24U SOFTWARE

# Initiative '24

HOnza Koudelka

# About Me

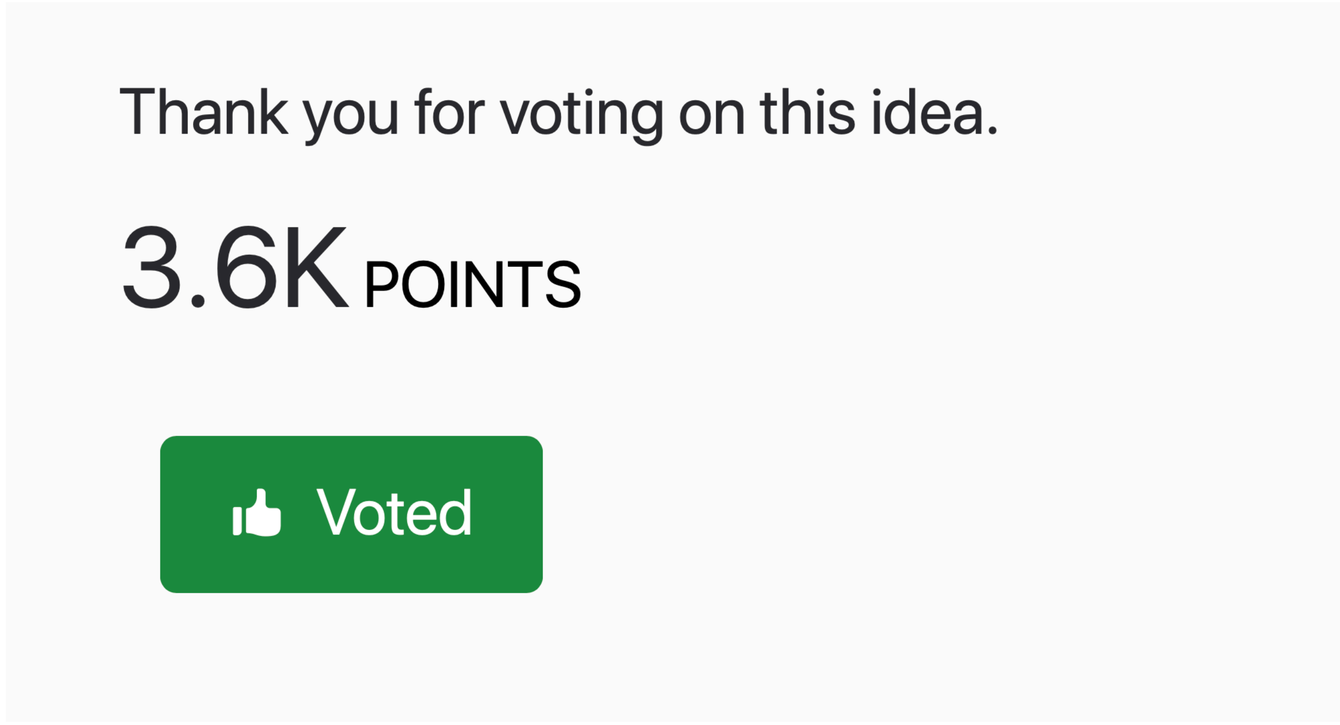
- Co-founder and co-owner of 24U
- FileMaker developer since 1991
- Mad Optimizer
- Achiever of the Impossible…
- Speaking at FileMaker events since 2011
- For more about me see **honza.guru**

Claris FileMaker
CERTIFIED DEVELOPER

24U SOFTWARE

# Significantly faster calculation engine

5 years ago by **HOnza**



# 5 years of performance improvements

- 10 FileMaker version updates will now follow.
- For each update, try to recall:
- Did you notice a significant (and sufficient) improvement caused by the presented version update?
- Did you have to make changes to your solutions to take advantage of the new improvement?

24U SOFTWARE

# FileMaker 19.2.1

- **Find queries** can now be **cancelled**
- HTTP/2 provides faster web performance

# FileMaker 19.3.1

- Mac version runs **natively on Apple silicon**
- Windows version uses the Microsoft Edge (Chromium) engine instead of the Internet Explorer engine in Web Viwer

# FileMaker 19.4.1

- Performance **improvements for create, read, update, and delete operations**
- WebDirect increases number of stable concurrent web connections
- WebDirect generating CSS only once when switching layouts or resizing non-card windows
- For faster SQL queries about fields defined in a FileMaker Pro file, you can now use the system table named FileMaker_BaseTableFields rather than the existing FileMaker_Fields table.

