TOPCAT Visualisation on the Web

Mark Taylor (Bristol)

ADASS 2020
Virtual Granada

10 November 2020

$Id: talk.tex,v 1.22 2020/11/10 10:24:39 mbt Exp $

Outline

- TOPCAT/STILTS very short introduction
- Context
- Architecture
- Usage and applicability
- Demo
- Deployment
- Status and future work

TOPCAT/STILTS Overview

TOPCAT = Tool for OPerations on Catalogues And Tables

“Does what you want with tables”

TOPCAT

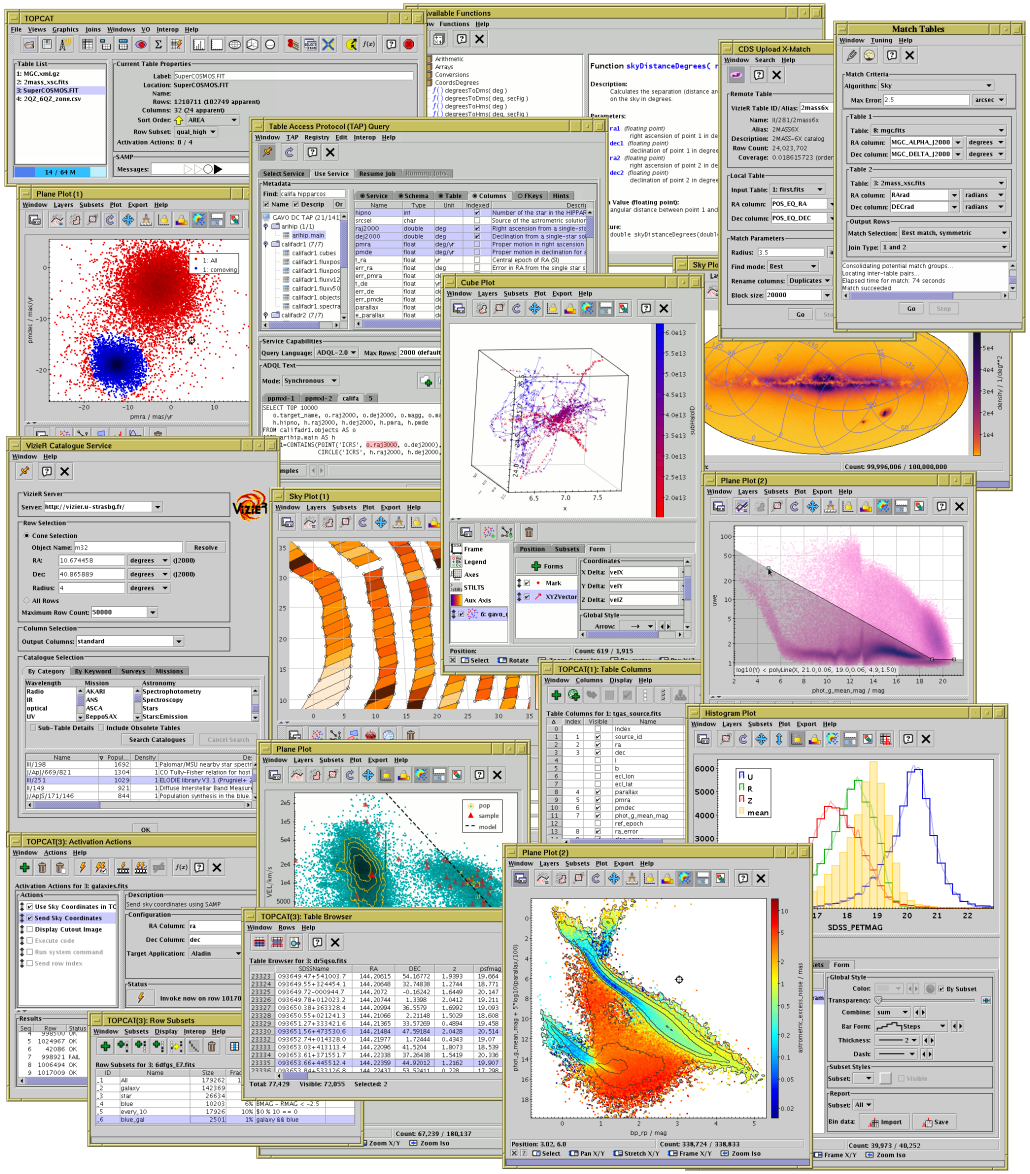
- Desktop GUI Java application
- Good for interactive exploration

STILTS

- Suite of command-line tools
- Good for scripted/reproducible/batch use

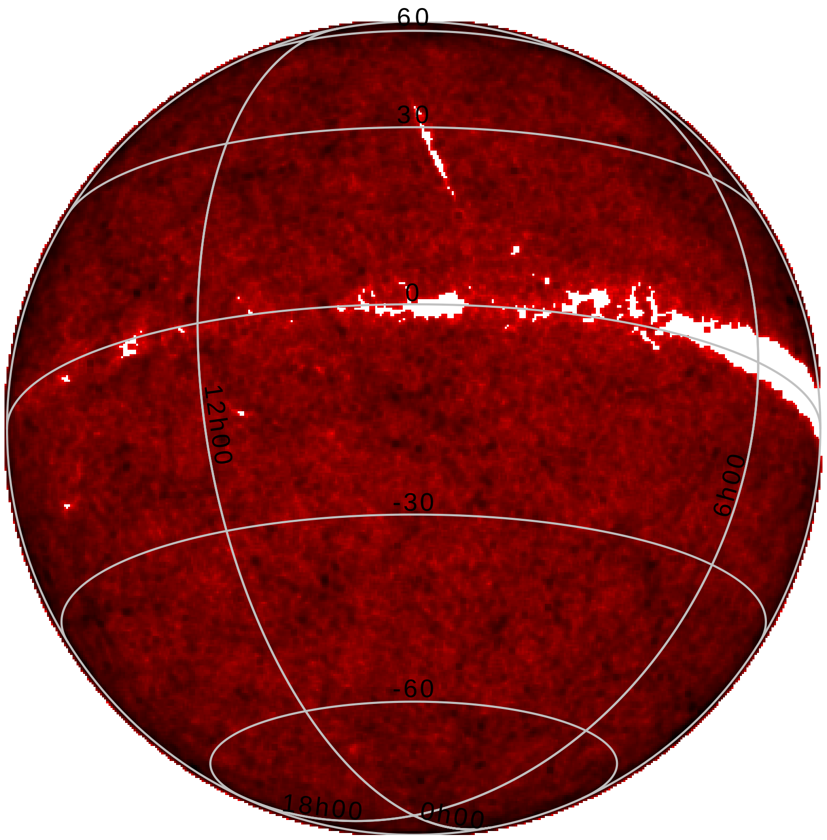
Overall aim:

