DYALOG

Glasgow 2016

Technical Road Map: Under The Covers

Jay Foad

Agenda

➢ Version 15.0 features

➢ Version 16.0 features and plans

➢ Other developments
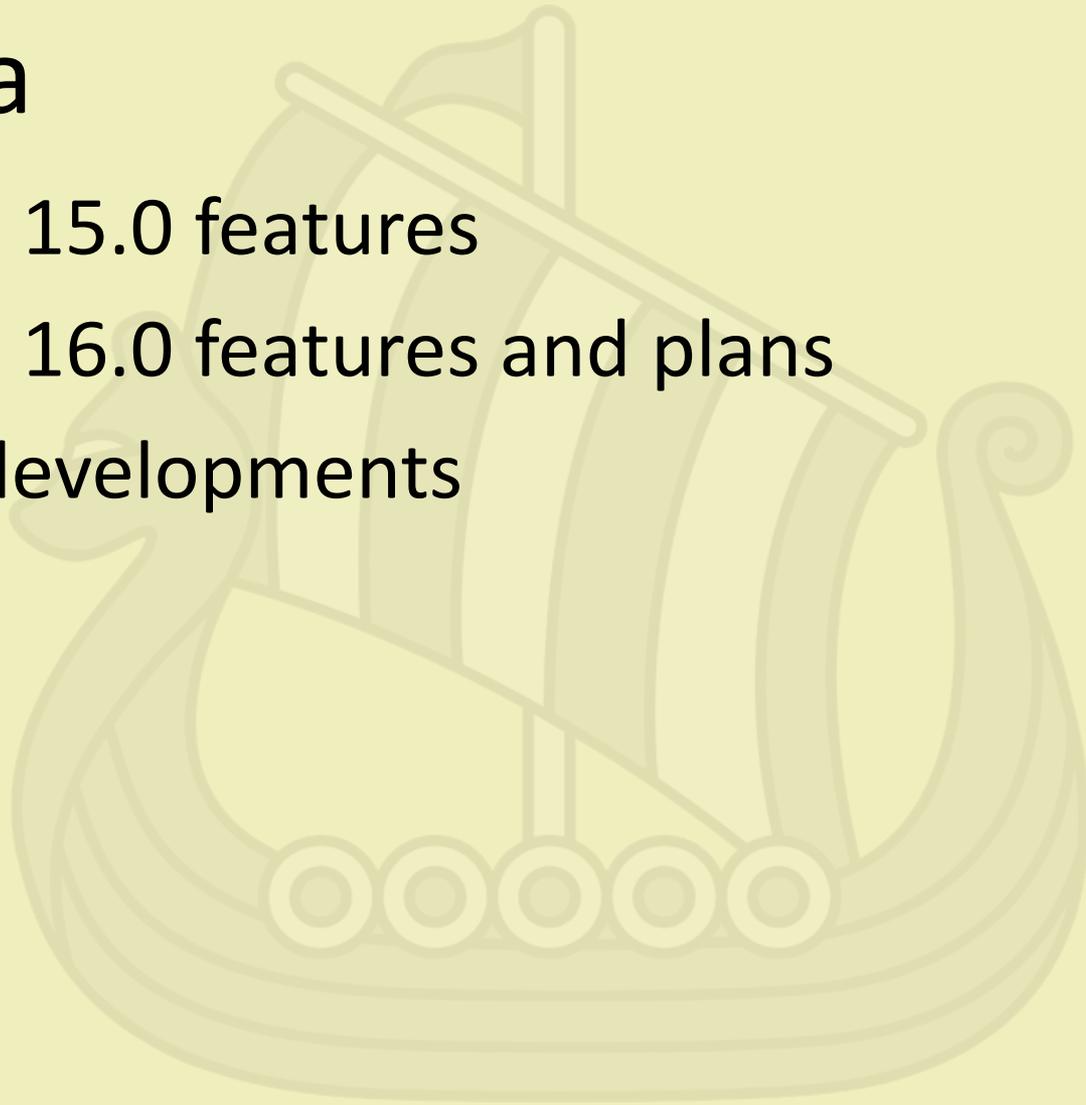
Version 15.0

➢ Performance

➢ Native file functions

➢ APL source in Unicode text files

➢ RIDE 3.0

➢ Hashed arrays

➢ Case conversion

➢ Random numbers

➢ Catenate reduction of empty arrays

Technical Road Map: Under The Covers

Version 15.0 Performance

Performance Comparison
Between Windows-64 15.0.27176.0 W Development and Windows-64 14.1.25052.0 W Development

