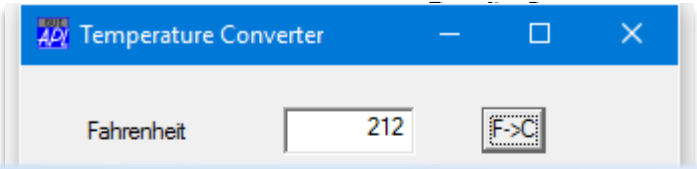
APL demonstration 1975
Imperial College London

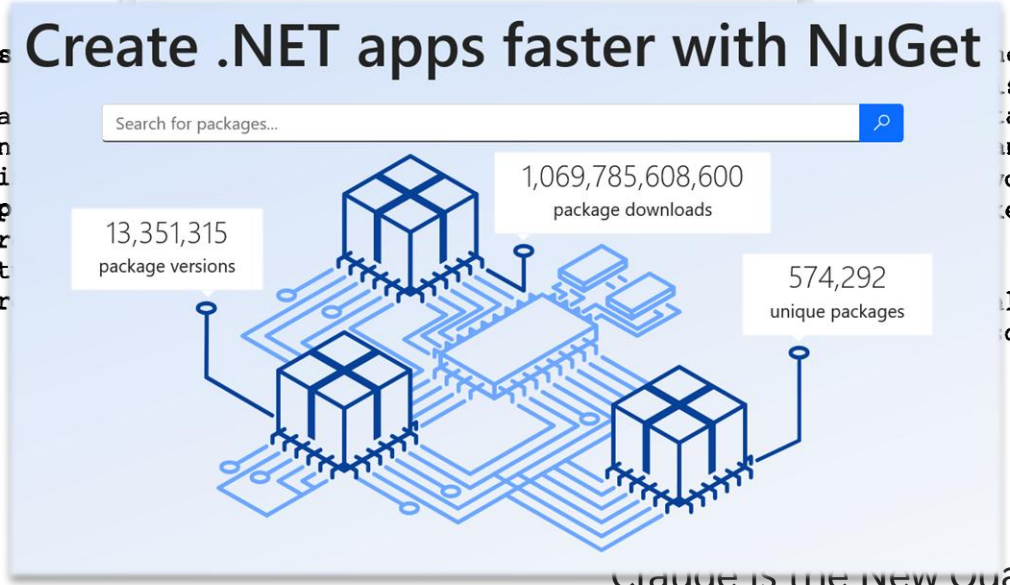


System Interaction

SM: A Full-Screen Manager for Dyalog APL

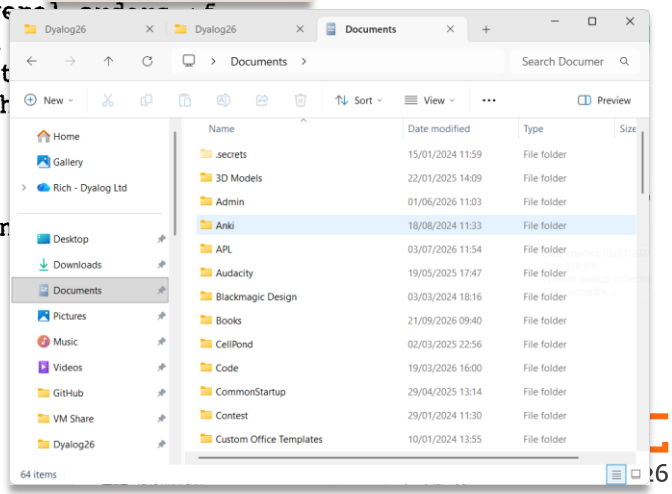
A. D. Curtin, J. M. Scholes



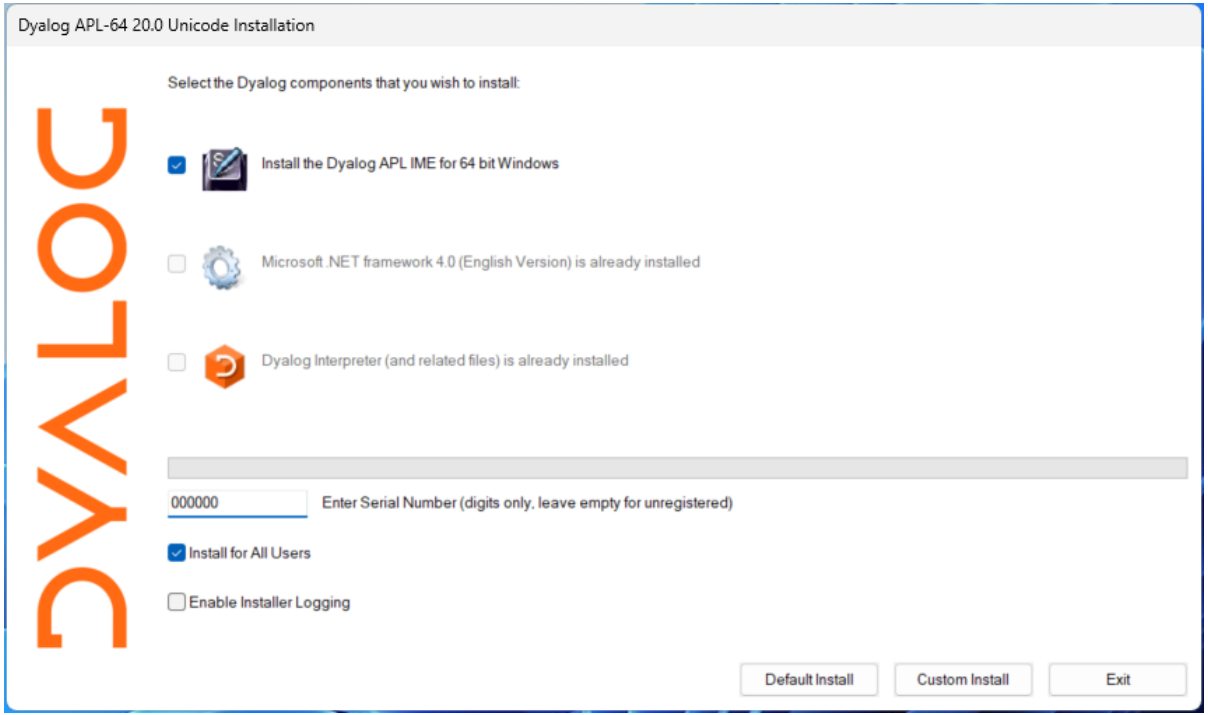




~↑*⍳⊤∇ ⍺*⍰



Claude is the New Quad



Claude is the New Quad

But when I had my document open and was happily typing away, I found myself wanting to save my work and I instinctively reach for my favourite shortcut: Ctrl S

But when I had my document open and was happily typing away, I found myself wanting to save my work and I instinctively reach for my favourite shortcut: Ctrl S

But when I had my document open and was happily typing away, I found myself wanting to save my work and I instinctively reach for my favourite shortcut: Ctrl S

oh.

Typing glyphs

APL uses a range of special graphic symbols to represent most functions and operators. While keyboard mappings become memorized over time, entering APL characters can frustrate the beginner. However, a study involving high school students found that typing and using APL characters did not hinder the students in any measurable way. There are several convenient ways to enter the glyphs.

Contents [hide]

1 Hardware
2 By platform
2.1 Web
2.2 Text editors
2.3 Linux
2.4 Android
2.5 Windows
2.5.1 Troubleshooting
2.6 macOS
2.7 iOS
3 By method
3.1 Shifting key
3.2 Prefix key
3.3 Long-press
3.4 Keyword look-up
3.5 ASCII symbol composition
3.6 Text replacement
3.7 On-screen language bar
3.8 LaTeX
3.8.1 Listings
3.8.2 Metafont
3.9 plainTeX
4 References

Hardware [edit]



Once the keyboard map is configured to one's liking, it may be beneficial (especially for new users learning APL) to have a keyboard with APL symbols. There are a few solutions to this, such as purchasing a dedicated desktop keyboard, key caps for mechanical keyboards, or a keyboard decal.

layouts.md	en GB layout and Windows instructions	last year

README | Code of conduct | Contributing | MIT license | Security

Kanata APL Layouts

International keyboard layouts for APL.