- Make table manipulation easy,
  so users can concentrate on **doing science**



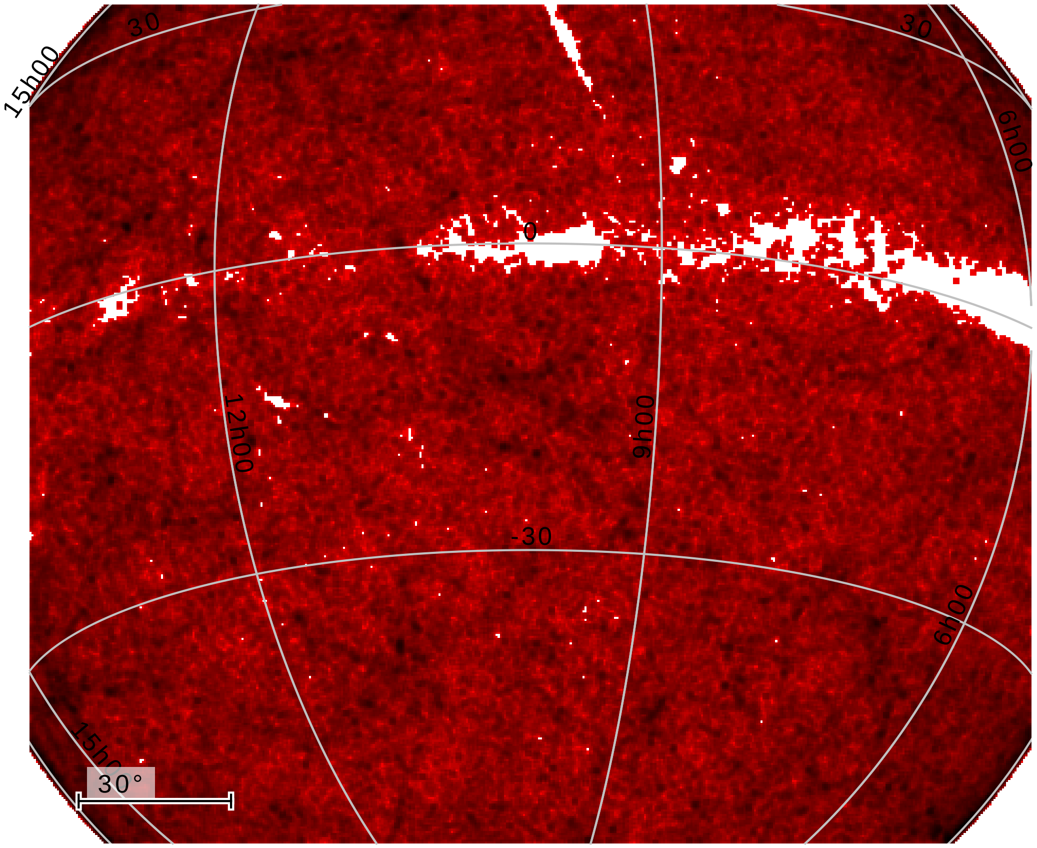
Visualisation Strengths

- Scalability
  - Multi-million row plots
  - No special data preparation required
  - Low memory usage
  - Multithreaded rendering
- Flexibility
  - Many different plot types
  - Many different configuration options
- Interactivity
  - Navigate round a 2d/3d plot
  - Smooth high/low point density transition
  - Config option changes instantly visible
- Linked Views
  - Select in one view, see inclusion in others
  - See information about selected points



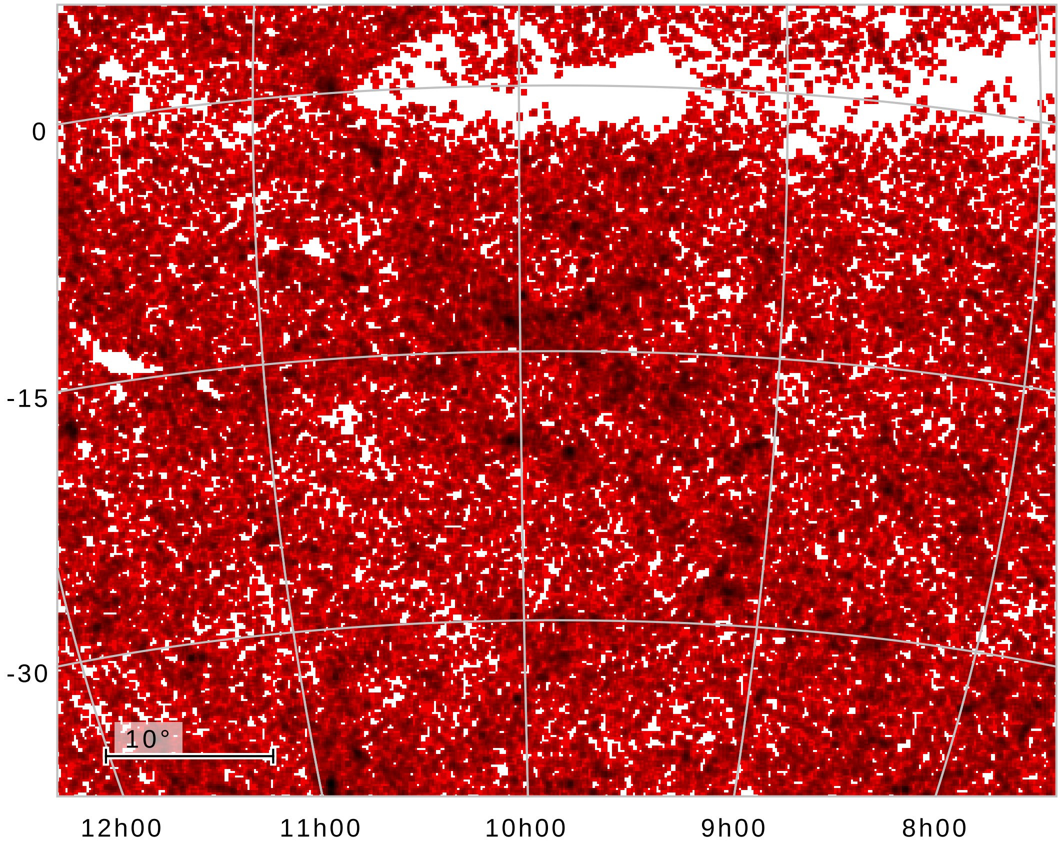
Visualisation Strengths

- Scalability
  - Multi-million row plots
  - No special data preparation required
  - Low memory usage
  - Multithreaded rendering
- Flexibility
  - Many different plot types
  - Many different configuration options
- Interactivity
  - Navigate round a 2d/3d plot
  - Smooth high/low point density transition
  - Config option changes instantly visible
- Linked Views
  - Select in one view, see inclusion in others
  - See information about selected points



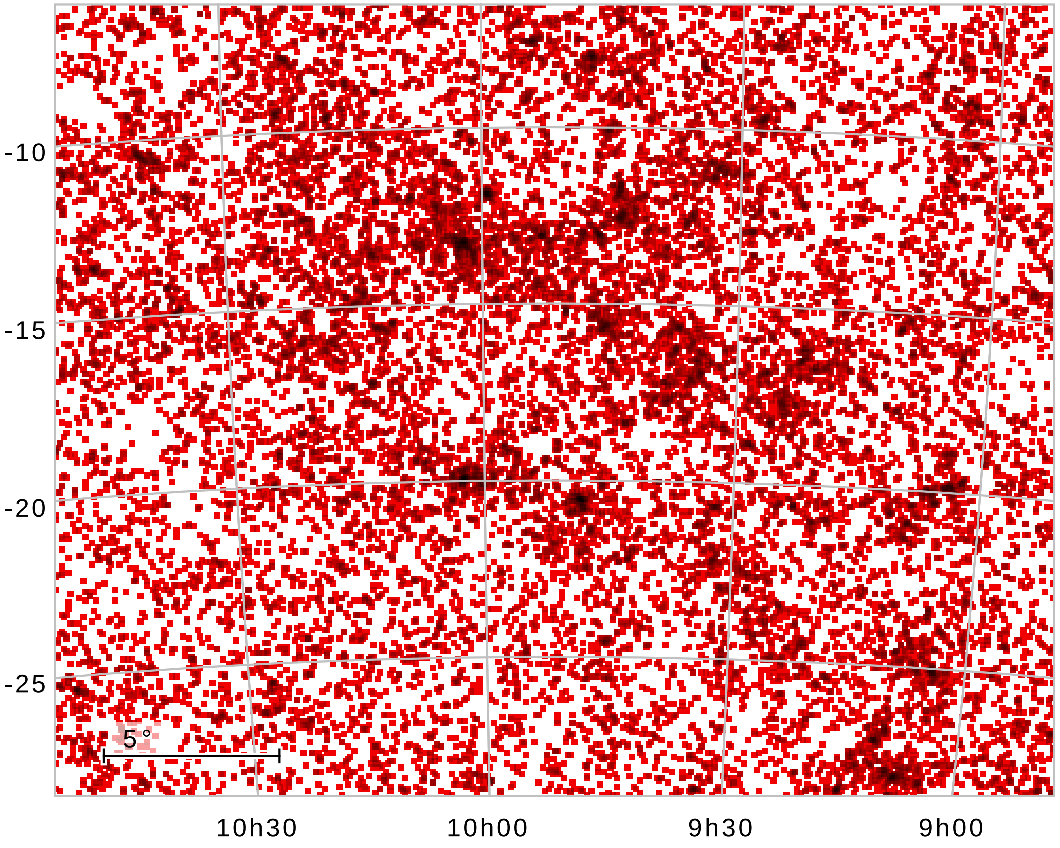
Visualisation Strengths

- Scalability
  - Multi-million row plots
  - No special data preparation required
  - Low memory usage
  - Multithreaded rendering
- Flexibility
  - Many different plot types
  - Many different configuration options
- Interactivity
  - Navigate round a 2d/3d plot
  - Smooth high/low point density transition
  - Config option changes instantly visible
- Linked Views
  - Select in one view, see inclusion in others
  - See information about selected points