Geometric mean of 136 expression groups: -25.5%

Relative Performance of 136 Expression Groups

97.8% - with 126/136 more than 2% better   2.2% - with 0/136 more than 2% worse
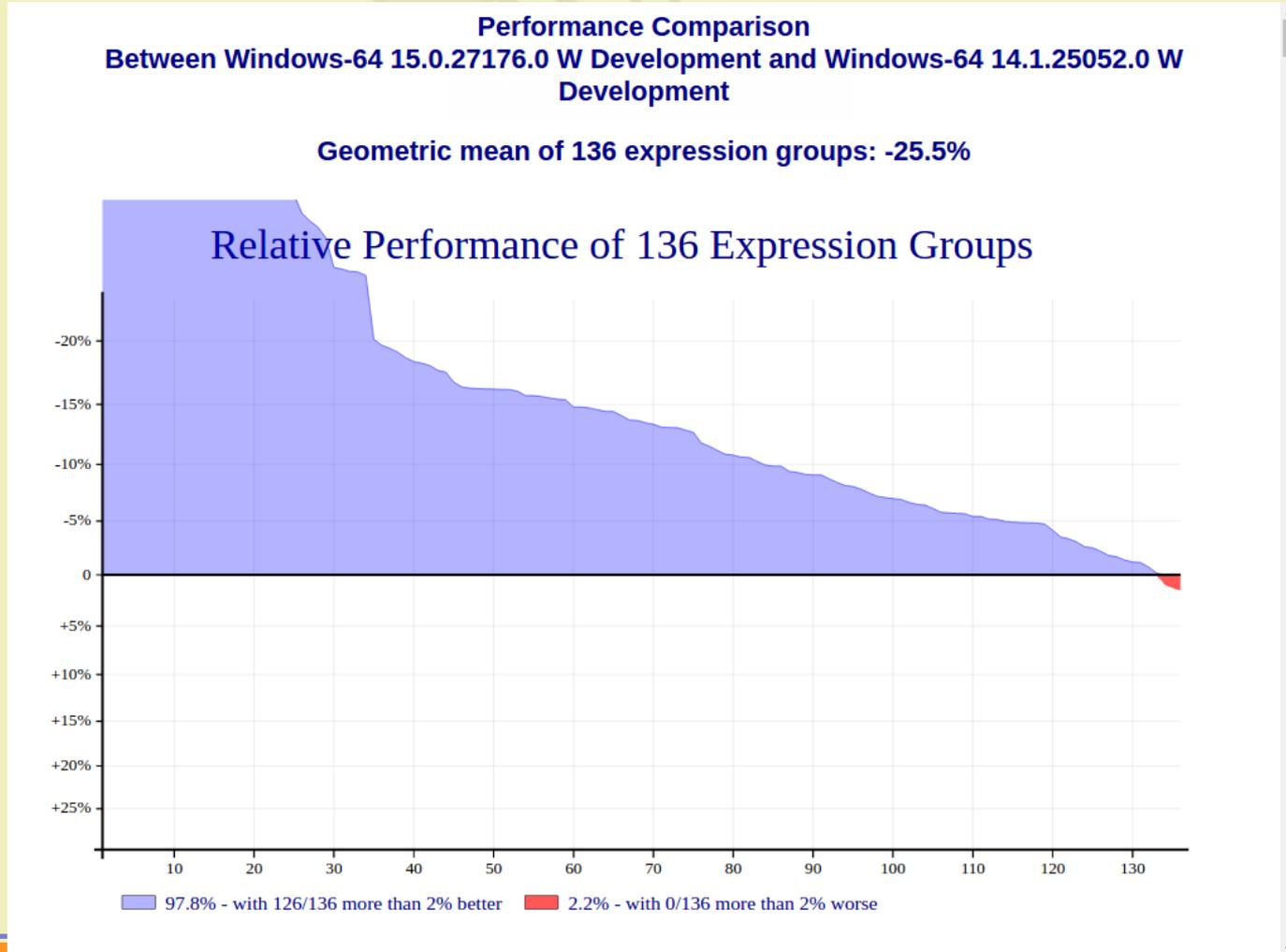
Version 15.0 Performance

➤ Dyadic ⍳ family
(especially on small integers and characters)
➤ Partition pseudo-operator b (≠⍤⊂) x
➤ More special cases for Rank ⍤ and Key ⌸
➤ Dyadic ⎕FMT and dyadic ⍕
➤ Inner product
➤ Boolean reductions
➤ Windowed ⌈/ and ⌊/
➤ Grade up ⍋ and down ⍒ on permutations
➤ And on and on...

