

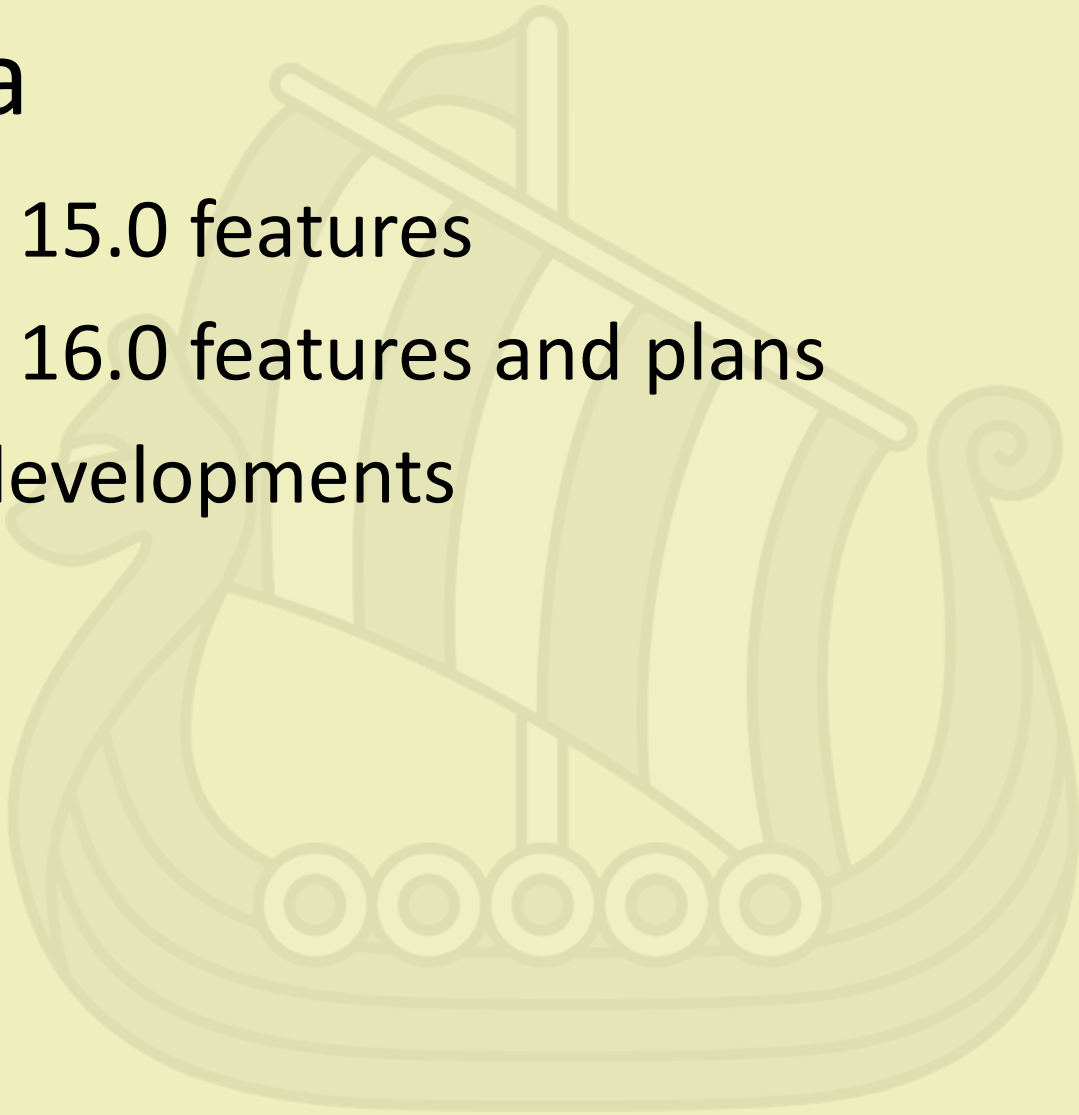
Technical Road Map: Under The Covers

Jay Foad



Agenda

- Version 15.0 features
- Version 16.0 features and plans
- Other developments

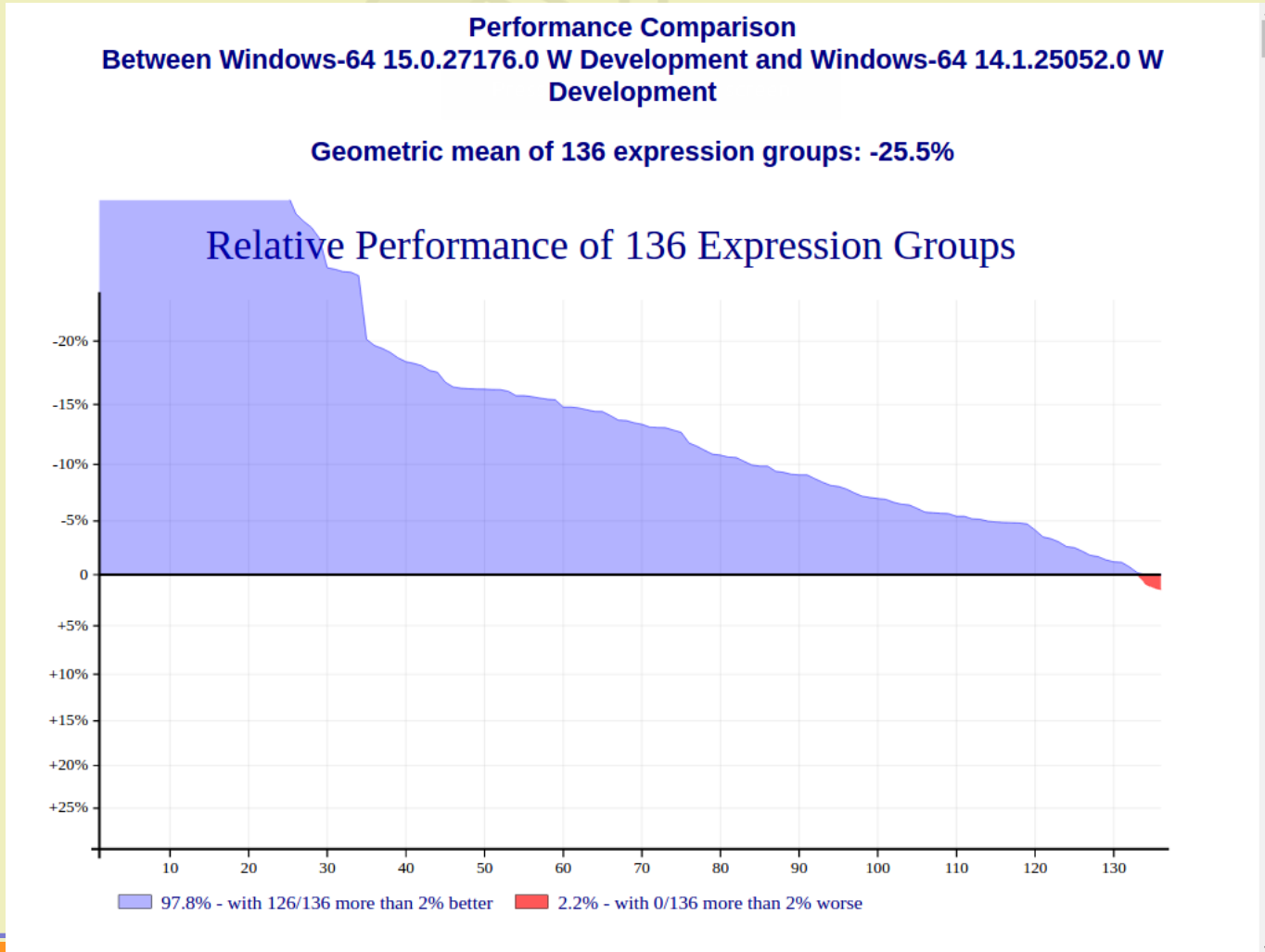


Version 15.0

- Performance
- Native file functions
- APL source in Unicode text files
- RIDE 3.0
- Hashed arrays
- Case conversion
- Random numbers
- Catenate reduction of empty arrays



Version 15.0 Performance



Version 15.0 Performance

- Dyadic τ family
(especially on small integers and characters)
- Partition pseudo-operator $b (\neq \cdot c) x$
- More special cases for Rank \circ and Key \boxminus
- Dyadic \square FMT and dyadic \wp
- Inner product
- Boolean reductions
- Windowed $\lceil /$ and $\lfloor /$
- Grade up \blacktriangle and down \blacktriangledown 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

- 1500 \pm hashes an array
- Supersedes the old A \circ z mechanism
- Improves the performance of all set functions
- Hash table is updated on:
 - Append , \leftarrow and ; \leftarrow
 - Chop ↓ $\ddot{\leftarrow}$



Version 15.0 Random numbers

$\square RL \leftarrow 0 \ 1$

- Second item selects the RNG
- 16807 is now deprecated



Version 15.0 Catenate reduction

$\Rightarrow, / 2 \uparrow 'AB' 'CD' \leftrightarrow 'ABCD'$

$\Rightarrow, / 1 \uparrow 'AB' 'CD' \leftrightarrow 'AB'$

$\Rightarrow, / 0 \uparrow 'AB' 'CD' \leftrightarrow ''$



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?

