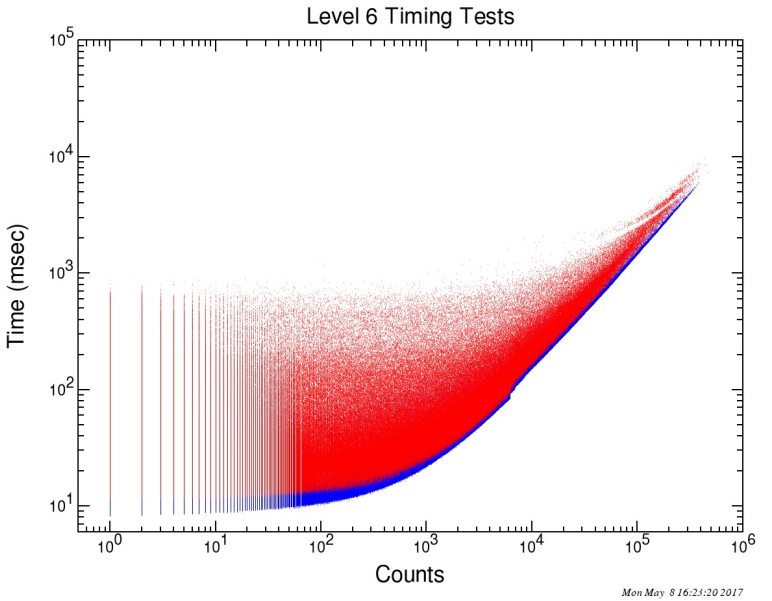


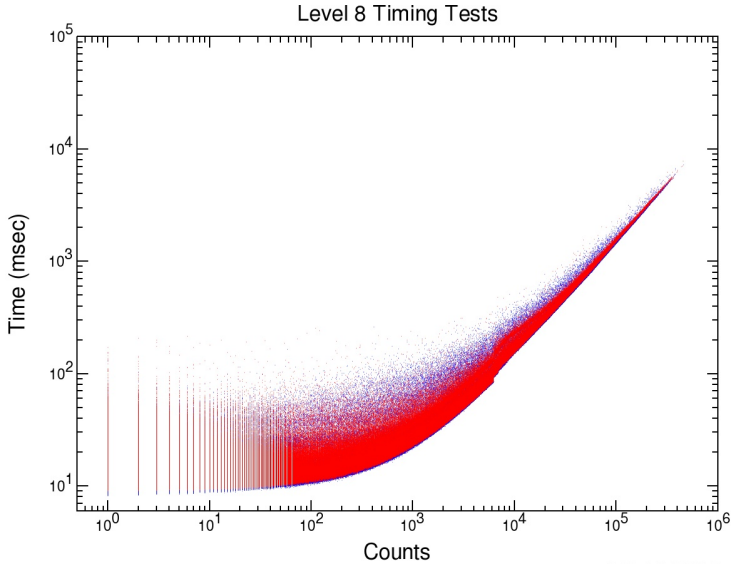
HTM blue,
HEALPix red

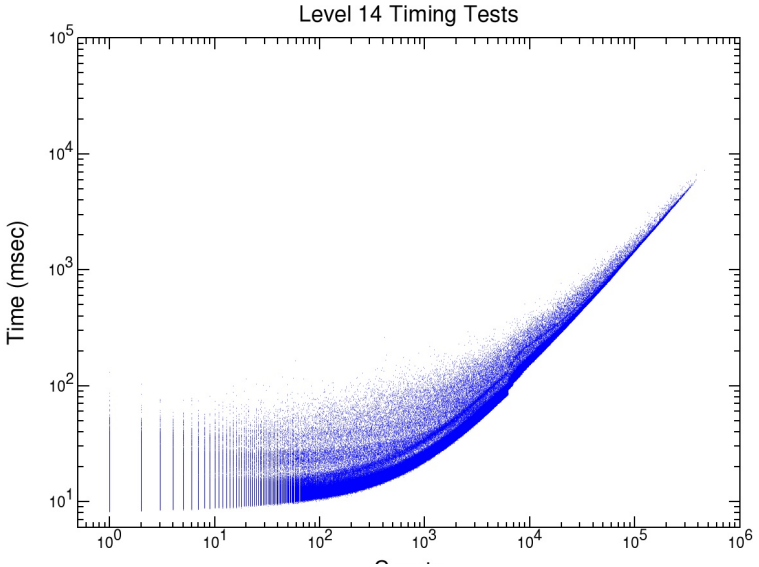
HTM Level 14
HEALPix Level 14

Effect of varying the index level

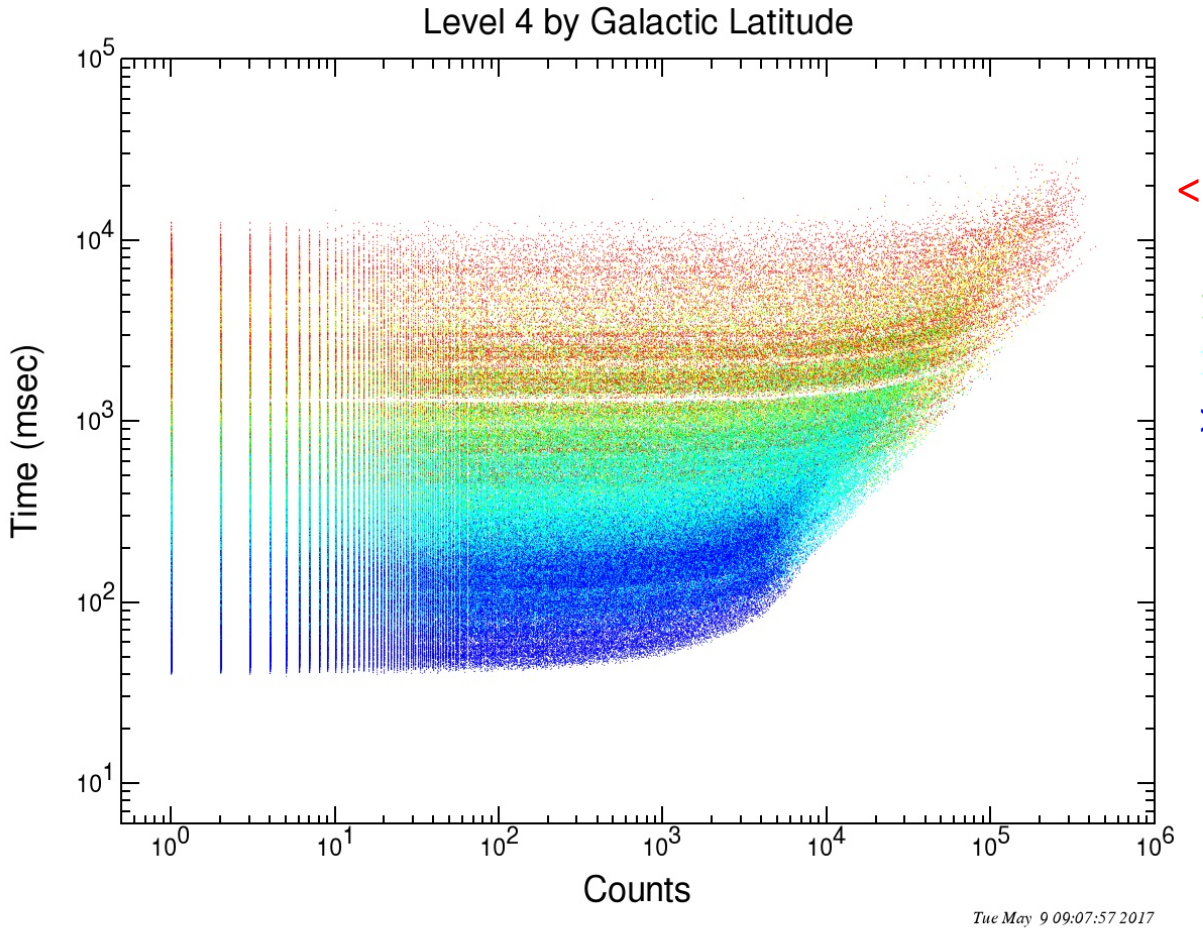








Level 4 results encoded by Galactic Latitude