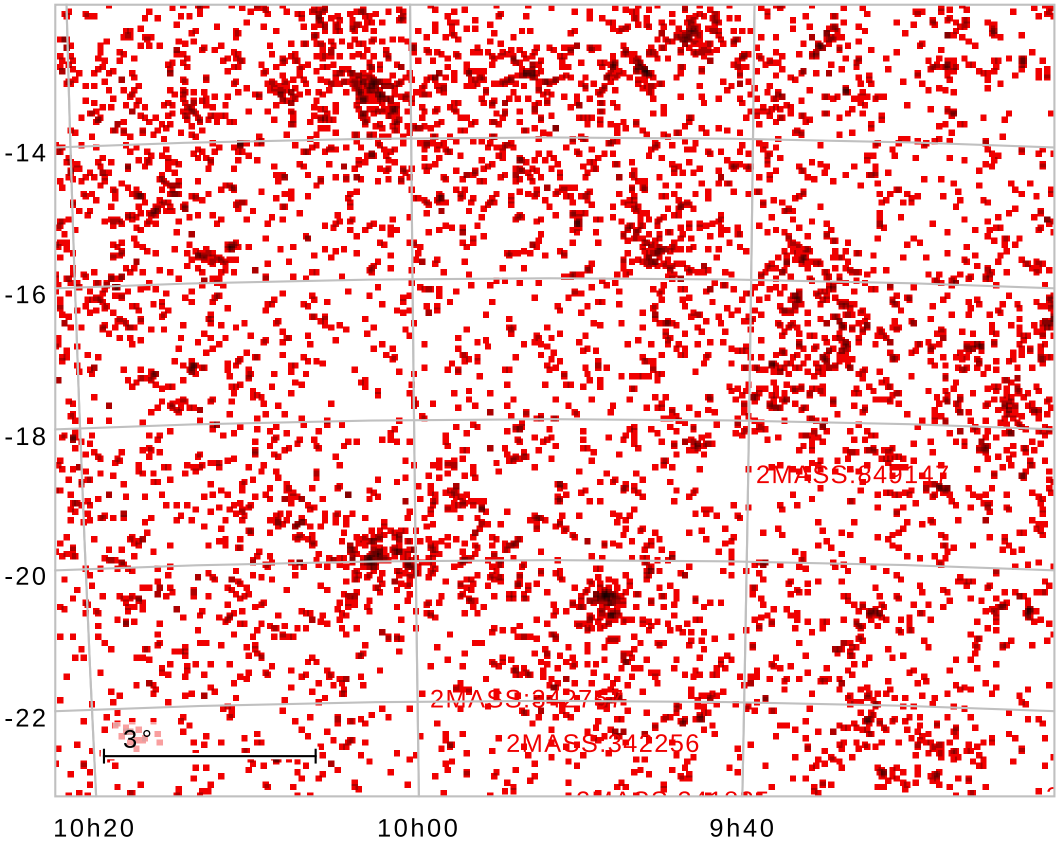
Visualisation Strengths

- Scalability
  - Multi-million row plots
  - No special data preparation required
  - Low memory usage
  - Multithreaded rendering
- Flexibility
  - Many different plot types
  - Many different configuration options
- Interactivity
  - Navigate round a 2d/3d plot
  - Smooth high/low point density transition
  - Config option changes instantly visible
- Linked Views
  - Select in one view, see inclusion in others
  - See information about selected points



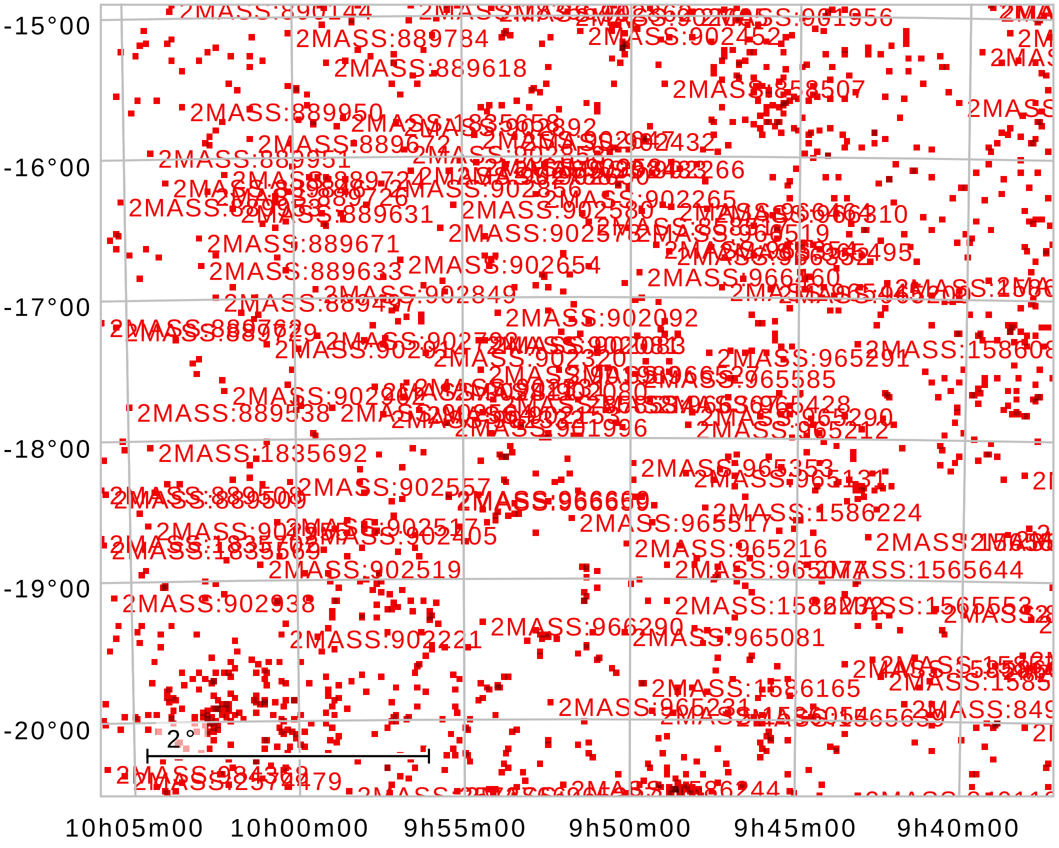
Visualisation Strengths

- Scalability
  - Multi-million row plots
  - No special data preparation required
  - Low memory usage
  - Multithreaded rendering
- Flexibility
  - Many different plot types
  - Many different configuration options
- Interactivity
  - Navigate round a 2d/3d plot
  - Smooth high/low point density transition
  - Config option changes instantly visible
- Linked Views
  - Select in one view, see inclusion in others
  - See information about selected points



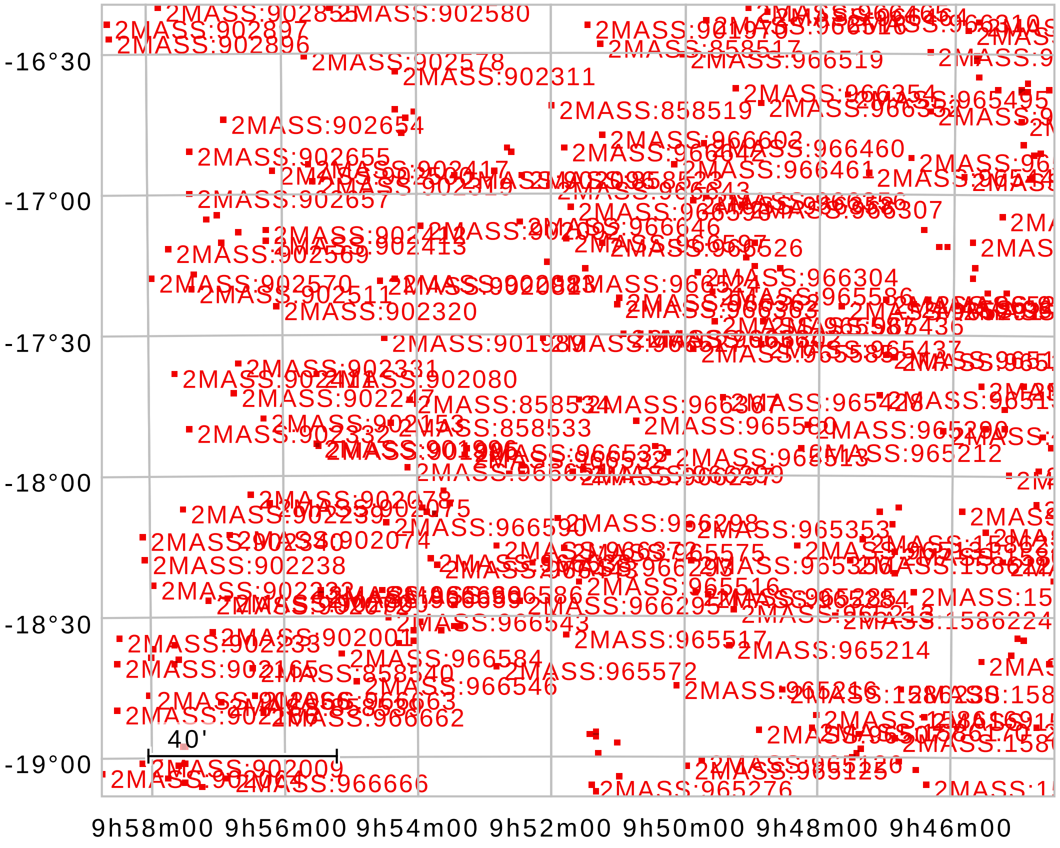
Visualisation Strengths

- Scalability
  - Multi-million row plots
  - No special data preparation required
  - Low memory usage
  - Multithreaded rendering
- Flexibility
  - Many different plot types
  - Many different configuration options
- Interactivity
  - Navigate round a 2d/3d plot
  - Smooth high/low point density transition
  - Config option changes instantly visible
- Linked Views
  - Select in one view, see inclusion in others
  - See information about selected points