Version 15.0 Performance

C compilers upgraded

➢Windows: from Visual Studio 2005 to 2015

➢AIX: from XL C/C++ 12.1 to 13.1

➢Linux: from GCC 4.3.3 to 5.3.0

Version 15.0 Native file functions

⎕NINFO lists the contents of a folder

⎕MKDIR creates folders

⎕NEXISTS and ⎕NDELETE check for existence of and remove files or folders without having to open them

⎕NGET and ⎕NPUT read and write Unicode text files

Version 15.0 APL source in text files

```
)ED c:/bar.dyalog
⎕FIX 'file://c:/foo.dyalog'
```

Interpreter will remember and update source file

Editor also highlights JSON and XML

`:Require` is built in
(replaces SALT's `∆:require`)

Version 15.0 RIDE 3.0

➢ Default UI on macOS

➢ Workspace explorer

➢ Source code navigation

➢ Auto-PW

➢ Value tips

Version 15.0 Hashed arrays

- ➤ `1500ɪ` hashes an array

- ➤ Supersedes the old `A∘ɩ` mechanism

- ➤ Improves the performance of all set functions

- ➤ Hash table is updated on:

  - • Append `,←` and `⍪←`
  - • Chop `↓⍨←`

Version 15.0 Random numbers

```
⎕RL ← 0 1
```

➤ Second item selects the RNG

➤ `16807⌶` is now deprecated

Version 15.0 Catenate reduction

```
⊃,/ 2↑'AB' 'CD'   ↔   'ABCD'
⊃,/ 1↑'AB' 'CD'   ↔   'AB'
⊃,/ 0↑'AB' 'CD'   ↔   ''
```

Version 15.0 Bytecode compiler

➢ Gives a factor of 1.5 to 2 speed-up on code using small arrays

➢ Version 15.0 work:

- Better support for global variables
- Show compiler errors in the editor
- Some support for nested dfns
- Plus many parser improvements

Version 15.0 Release

➢ Released 30 June 2016

➢ Free for non-commercial use!

Version 16.0

➢ Performance

➢ New primitives

➢ JSON and CSV support

➢ Embedded web browser

➢ Run time warnings

➢ Options on system commands

➢ RIDE 4.0

➢ Issuing updates

Version 16.0 Performance

➢ We're working on it!

➢ For more details:
(D08) Performance: The Neverending Story
Roger Hui and Jay Foad
Tomorrow at 10:00

Version 16.0 New primitives

➤ Where

➤ Enclose if simple

➤ At (aka Merge)

And maybe:

➤ Tile (aka Tessellate)

➤ Interval index

Version 16.0 New primitives

Where are the 1s in a Boolean vector?

$$⍸ \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad\quad ↔ \quad\quad 1 \quad 3 \quad 6$$

Enclose-if-simple:

$$\{≡⊆ω\}\,'one'\ 'two' \quad ↔ \quad 2$$
$$\{≡⊆ω\}\,'one' \quad\quad\quad ↔ \quad 2$$

At is a triadic merge operation:
this array but with *these* items at *those* places

Version 16.0 New primitives (maybe)

Tile is one case of the Sharp APL Cut operator

Interval index looks up items in a sorted sequence of intervals

(D06) New Primitive Functions and Operators

John Scholes and Roger Hui

14:45 this afternoon

Version 16.0 JSON and CSV

JSON support will graduate from an I-beam
to a system function ⎕JSON

We're also considering ⎕CSV

If you're interested, please talk to us!

Version 16.0 Language features

- ➢ Dynamic scope for dfn error guards

- ➢ Options on system commands
  ```
  )SAVE -force foo.dws
  ```

- ➢ Run time warnings for dubious constructs, common migration problems, etc

Version 16.0 Issuing updates

➤ We'd like to move away from the patch workspace

➤ Obvious replacements are:

- MSI update packages on Windows
- Authenticated package repositories on Linux/Pi

Compilation: Co-Dfns

Experimental compiler by Aaron Hsu (Indiana University)

➢ Support for Boolean vectors in the compiler
➢ Basically complete scalar runtime
➢ Most mixed functions, and operators
➢ Full GPU execution by default in most cases
➢ Support for caching data on GPU
➢ Improved reliability and stability fixes across the board
➢ Better user documentation and interface

(U06) Co-dfns Report: GPU Performance, Workflow and Usability
Aaron Hsu, 09:00 Tomorrow