- About
- Installing kanata
- Running a layout
- Available layouts
  - Shifting key layouts
  - Prefix key layouts
- Typing APL
- Optional modules

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 1

rikedyp Rich Park

```
88      (deflayermap APL-shift
89        Backquote     (unicode ⍁)
90        1             (unicode ⌶)
91        2             (unicode ⍫)
92        3             (unicode ⍒)
93        4             (unicode ⍋)
94        5             (unicode ⌽)
95        6             (unicode ⍉)
96        7             (unicode ⊖)
97        8             (unicode ⍟)
98        9             (unicode ⍱)
99        0             (unicode ⍲)
100       -             (unicode !)
```

But when I had my document open and was happily typing away, I found myself wanting to save my work and I instinctively reach for my favourite shortcut: Ctrl S

So now I can select the text above to delete it.
I hold Shift and ↑

But when I had my document open and was happily typing away, I found myself wanting to save my work and I instinctively reach for my favourite shortcut:
Ctrl S

So now I can select the text above to delete it.
I hold Shift and ↑

But when I had my document open and was happily typing away, I found myself wanting to save my work and I instinctively reach for my favourite shortcut:
Ctrl S

So now I can select the text above to delete it.
I hold Shift and ↑

But when I had my document open and was happily typing away, I found myself wanting to save my work and I instinctively reach for my favourite shortcut: Ctrl S

So now I can select the text above to delete it.
I hold Shift and ↑

But when I had my document open and was happily typing away, I found myself wanting to save my work and I instinctively reach for my favourite shortcut: Ctrl S

So now I can select the text above to delete it.
I hold Shift and ↑

But when I had my document open and was happily typing away, I found myself wanting to save my work and I instinctively reach for my favourite shortcut: Ctrl S

So now I can select the text above to delete it.
I hold Shift and ↑

What's cooking, Rich?