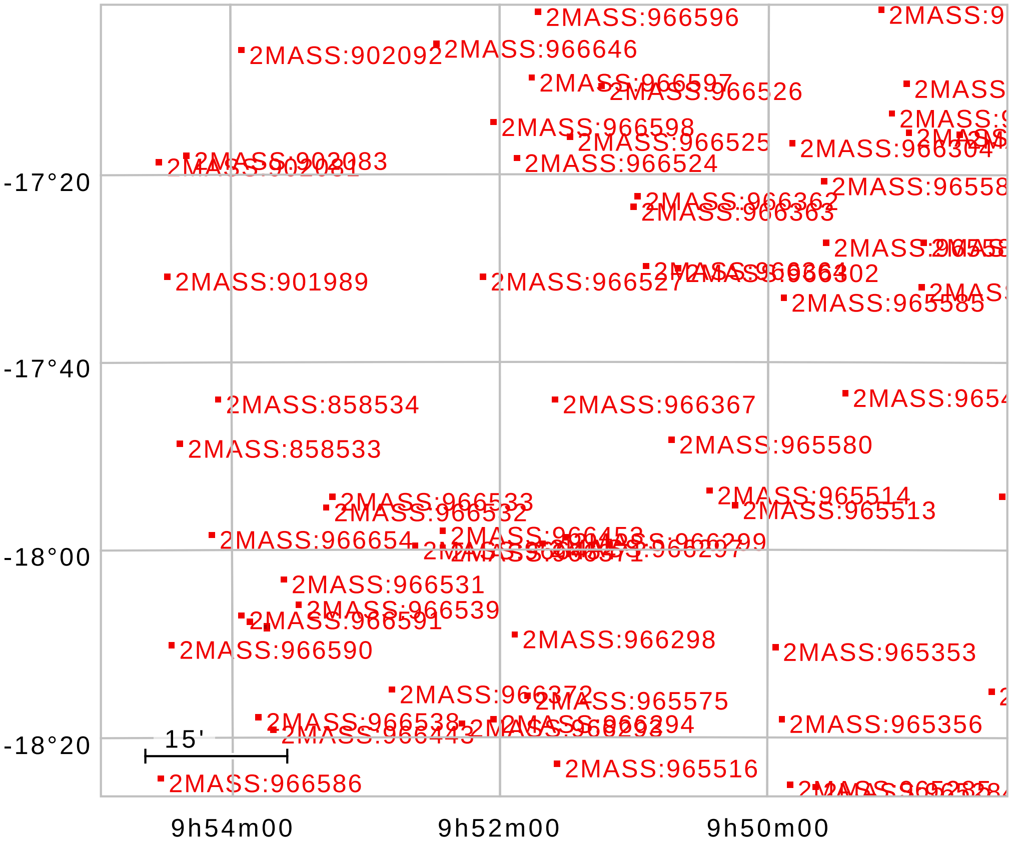
Visualisation Strengths

- Scalability
  - Multi-million row plots
  - No special data preparation required
  - Low memory usage
  - Multithreaded rendering
- Flexibility
  - Many different plot types
  - Many different configuration options
- Interactivity
  - Navigate round a 2d/3d plot
  - Smooth high/low point density transition
  - Config option changes instantly visible
- Linked Views
  - Select in one view, see inclusion in others
  - See information about selected points



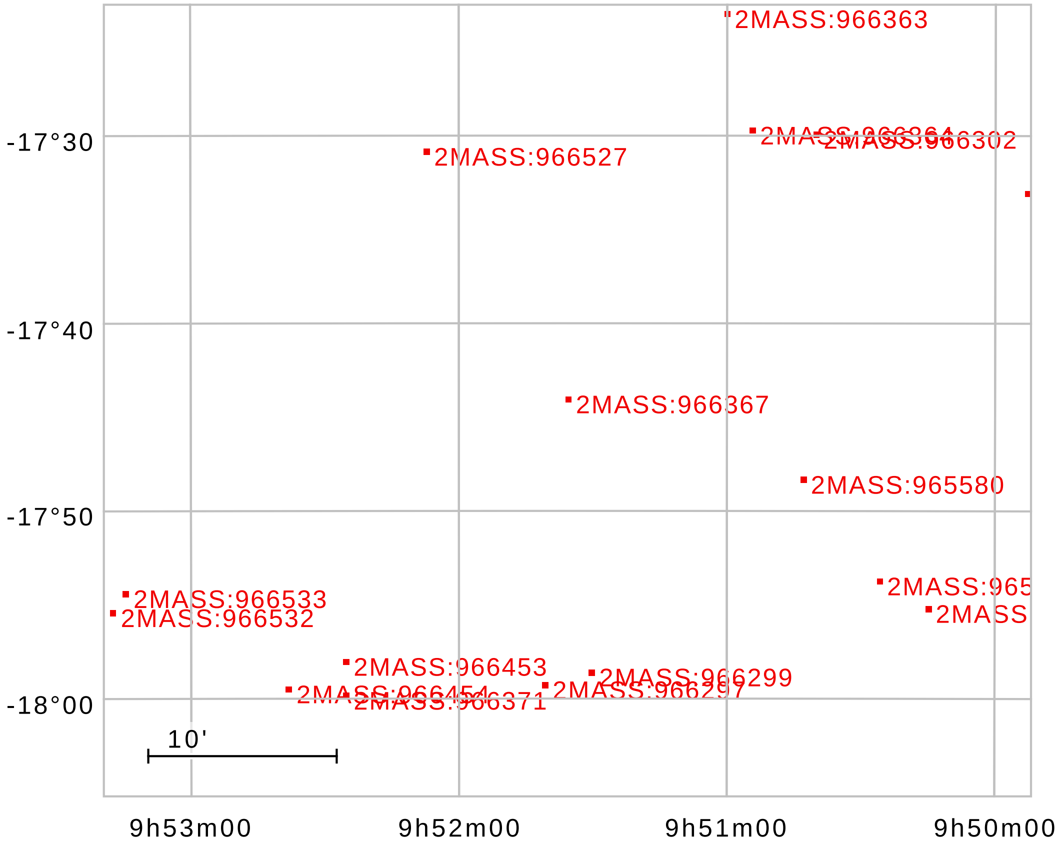
Visualisation Strengths

- Scalability
  - Multi-million row plots
  - No special data preparation required
  - Low memory usage
  - Multithreaded rendering
- Flexibility
  - Many different plot types
  - Many different configuration options
- Interactivity
  - Navigate round a 2d/3d plot
  - Smooth high/low point density transition
  - Config option changes instantly visible
- Linked Views
  - Select in one view, see inclusion in others
  - See information about selected points