< 5 degrees: red

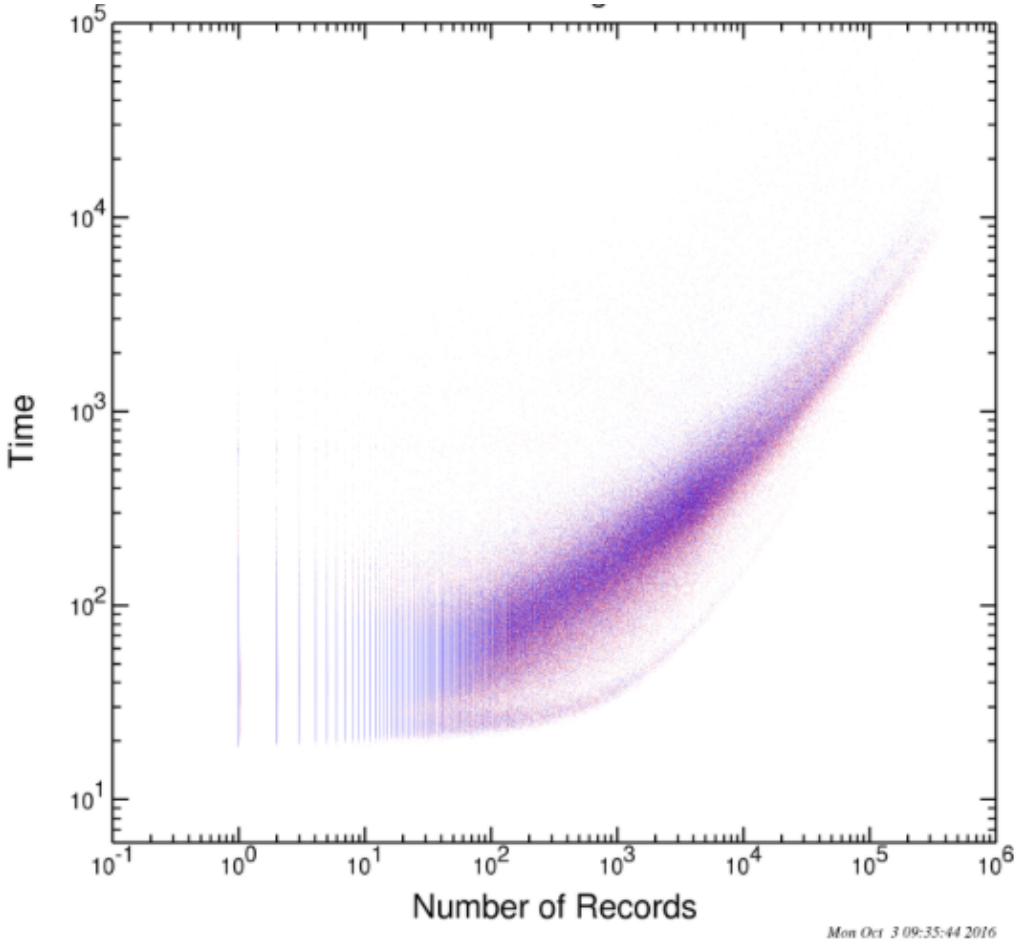
5-10 degrees: yellow

10-15 degrees: green

15-40 degrees: cyan

> 40 degrees: blue

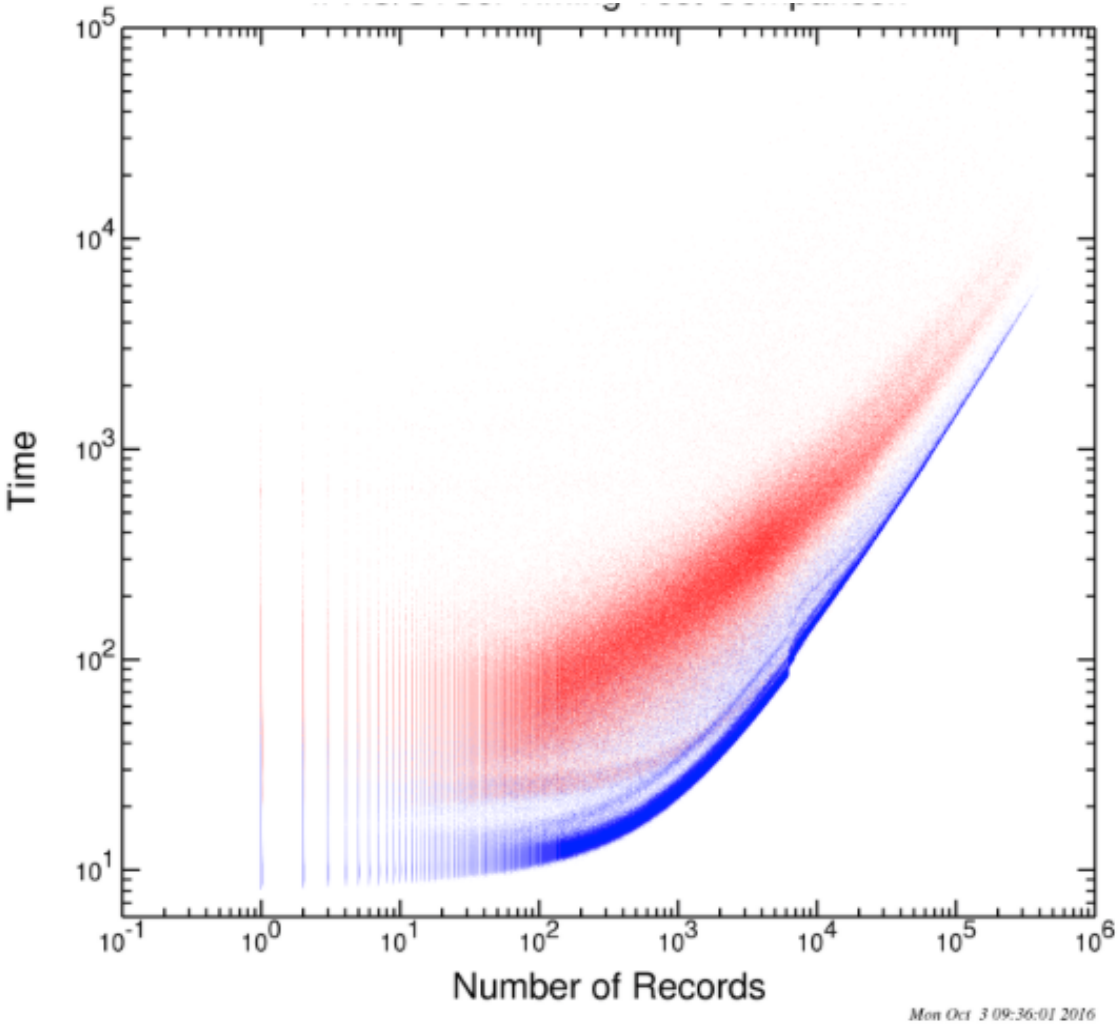
Windows Level 14



HTM blue,
HEALPix red

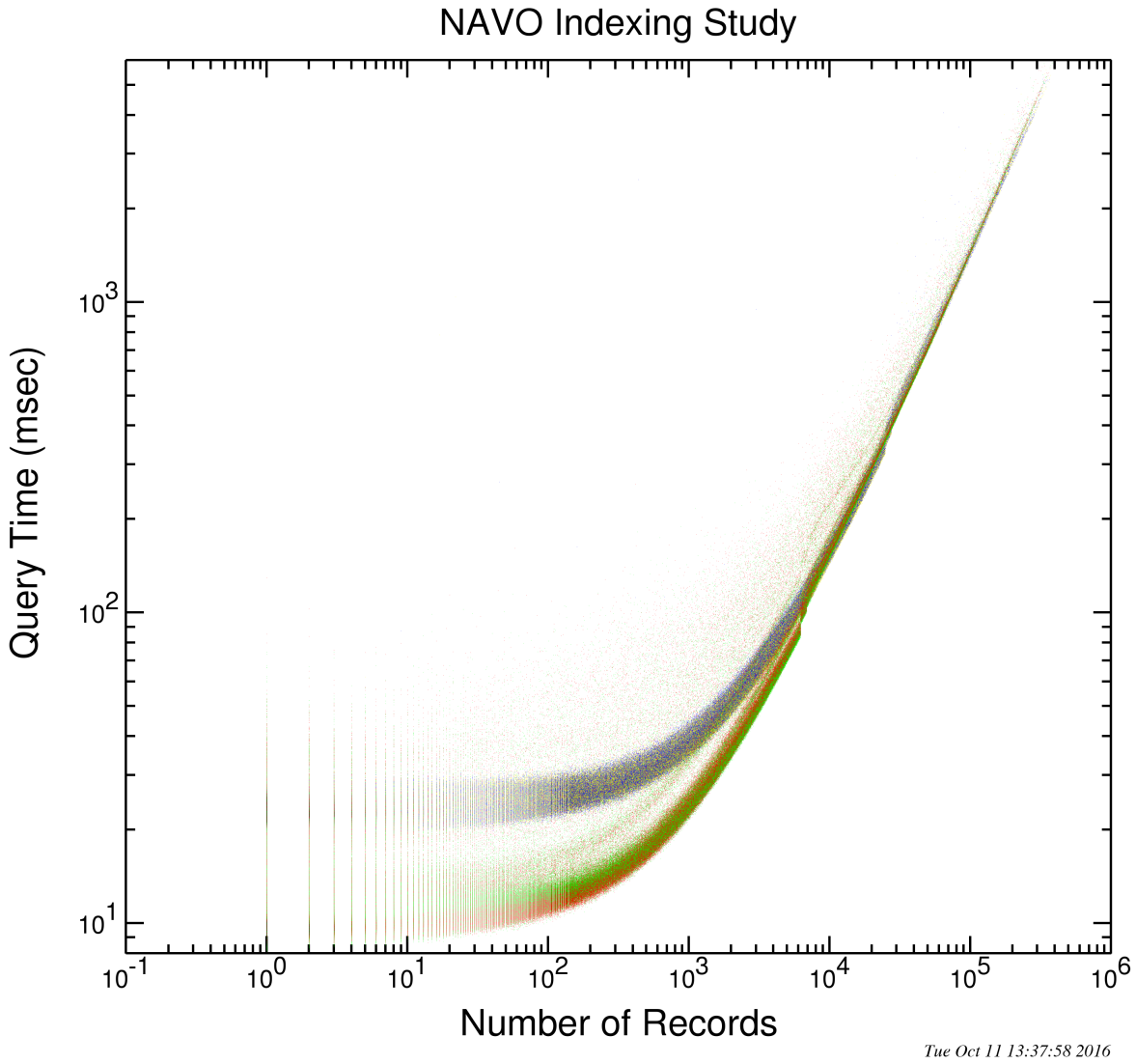
HTM Level 14
HEALPix Level 14

Windows and Solaris HEALPix Level 14



SunOS blue,
Windows red

Summary Chart Level 14



Windows/HEALPix blue
Windows/HTM yellow

Solaris/HEALPix red
Solaris/HTM green

What Have We Learned So Far?

- Choice of index less important than the configuration of the hardware platform.
- The total query time is dominated by I/O and is linear with number of records returned for large numbers of sources.
- Performance does improve as indexing level increases, but not much improvement from level 8 to level 14.

Next Steps

- Investigate detailed structure of timing tests (e.g. bands in level 4).
- Study effect of indexing depth on patchy dense data (e.g. Hubble Source Catalog).
- Study effect of increasing indexing level beyond level 14.
- Compare performance of SQLServer and PostgreSQL.
- Deliver software package.
- On-line documentation of results.
- Present final results at ADASS and write a paper!