How can I help you today?

+ Chat Cowork

Opus 5.5 Medium

Read a file, searched the web, read a page

Your config validated fine, so this is behavioural. The culprit is line 17:

```
___ (multi lshift _)
```

While lshift is held, every key in the `shift` layer resolves to a `multi` action rather than a plain key. Kanata only passes hardware auto-repeat through for keys whose resolved action is a plain key output; complex actions (`multi`, output chords like `S-x`, etc.) get their repeat events dropped. So you get one ↑ and nothing more. It isn't specific to arrows — holding

Write a message...
Opus 5 High
Claude is AI and can make mistakes. Please double-check responses.

Untitled

File Edit View

H1

Ln 1, Col 1 0 characters Plain text 440% Windows (CRLF) UTF-8

AltGr compositions

Press `AltGr` with one of seven keys to start a dead key sequence. The subsequent keystroke produces the composed character.

Start with	Dead key	Gives you
`AltGr` + `W`	^	superscripts, circumflex accents, the quad family
`AltGr` + `R`	¨	umlauts, diaeresis operators
`AltGr` + `Y`	~	tildes, fractions, currency
`AltGr` + `U`	_	subscripts, underbar glyphs, circled letters
`AltGr` + `A`	´	acute accents
`AltGr` + `S`	`	grave accents, box drawing on the number row
`AltGr` + `Space`		typographic spaces and dashes

Box drawing on the keypad

Hold an APL key and use the keypad. The characters are arranged the way they look on screen:

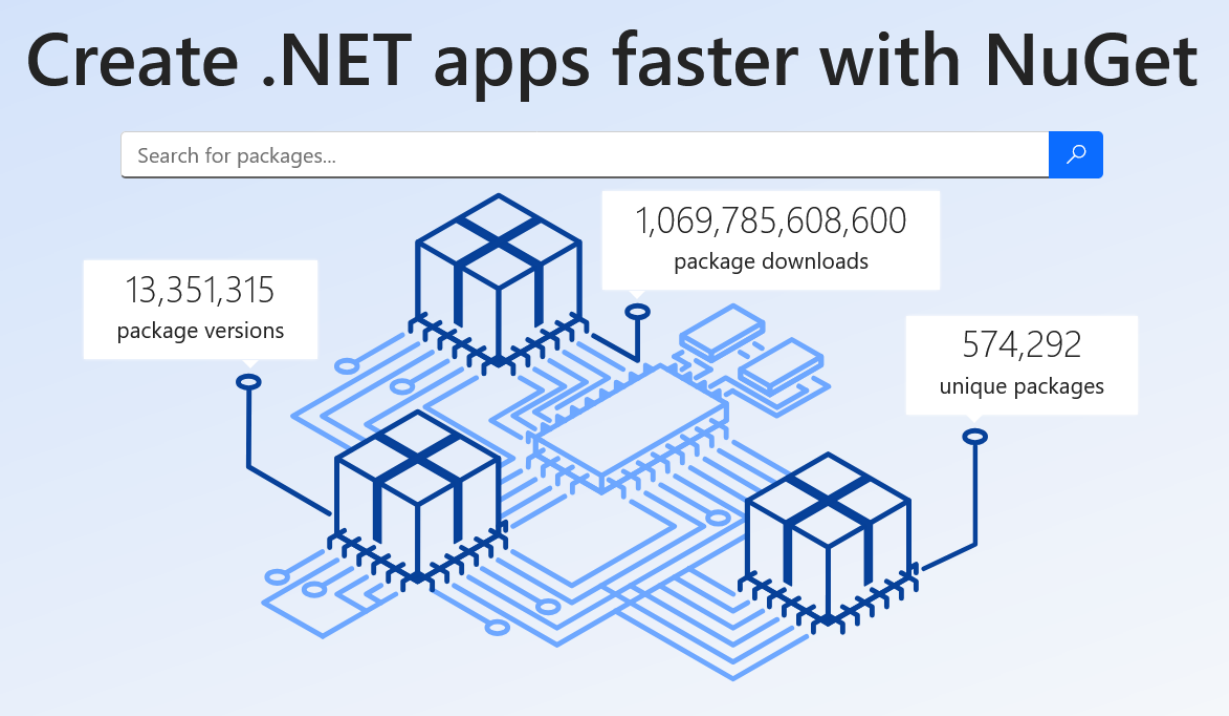
```
7 ┌   8 ┬   9 ┐
4 ├   5 ┼   6 ┤
1 └   2 ┴   3 ┘
  0 ─       . │
```

So `Caps` + `Numpad6` gives ┼. NumLock makes no difference, and the keypad behaves normally when no APL key is held.