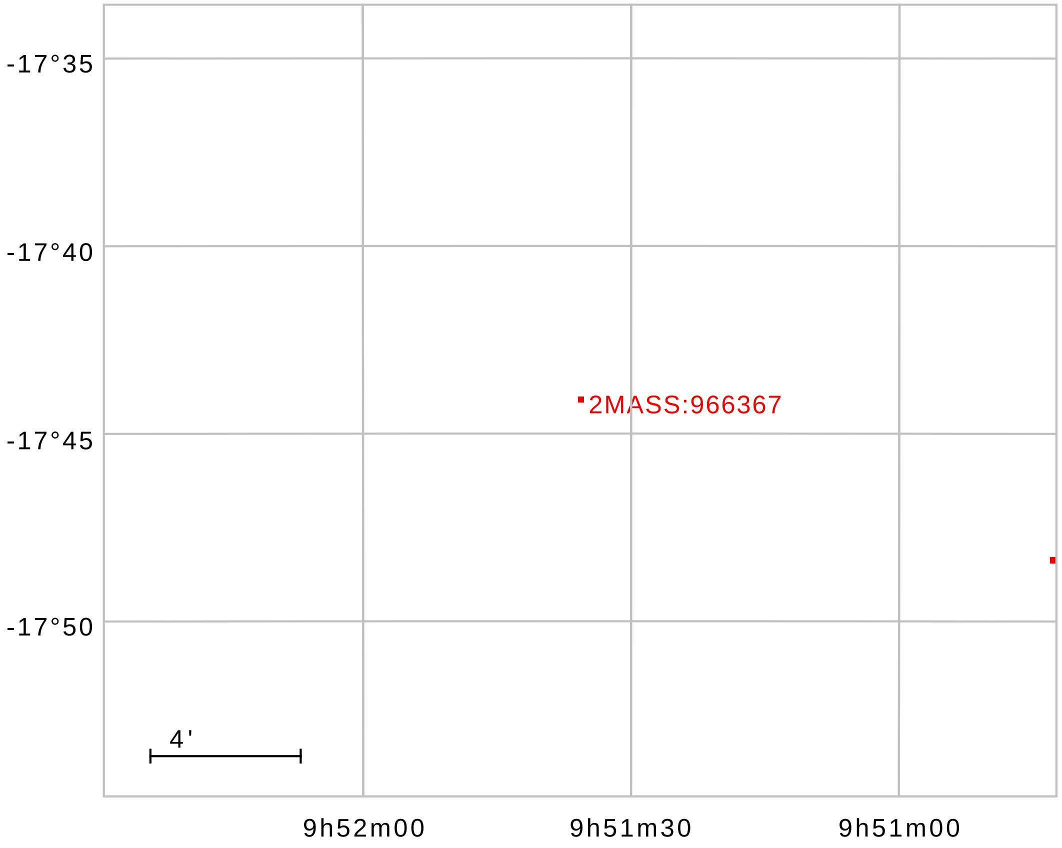
Visualisation Strengths

- Scalability
  - Multi-million row plots
  - No special data preparation required
  - Low memory usage
  - Multithreaded rendering
- Flexibility
  - Many different plot types
  - Many different configuration options
- Interactivity
  - Navigate round a 2d/3d plot
  - Smooth high/low point density transition
  - Config option changes instantly visible
- Linked Views
  - Select in one view, see inclusion in others
  - See information about selected points



Visualisation Strengths

- Scalability
  - Multi-million row plots
  - No special data preparation required
  - Low memory usage
  - Multithreaded rendering
- Flexibility
  - Many different plot types
  - Many different configuration options
- Interactivity
  - Navigate round a 2d/3d plot
  - Smooth high/low point density transition
  - Config option changes instantly visible
- Linked Views
  - Select in one view, see inclusion in others
  - See information about selected points



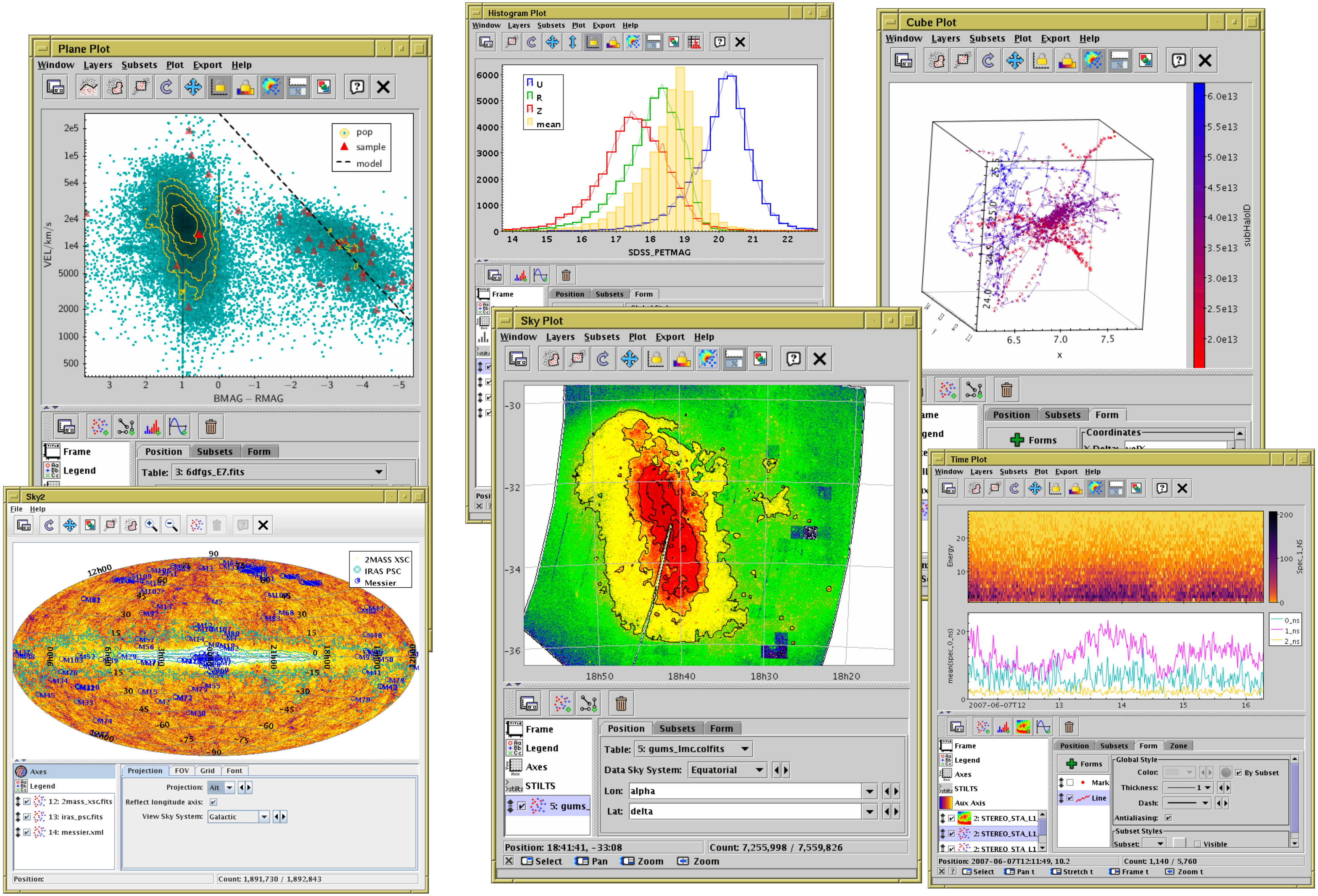


Table: 3: 6dfgs_E7.fits

Projection: Ait

Reflect longitude axis: ☑

View Sky System: Galactic

☑ 12: 2mass_xscfits

☑ 13: iras_psc.fits

☑ 14: messier.xml

Position:

Count: 1,891,730 / 1,892,843

☑ 5: gums_

Table: 5: gums_lmc.colfits

Data Sky System: Equatorial

Lon: alpha

Lat: delta

Position: 18:41:41, -33:08

Count: 7,255,998 / 7,559,826

☑ 2: STEREO_STA_L1

Color: ☑ By Subset

Thickness: 1

Dash:

Antialiasing: ☑

Subset: □ Visible

Position: 2007-06-07T12:11:49, 10.2

Count: 1,140 / 5,760

TOPCAT frequently asked question:

“Can I run it in a browser?”

TOPCAT frequently asked question:

"Can I run it in a browser?"

NO

TOPCAT frequently asked question:

"Can I run it in a browser?"

NO

Web applications are nice ...