# FileMaker 19.5.1

- Summary fields processed on server (under certain conditions)
- Scripting engine memory cache increased to 256 MB
- Layout CSS cached in FileMaker Server
- WebDirect performance with multiple concurrent connections improved

# FileMaker 19.6.1

- WebDirect CSS caching improved
- **Transactions**

# FileMaker 19.6.2

- WebDirect overall load time optimized

# FileMaker 20.1.1

- Perform Script on Server with Callback

# FileMaker 20.3.1

- Loop script step now includes the **Flush** option
- Database engine now caches relationships
- Optimized memory allocation for relationship changes
- Account enabled state cached to optimize login performance

# FileMaker 21.0.1

- Execute FileMaker Data API now supports write operations
- Windows: Search biox performance improved in Manage Layouts, Layout mode, and Script Workspace
- WebDirect CSS caching optimized during layout resize
- Database field definitions now cached by FileMaker Server

# FileMaker 21.1.1

- Constrain Found Set can be set to ignore indexes
- Externally stored container fields using secure storage can now opt to store files in fewer folders
- Server-side scripts can now use Perform Script on Server

# Have you benefited

from at least one of the improvements?

24U SOFTWARE

# Do you use calculations?

Will you benefit from all your calculations becoming faster?

Without having to change a single bit?

24U SOFTWARE

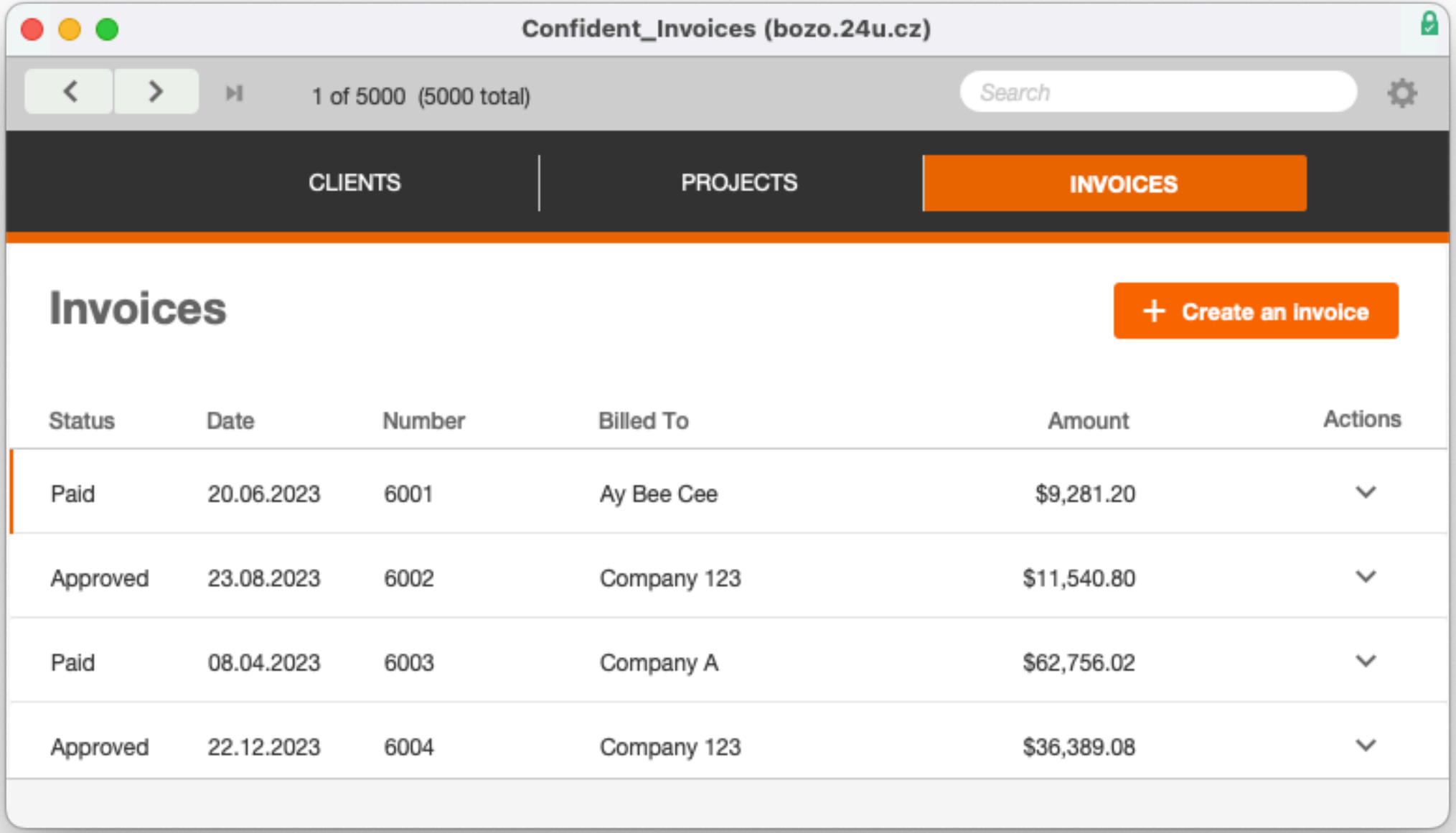
# FileMaker Calculations Performance

## When a Millisecond Matters


24U SOFTWARE

# How quickly you can "kill" a solution with a single "innocent" calculatin?

- Took the "Confident Invoices" starter file from Alexis Allen
- Added one calculation and used it in various places
  - Field on a layout
  - Conditional formatting
  - Record-level privileges
  - Sorting