This is enabled by two lines in `en-GB_APL.kbd`:

```
(deflayermap APL
  ...
  (t! numpad-box)               ;; inside the APL layer
```

Using .NET



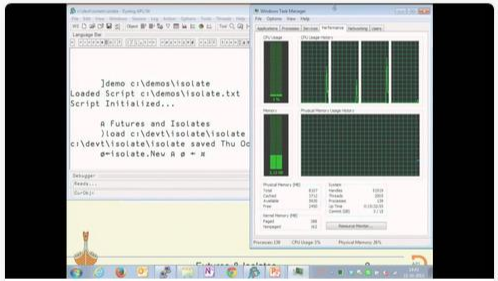
Claude is the New Quad

Video Library parallel Search

Videos Events Presenters Terms of Use

Showing 22 results

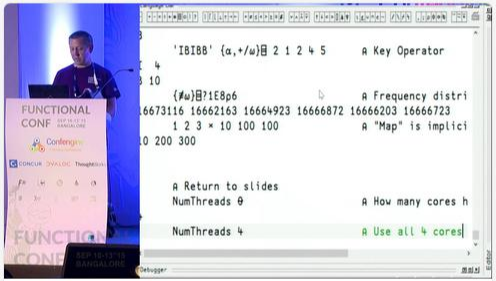
Sort: Relevance



Parallel Language Features in Version 14.0

Morten Kromberg

Oct 2013 in Dyalog '13



Parallel Programming in Dyalog using Futures and Isolates

Morten Kromberg

Sept 2015 in Functional Conf 2015



Segmented Scans and Nested Data Parallelism

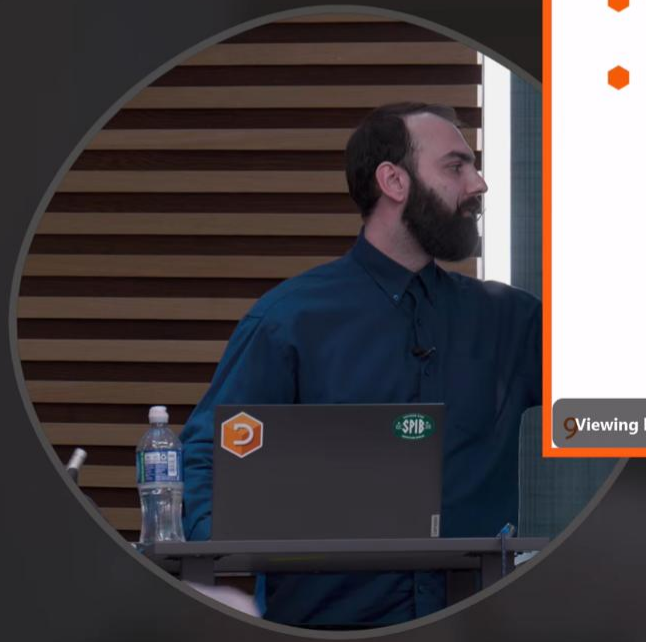
Andrzej Filinski

Oct 2012 in Dyalog '12









Tatin & NuGet Packages

- NuGet: library of .NET tools
- Can use from APL with]USING
- Porter2Stemmer NuGet package

```
    Stem¨(
        'analytic'
        'analytics'
        'analytically'
    )
analyt  analyt  analyt
```