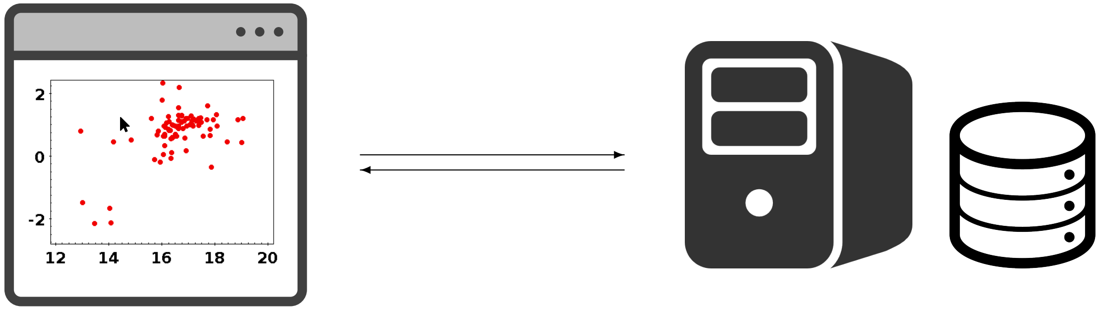
- No installation required!

but TOPCAT wouldn't make a good web application:

- GUI considerations: *too many windows!*
- Local data access issues: *memory mapping forbidden by sandbox*
- (also I don't want to rewrite it all in JavaScript)

... but maybe some server-side functionality would make sense
→ interactive visualisation

Remote Visualisation: Data Transfer



Tabular data ⇒ scatter plots

Two basic approaches:

Smart Client: Transfer data coords, once

- ▷ Server sends all coordinate data to browser, once
- ▷ Code in browser handles interactive navigation and (re-)rendering
- ▷ Works well for modest size datasets (smooth animation)
- ▷ Works badly/fails for very large datasets (download time, browser memory)
- ▷ Most available javascript plotting libraries do this

Dumb Client: Transfer rendered images, every frame

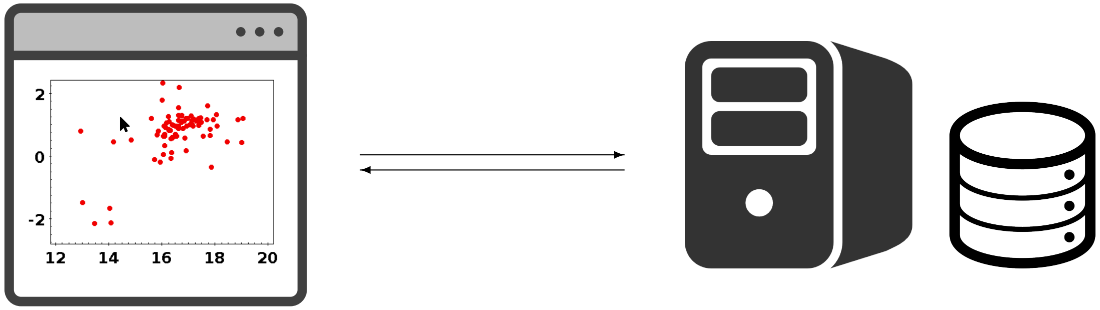
- ▷ Server sends image data (pixels) to browser
- ▷ Code in browser asks for updated image on every navigation action
- ▷ Works OK for any size datasets as long as server can handle them (jerky animation, but scales to millions of points)

Which is best?

- For a few thousand points **Smart Client** works better
- For a ~~few hundred thousand~~[*] million points or more, you have to use **Dumb Client**
- Hybrid options (e.g. transfer density map data, render on browser)?
  - ▷ ...maybe, but hard to get good zooming right

[*] see Aladin-Lite talk O1-68

Remote Visualisation: Data Transfer



Tabular data ⇒ scatter plots

Two basic approaches:

Smart Client: Transfer data coords, once

- ▷ Server sends all coordinate data to browser, once
- ▷ Code in browser handles interactive navigation and (re-)rendering
- ▷ Works well for modest size datasets (smooth animation)
- ▷ Works badly/fails for very large datasets (download time, browser memory)
- ▷ Most available javascript plotting libraries do this

Dumb Client: Transfer rendered images, every frame

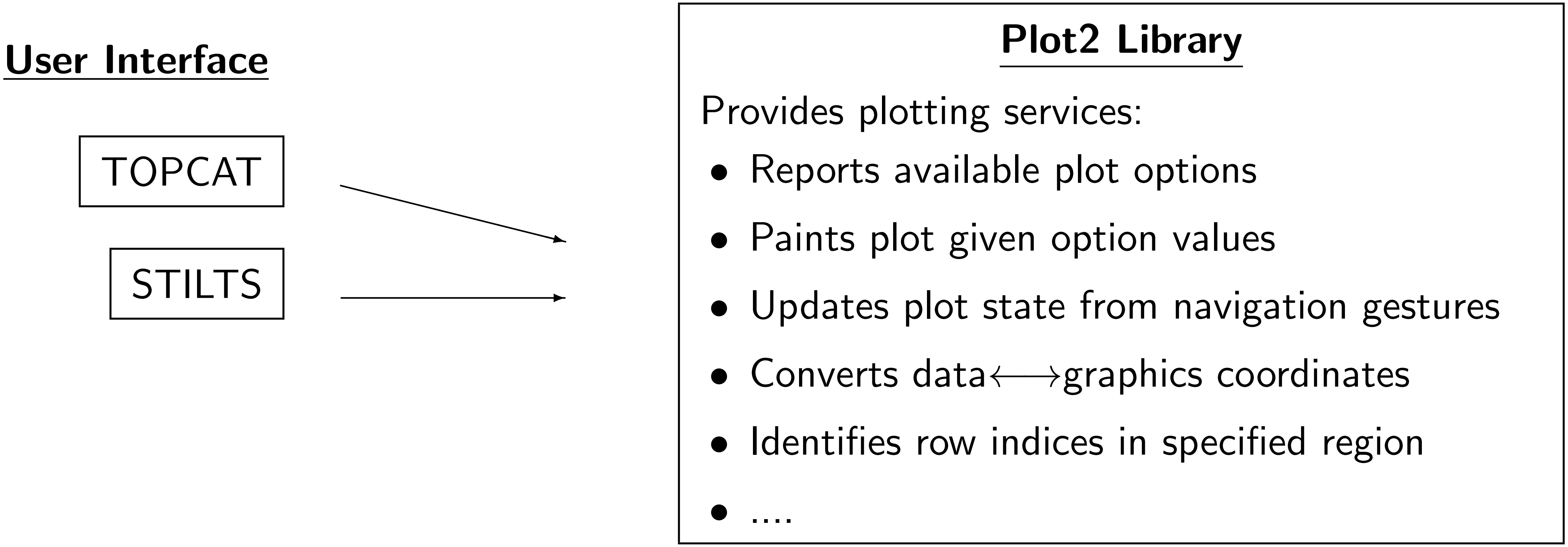
- ▷ Server sends image data (pixels) to browser
- ▷ Code in browser asks for updated image on every navigation action
- ▷ Works OK for any size datasets as long as server can handle them (jerky animation, but scales to millions of points)

Which is best?

- For a few thousand points **Smart Client** works better
- For a ~~few hundred thousand~~[★] million points or more, you have to use **Dumb Client**
- Hybrid options (e.g. transfer density map data, render on browser)?
  - ▷ ...maybe, but hard to get good zooming right

[★] see Aladin-Lite talk O1-68

TOPCAT/STILTS Visualisation Architecture



User Interface

TOPCAT

STILTS

Plot2 Library

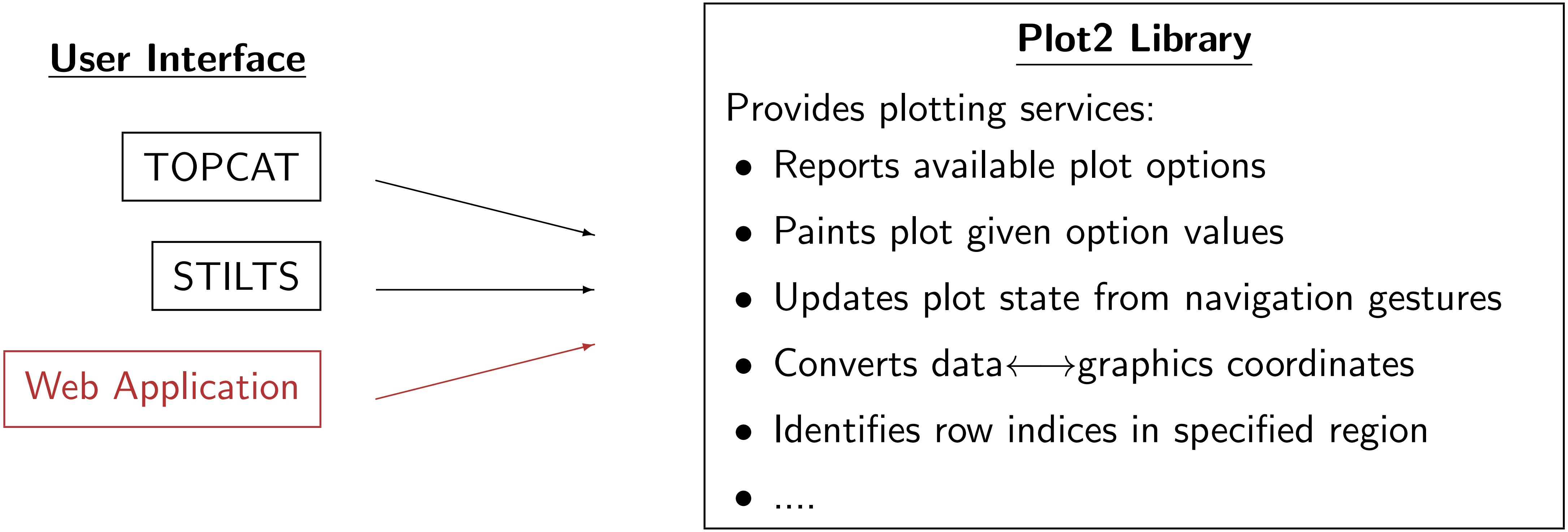
Provides plotting services:

- Reports available plot options
- Paints plot given option values
- Updates plot state from navigation gestures
- Converts data ⟷ graphics coordinates
- Identifies row indices in specified region
- ....