1 1 0 1 0 0 1 ↔ 1 3 6

Enclose-if-simple:

{≡⊆ω} 'one' 'two' ↔ 2

{≡⊆ω} 'one' ↔ 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



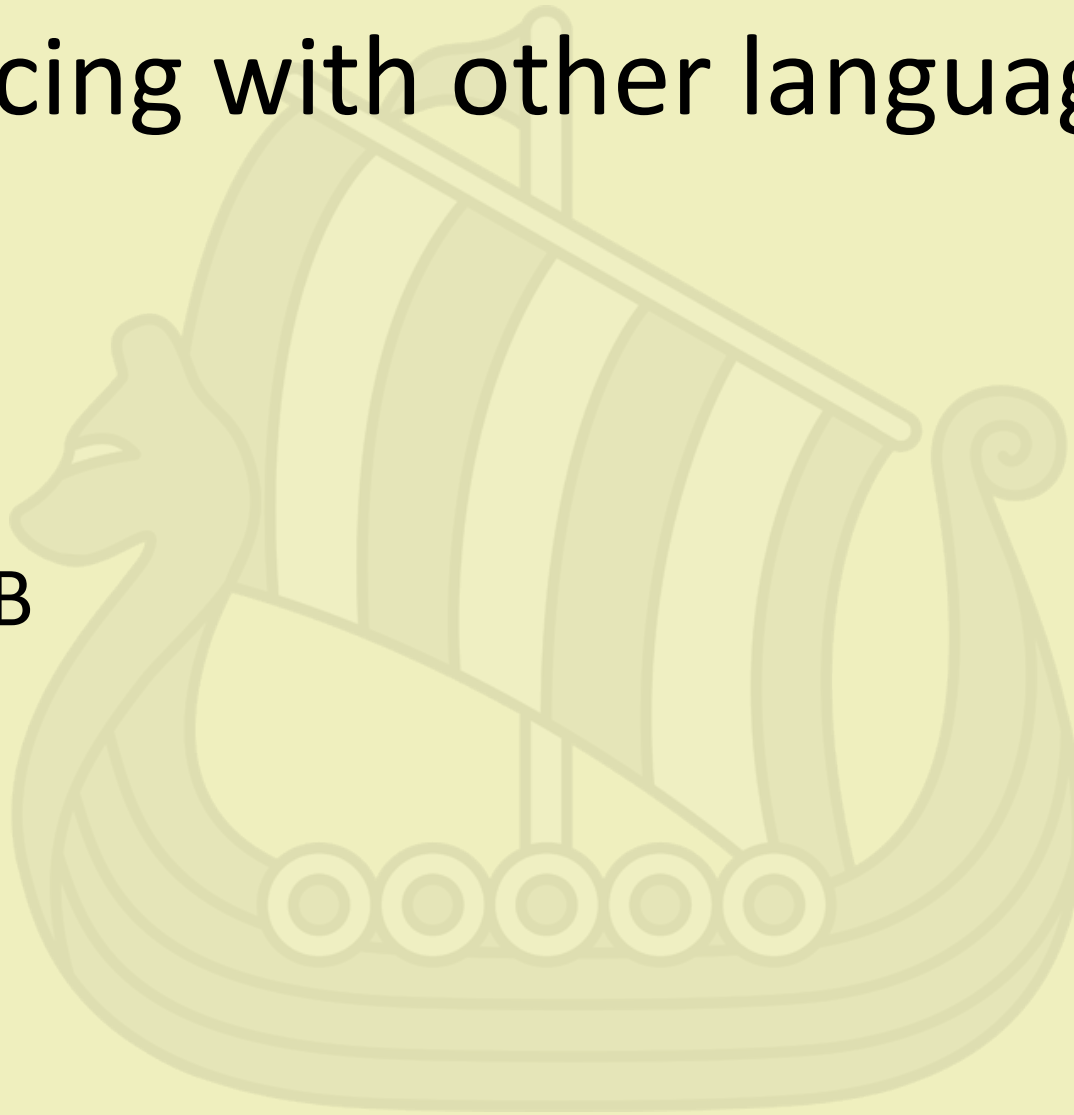
Compilation: Harnessing the GPU

- 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
    +/∇ωω-1 2
}
⍳←fib⍵
$ echo 5 6 7 |
./fib
5 8 13
$
```



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