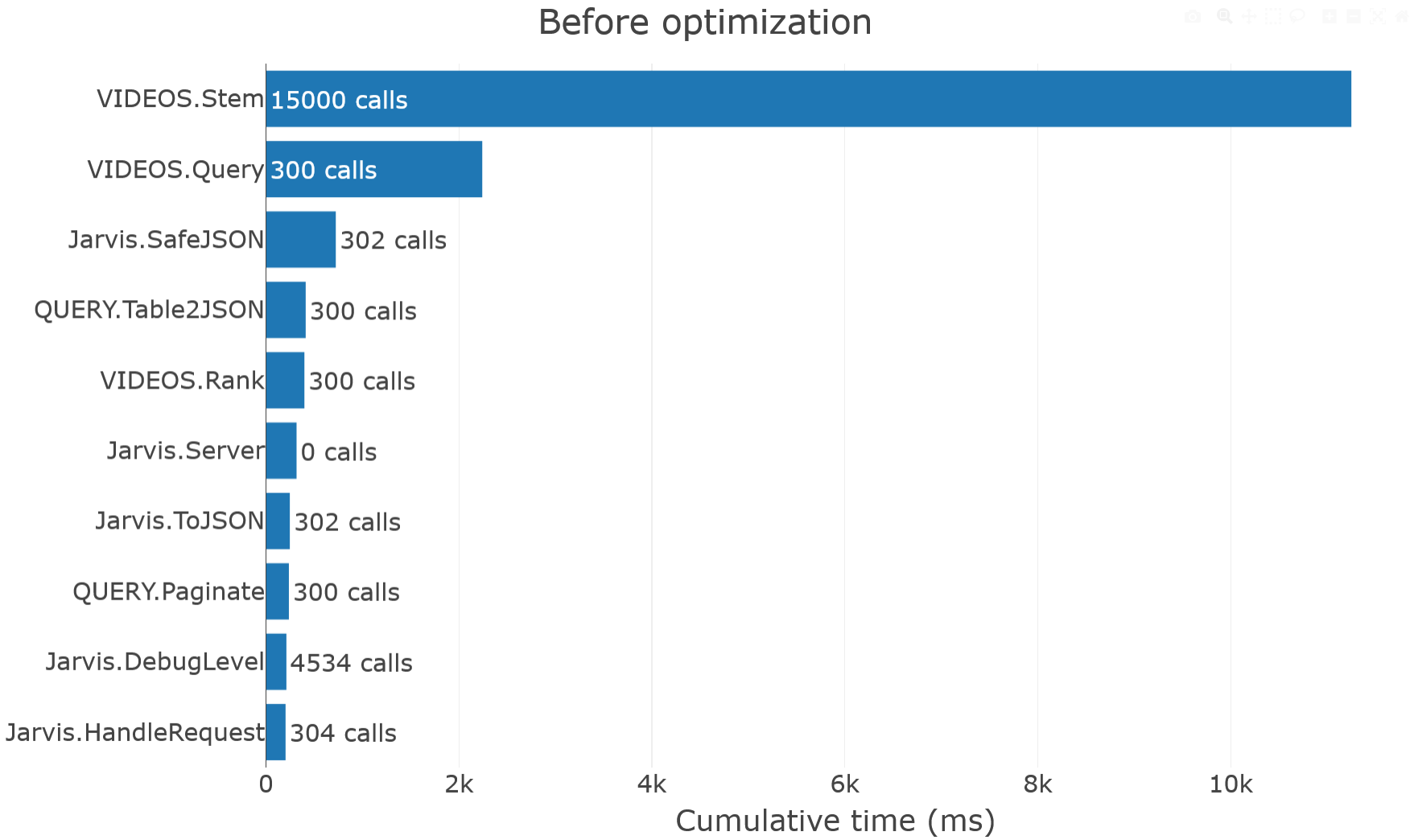
NuGet Porter2Stemmer

```
      Stem¨(
          'analytic'
          'analytics'
          'analytically'
      )
analyt   analyt   analyt
```

NuGet Porter2Stemmer

```
      stemmer←⎕NEW Porter2Stemmer.EnglishPorter2Stemmer