Well, more or less.

- There are some additional client-specific arrangements
  - Data caching, session management, ...
- But clients treat all plots the same
  - (Almost) no UI-side code for specifics of 2D/3D/sky/scatter/shading/density/histogram/...
  - So when adding a new (web app) UI, complexity does not scale with (large) number of existing plot options

TOPCAT/STILTS Visualisation Architecture



User Interface

TOPCAT

STILTS

Web Application

Plot2 Library

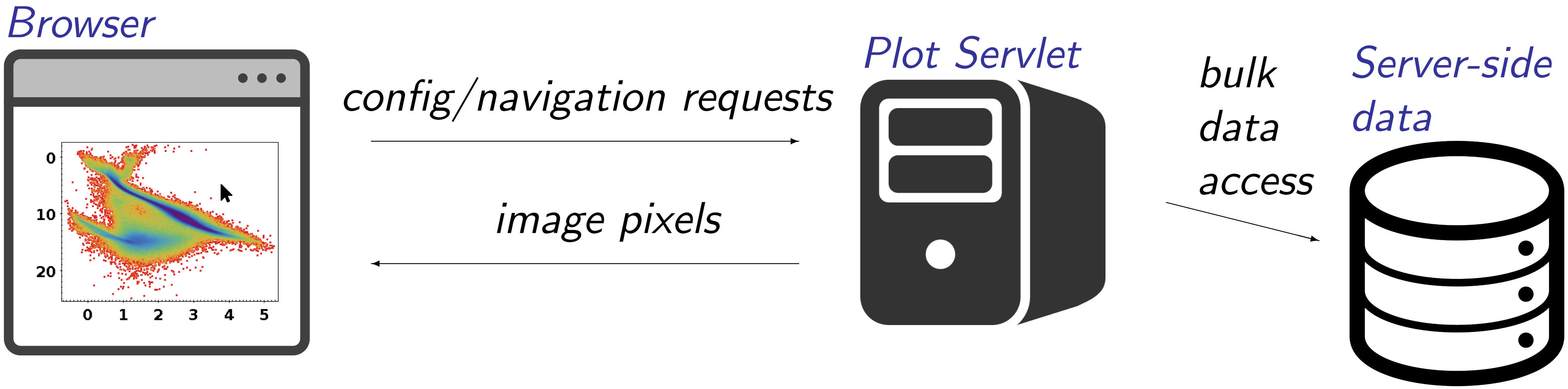
Provides plotting services:

- Reports available plot options
- Paints plot given option values
- Updates plot state from navigation gestures
- Converts data⟷graphics coordinates
- Identifies row indices in specified region
- ....

Well, more or less.

- There are some additional client-specific arrangements
  - ▷ Data caching, session management, ...
- But clients treat all plots the same
  - ▷ (Almost) no UI-side code for specifics of 2D/3D/sky/scatter/shading/density/histogram/...
  - ▷ So when adding a new (web app) UI, complexity does not scale with (large) number of existing plot options

Remote Visualisation Architecture



Browser

config/navigation requests

image pixels

Plot Servlet

bulk data access

Server-side data

- Plot configuration: Initial request sets up plot session
- Navigation: User mouse gestures trigger requests for image updates
- Bulk data stays on server, image rendering is done on server
- Only rendered image (pixels/vectors) is transferred

  ⇒ Data transfer, browser resource usage does not scale with row count
- Dumb web client doesn't understand plot details

  ⇒ Web app complexity does not scale with plot options
- Some non-image endpoints also available
  - ▷ Image bounds in data coordinates
  - ▷ Visible row count
  - ▷ Graphics → data coordinate conversion (e.g. cursor position)
  - ▷ Row data for point nearest position (e.g. click to view point data)
- Various caching arrangements to improve performance
- User experience is typically a few frames per second *(YMMV)*

Example HTML

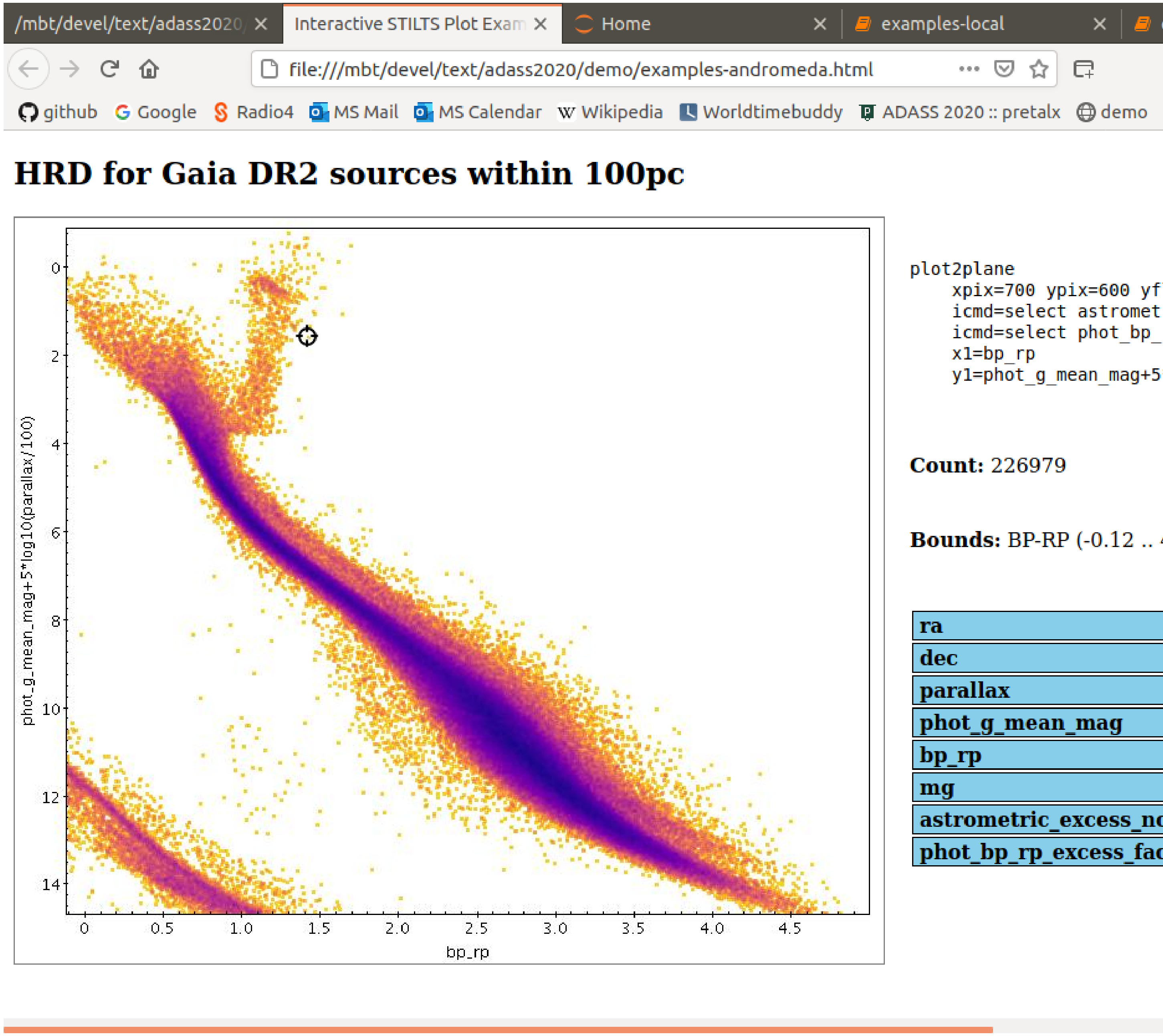
Insert plot in page by passing STILTS-like params to JS library function:

```
<html><body>
<script src="plot2Lib.js">
<script>
   onload = function() {
      var serverUrl = "plot";
      var plotNode = plot2.createPlotNode(serverUrl, plot2.wordsToPlotTxt([
          "plot2plane",
          "in=hrd-100pc.fits",
          "yflip=true",
          "icmd=select astrometric_excess_noise<1",
          "layer1=mark",
          "x1=bp_rp",
          "y1=phot_g_mean_mag+5*log10(parallax/100)",
          "shading1=density",
          "densemap1=plasma",
      ]));
      document.getElementById("hrd-plot").appendChild(plotNode);
      ...
   }
</script>

<h2>Herzsprung-Russell Diagram for sources within 100pc</h2>
<p>229k/338k sources plotted</p>
<div id="hrd-plot"></div>
   ...
</body></html>
```