Compilation: Harnessing the GPU

➢ GPU algorithms: see Aaron Hsu's work

➢ CPU/GPU communication:
fully automatic offloading is still an unsolved problem

➢ We need to build expertise

➢ We need to find a sweet spot for APL

Interfacing with other languages

➢ R

➢ Python

➢ Julia

➢ MATLAB
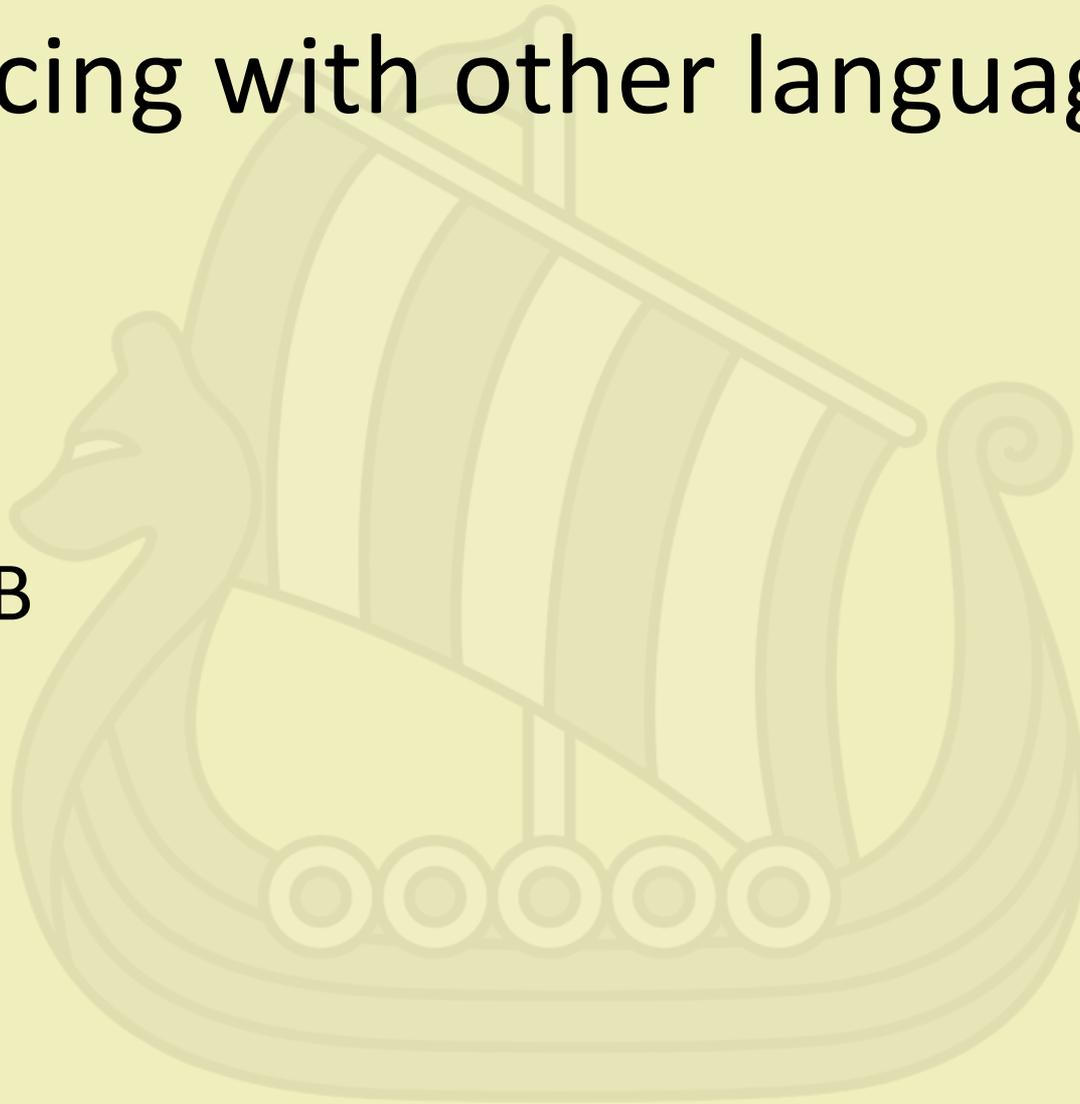
Scripting: for applications

➢ Support for editing scripts is now built into the interpreter

➢ Already starting to replace the lower layers of SALT

➢ Will be a key part of APL packages, or the "APL Project Project"

Scripting: for utilities

➤ Use APL as a scripting language in a UNIX shell…

➤ Or on Windows…

➤ Or both!

```
$ cat fib
#!/usr/bin/apl
fib←{
    ω∊0 1:ω
    +/∇¨ω-1 2
}
□←fib¨□
$ echo 5 6 7 |
./fib
5 8 13
$
```

ubuntu
on
Windows 10

CTO Goals

➢ Harness the GPUs for APL number crunching

➢ Learn from, and interface with, Julia, Python etc.

➢ Keep pushing core performance
(aka business as usual)

➢ Expand the core development team