      words←'alphabetical' 'statically' 'coordinated'
      {(stemmer.Stem∘⊂⍤⊃⍵).Value} words
alphabet  static  coordin

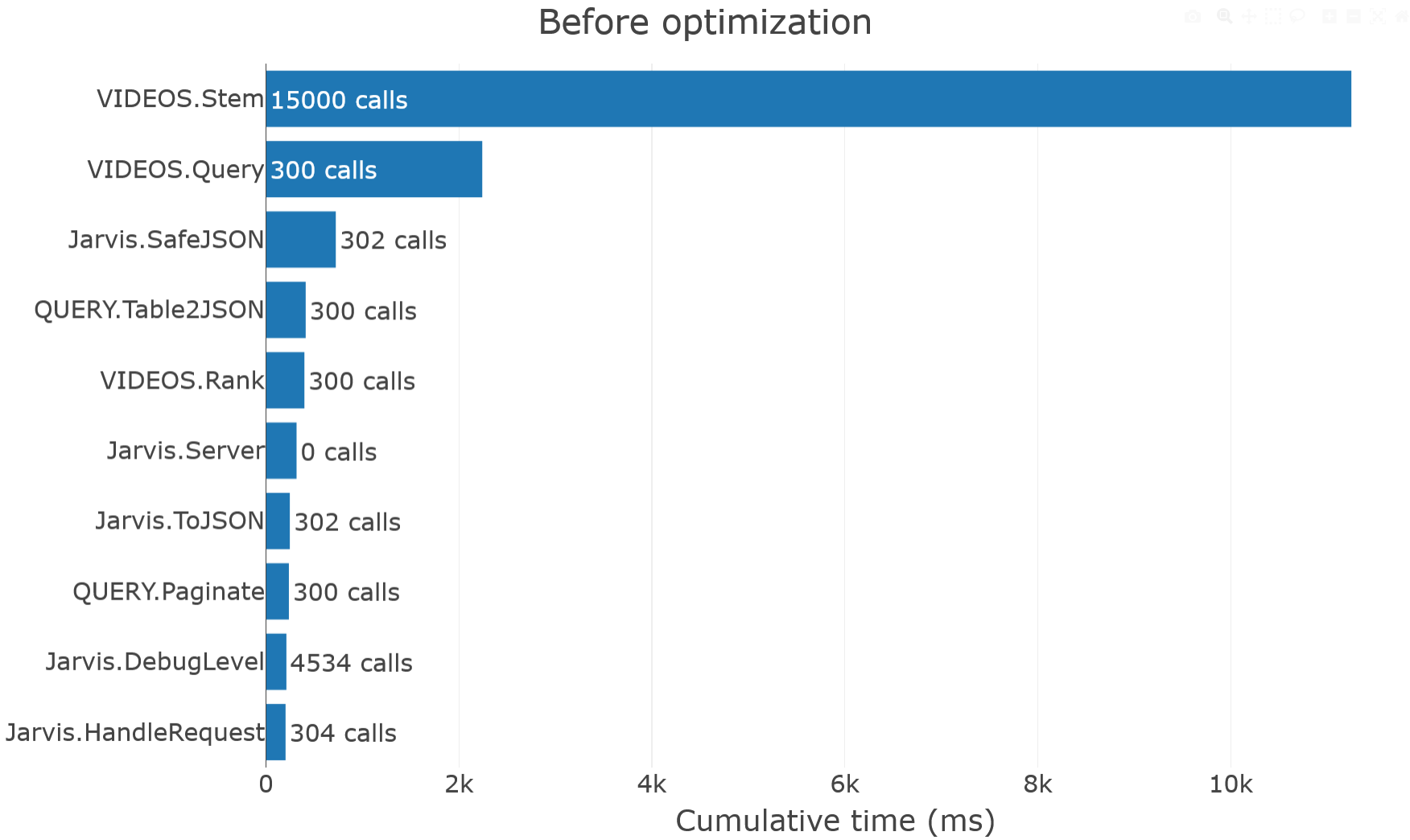
      DCMS.Stemming.Batch.StemAll ⊂words
alphabet  static  coordin
```