Example: Preconfigured Plot

- Web page defines plot
- User can pan & zoom
- Clicking on point displays row data


/mbt/devel/text/adass2020/
Interactive STILTS Plot Exam
Home
examples-local
file:///mbt/devel/text/adass2020/demo/examples-andromeda.html
github
Google
Radio4
MS Mail
MS Calendar
Wikipedia
Worldtimebuddy
ADASS 2020 :: pretalx
demo
0
2
4
6
8
10
12
14
phot_g_mean_mag+5*log10(parallax/100)
0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
bp_rp

HRD for Gaia DR2 sources within 100pc

```
plot2plane
    xpix=700 ypix=600 yf
    icmd=select astromet
    icmd=select phot_bp_
    x1=bp_rp
    y1=phot_g_mean_mag+5
```

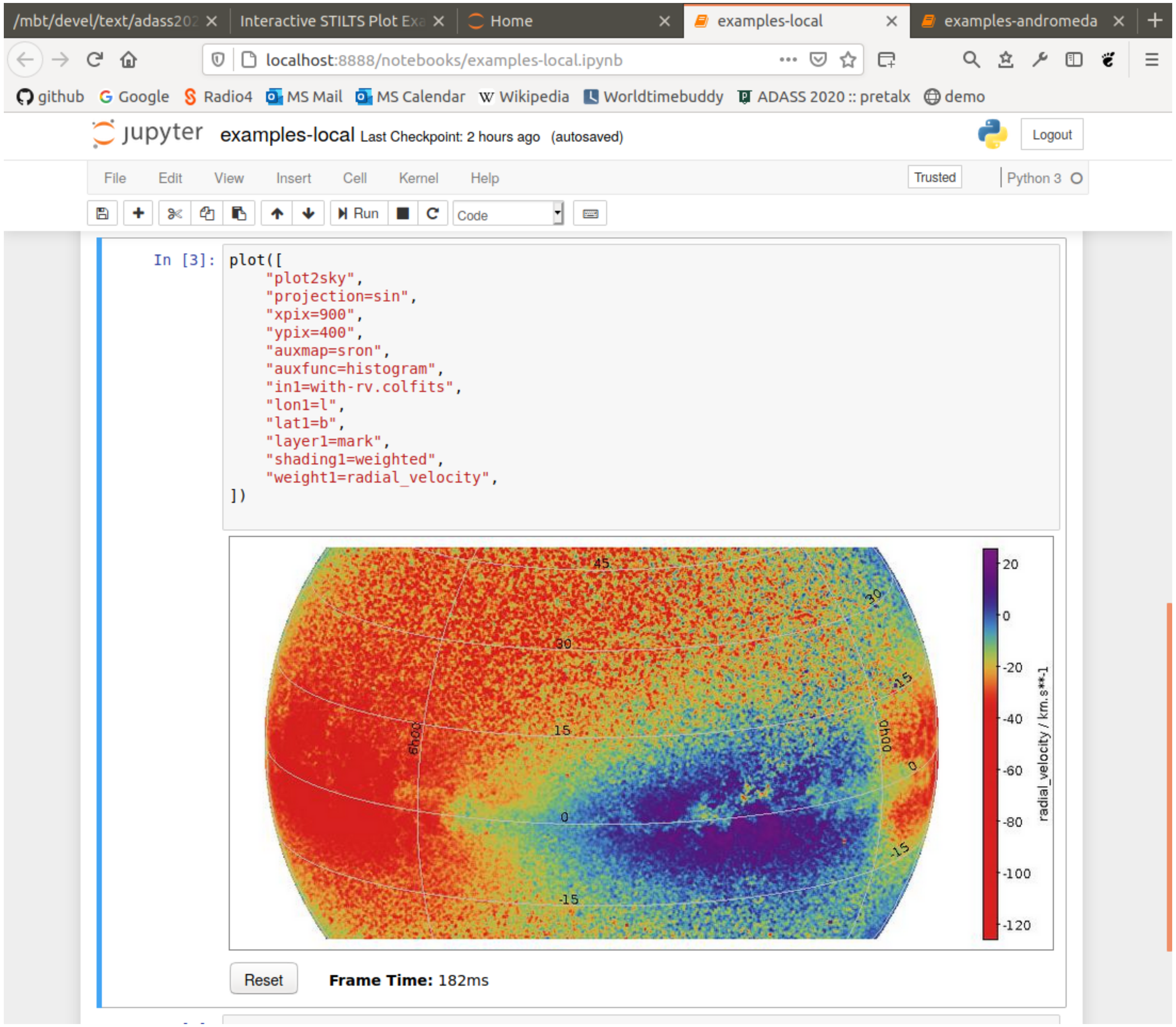
Count: 226979

Bounds: BP-RP (-0.12 .. 4

ra
dec
parallax
phot_g_mean_mag
bp_rp
mg
astrometric_excess_no
phot_bp_rp_excess_fac

Example: Jupyter Notebook

- Boilerplate code sets up `plot` function
- Point at either local or remote server
- Create/edit cells to display interactive plots



```
In [3]: plot([
            "plot2sky",
            "projection=sin",
            "xpix=900",
            "ypix=400",
            "auxmap=sron",
            "auxfunc=histogram",
            "in1=with-rv.colfits",
            "lon1=l",
            "lat1=b",
            "layer1=mark",
            "shading1=weighted",
            "weight1=radial_velocity",
        ])
```

Frame Time: 182ms

Usage Scenarios

Client usage possibilities:

- Data provider includes preconfigured plots in web pages
    - ▷ Archive query result page quick look
- Scientist includes preconfigured plots in web page
    - ▷ Interactive content related to research results
- Jupyter notebook client with configurable embedded plots
    - ▷ Plot local data on local server
    - ▷ Plot user results or fixed large tables on a science platform
- Custom client web app allows user to specify arbitrary plots

Available data is controlled by the service in all cases

Deployment

Service deployment options:

- Servlet for use in container:

  Class `uk.ac.starlink.ttools.server.PlotServlet`

- Stilts internal server (convenient for testing):

```
% stilts server port=8080 tablefactory=dirs:/mbt/data/plot2data
Server running at http://127.0.0.1:8080/stilts/
```

- Docker image https://hub.docker.com/r/mbtaylor/plotserv:

```
% docker pull mbtaylor/plotserv
% docker run -dp 8080:8080
             --mount type=bind,src=/my/data/directory,dst=/data,readonly
             --mount type=tmpfs,dst=/tmp,tmpfs-size=2G
             mbtaylor/plotserv
```

Resource Requirements

Server requirements:

- Data files in FITS (or other STIL-friendly format; JDBC should work but not tested)
  — no data preparation or indexing required
- Disk cache, small or large (caches prepared column data and initial image on first plot)
- CPU (multiple cores good) & disk I/O (SSDs good)

Client requirements

- Any browser
- Minimal resource usage: low CPU, low memory, fairly low bandwidth
- Good network latency helps though

Status and Future

Working but experimental

- Available in recent STILTS release v3.3

  http://www.starlink.ac.uk/stilts/

- Not tested under heavy multi-user loads
- Possibilities for improved functionality:
  - ▷ Tweak caching arrangements
  - ▷ Improve data access/security options
  - ▷ Improve session management (store more state on client)
  - ▷ Improve client side UI javascript
  - ▷ Add web app UI options to change plot config as well as navigate
    - ○ Adjust colour maps, marker shape/size, sky projection, line thickness, binning, ...
    - ○ More TOPCAT-like experience
- Wait and see what users want

Summary

Remote interactive visualisation in a browser:

- Large datasets (multi-million row)
- Many astro-friendly plot types/options
- Can focus on individual points
- Modest client resource requirements

Interested in deploying it?

- Talk to me:
  - ▷ Discord: `mbtaylor#7395`
  - ▷ Email: `m.b.taylor@bristol.ac.uk`
- Run it locally:
  - ▷ Download `http://www.starlink.ac.uk/stilts/stilts.jar`
  - ▷ Run `java -jar stilts.jar server`
- Play with a running instance:
  - ▷ `https://andromeda.star.bristol.ac.uk:8080/plotserv/`
  - ▷ `http://andromeda.star.bristol.ac.uk:8082/stilts/plot/ex-plots.html`

    (but if you do it all at once I don't know what will happen)