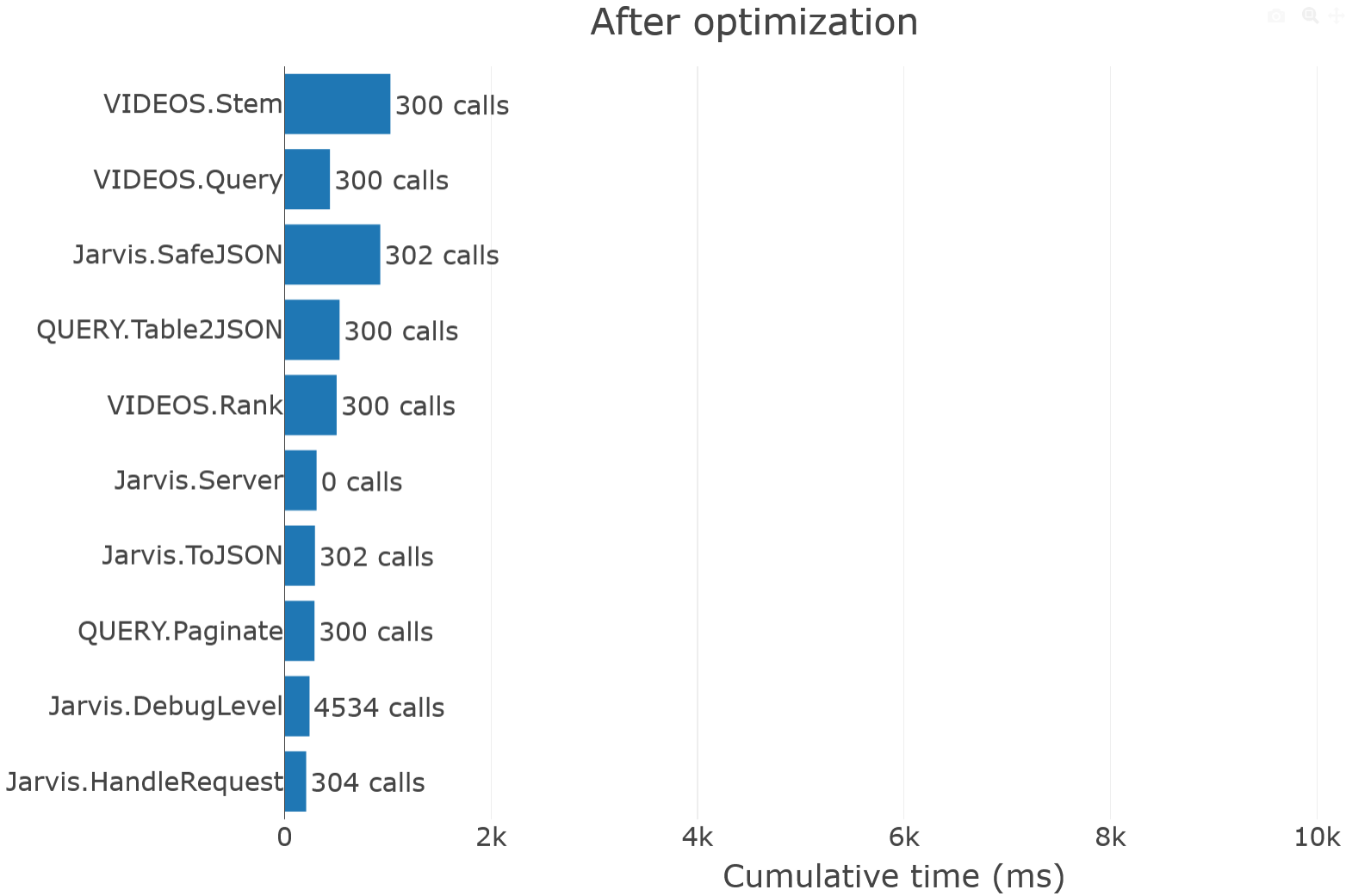

Before optimization
VIDEOS.Stem
15000 calls
VIDEOS.Query
300 calls
Jarvis.SafeJSON
302 calls
QUERY.Table2JSON
300 calls
VIDEOS.Rank
300 calls
Jarvis.Server
0 calls
Jarvis.ToJSON
302 calls
QUERY.Paginate
300 calls
Jarvis.DebugLevel
4534 calls
Jarvis.HandleRequest
304 calls
0
2k
4k
6k
8k
10k
Cumulative time (ms)

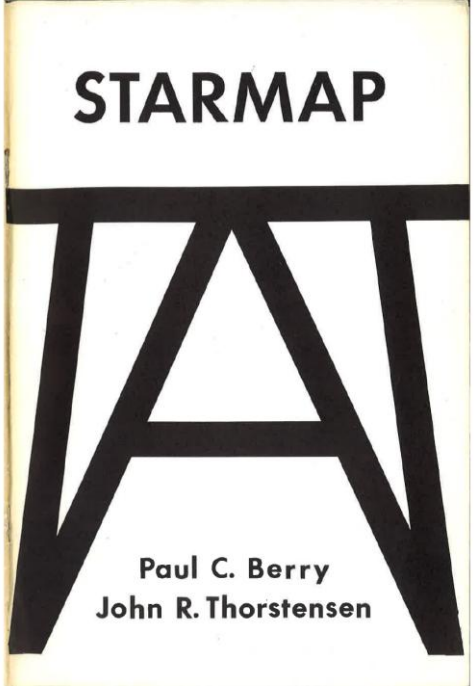
NuGet Porter2Stemmer

```
public static class Batch
{
    private static readonly EnglishPorter2Stemmer Stemmer = new();

    // Nested format: APL nested vector of char vectors <-> string[]
    public static string[] StemAll(string[] words)
    {
        var stems = new string[words.Length];
        for (var i = 0; i < words.Length; i++)
        {
            stems[i] = Stemmer.Stem(words[i]).Value;
        }
        return stems;
    }
```


Before optimization
VIDEOS.Stem
15000 calls
VIDEOS.Query
300 calls
Jarvis.SafeJSON
302 calls
QUERY.Table2JSON
300 calls
VIDEOS.Rank
300 calls
Jarvis.Server
0 calls
Jarvis.ToJSON
302 calls
QUERY.Paginate
300 calls
Jarvis.DebugLevel
4534 calls
Jarvis.HandleRequest
304 calls
0
2k
4k
6k
8k
10k
Cumulative time (ms)


After optimization
VIDEOS.Stem
300 calls
VIDEOS.Query
300 calls
Jarvis.SafeJSON
302 calls
QUERY.Table2JSON
300 calls
VIDEOS.Rank
300 calls
Jarvis.Server
0 calls
Jarvis.ToJSON
302 calls
QUERY.Paginate
300 calls
Jarvis.DebugLevel
4534 calls
Jarvis.HandleRequest
304 calls
0
2k
4k
6k
8k
10k
Cumulative time (ms)



- Descriptions of the calculations
- The relevant APL
- The result of plotting

Calculating the Positions of the Planetary Bodies

The function *CALCULATEPLANETS* finds *PLANETS*, a table of the positions of the sun, moon, and planets at the desired date. When first calculated by the function *PLANETPOS*, these positions are stated in 3-dimensional heliocentric Cartesian coordinates. But the function *EARTHVIEW* converts them to geocentric polar coordinates (right ascension in hours, declination in degrees, and distance in astronomical units), locating the planets with respect to the center of the Earth.

In order to plot the sky above a particular place, the function *SKYPOS* (see p. 29) is used to calculate *AA*, a table of altiude and azimuth with respect to given time and location on the Earth's surface. The function *VISIBLE* is used to select from *P* those members that are above the horizon, and saves them in the table *AAP*. Finally, the *PROJECTION* of these coordinates onto a flat surface is calculated, and translated to the Cartesian form expected by the plotting function; the function *IF* is simply a compression of the left argument by the right, along the first axis, defined by the APL symbol ⌿.

```
∇ CALCULATEPLANETS; AA; MOON; SUN; KOHOUTEK
  PLANETCOORD←AAP←PLANETS←VP←⍳0
  PLANETS←DATE EARTHVIEW DATE PLANETPOS (3≠⍳9)⌿PLANETS
  SUN←DATE EARTHVIEW 0 0 0
  K←100≥|DATE-JNU 12 28 1973
  KOHOUTEK←DATE EARTHVIEW (DATE IF K) COMETPOS KOHOUTEK
  MOON←MOONPOS DATE
  PHASE←MOON[1;1] MOONPHASE SUN[1;1]
  PLANETS←MOON,[1] SUN,[1] PLANETS,[1] KOHOUTEK
  MOON←MOON[;3] PARALLAXADJUST (LAT,DATE,TIME) SKYPOS MOON
  AA←MOON,[1] (LAT,DATE,TIME) SKYPOS 1 0⌿PLANETS
  PLANETCOORD←MAPCARTESIAN PROJECTION AAP←AA IF VP←VISIBLE AA
```

Execution of *CALCULATEPLANETS* causes new values to be assigned to four global variables. (These are initially set to ⍳0 in the first statement, mainly to draw attention to a list of the gobal variables which will be reset as a consequence of executing this function.) The four are:

PLANETS — The right ascension and declination of the moon, sun, planets, and Kohoutek.

VP — A logical vector indicating which planets are visible from the place, date, and time requested.

AAP — The altitude and azimuth of the visible planets.

PLANETCOORD — The Cartesian coordinates used to plot the projection of the visible planets.

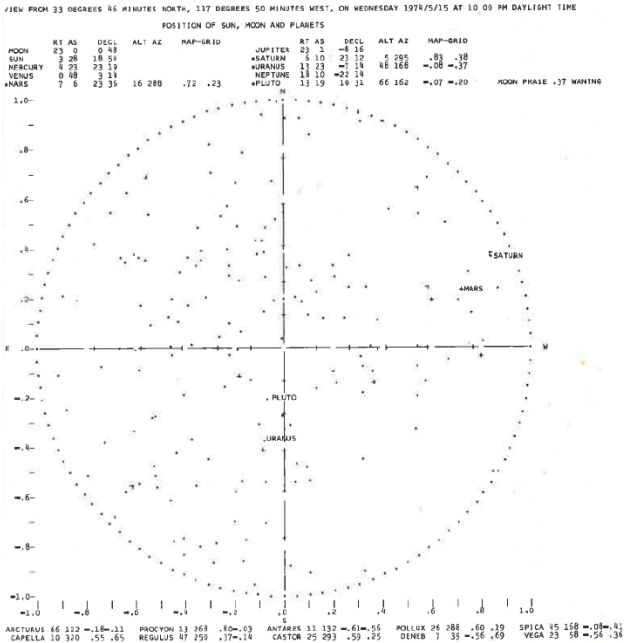


Fig. 2 Sample chart produced by the STARMAP workspace

What is STARMAP?

- It’s a wonderful little book
- Original from 1973
  - IBM technical demo collaboration w/ an astronomer
- This project based on version from 1978
- It uses a plotter!
  - We need a new UI

What is STARMAP, the repository?

- Transcribed by Stephen Taylor using his physical copy
  - He used Claude
  - There may have been some transcription issues
    - Neil used Claude to fix those

Corrections

- Fixed any errors that seemed to be transcription-based

Corrections

- Fixed any errors that seemed to be transcription-based
- Used Claude to write tests against *current* star charts
  - Seems we were slightly disagreeing with 1973
  - Generated tests based on current JPL datasets
    - Accepted the JPL data as correct, even when it disagreed with the source

Working model, so now what?

A pre-recorded demo of a new UI that sits on top of STARMAP follows:

- Input via a ‘friendly’ browser form
- Now animated over a period of time
- Visual confirmation that it seems to work
  - Celestial bodies seem to move at likely relative speed – note the moon
- Available at https://starmap.kirsopp.me

STARMAP animation generator

Start date

15/05/1974

End date

12/06/1975

UTC time (hours, decimal)

5

Frames per second

15

e.g. 5 = 05:00 UTC = 10 PM PDT

Latitude

52.526

Longitude (east +, west −)

+0.388

Large display Reduce size (every other day)

Generate SVG Generate .mp4 (slow)

Small front-end to BIG front-end

Claude is the New □

- Stephen Taylor