# "Innocent" Unstored Calculation

```
Let ( [
t1 = Get ( CurrentTimeUTCMicroseconds )
s =

Case (
PaidAmount ≥ TotalAmount and ( PaidDate - DateDue ) < -2 ; "early" ;
PaidAmount ≥ TotalAmount and ( PaidDate - DateDue ) ≤ 2 ; "on time" ;
PaidAmount ≥ TotalAmount and ( PaidDate - DateDue ) > 2 ; "late" ;
DateDue < Get ( CurrentDate ) ; "overdue" ;
DateDue < ( Get ( CurrentDate ) + 1 ) ; "due" ;
"pending"
)

;
t2 = Get ( CurrentTimeUTCMicroseconds ) ;
$$count = $$count + 1 ;
$$time = $$time + t2 - t1
] ; s )
```

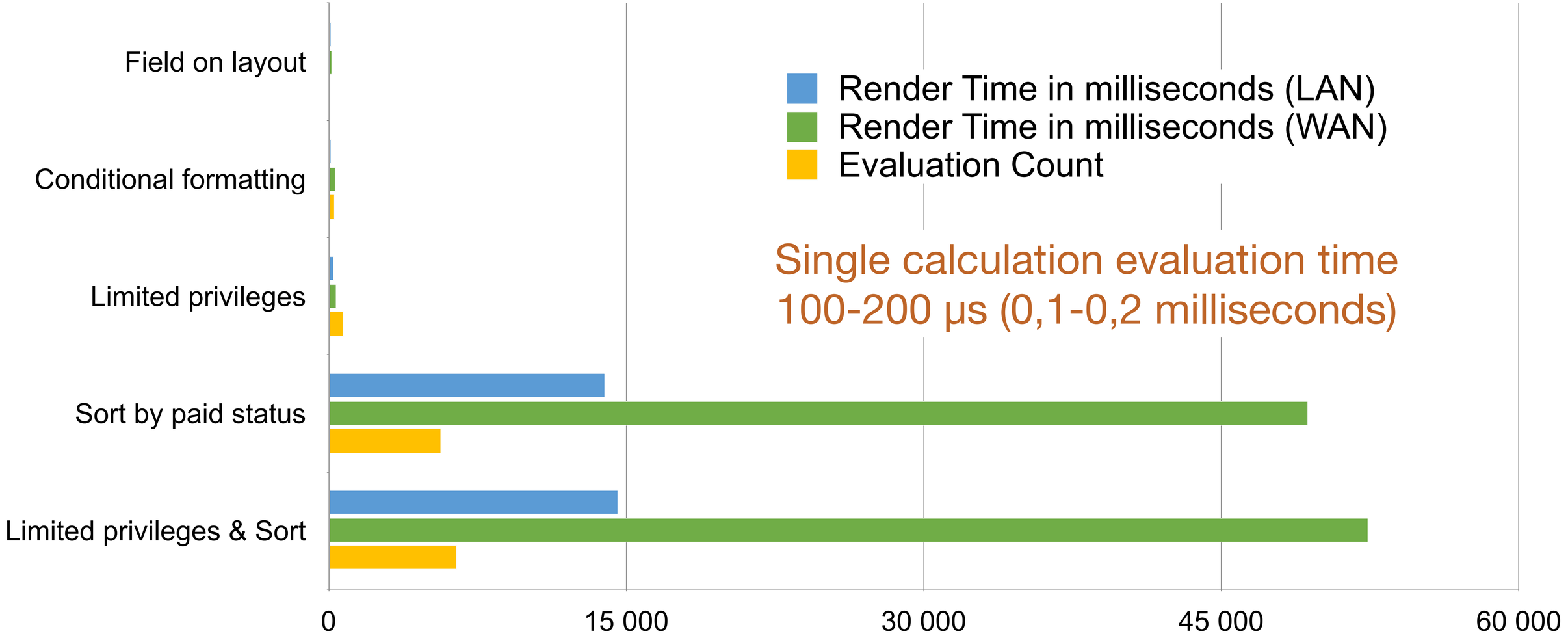
My calculation

Evaluation count

Evaluation time

24U SOFTWARE

# Performance Impact (list view with 10 visible records)

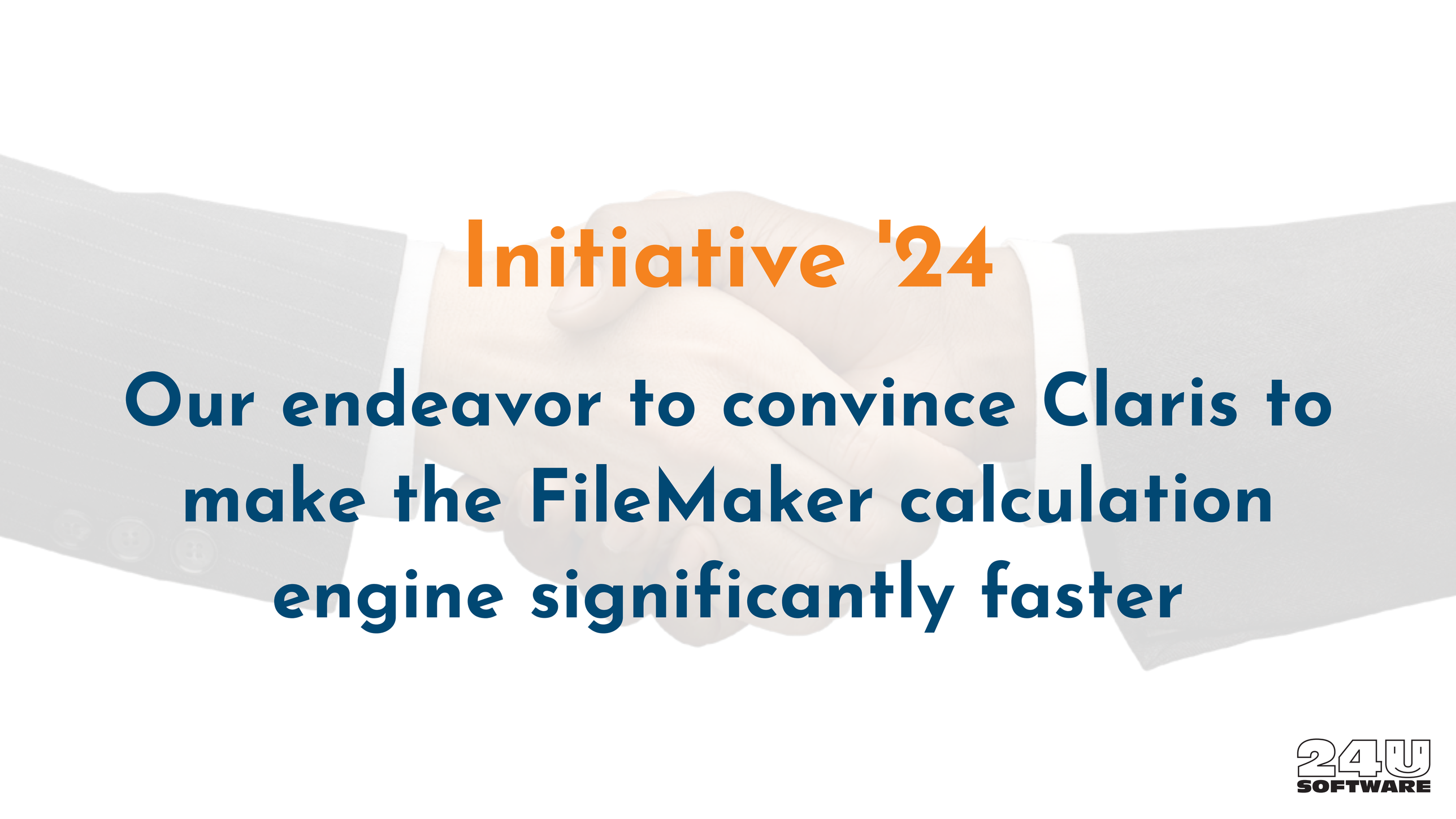



24U SOFTWARE

# Learn more in my article

https://24usw.com/millisecond

24U SOFTWARE



# Initiative '24

## Our endeavor to convince Claris to make the FileMaker calculation engine significantly faster

24U SOFTWARE

# Why? Calculations are everywhere!

- Calculated fields
- Field values auto-enter
- Field validation
- Container field external paths
- Script steps
- Conditional formatting
- Record-level privileges
- Custom menus
- File > Send > Mail menu command
- Records > Replace Field Contents menu command
- Script trigger parameters
- Hide object when
- Button bars
- Tab control tab names and widths
- Pop-over titles
- Tooltips
- Field placeholders
- Portal filters
- Web Viewers
- Charts
- Plug-ins
- Data Viewer
- Text objects **ADDED IN 20.2**
- Validation messages **ADDED IN 21.1**

24U SOFTWARE

# Reference Benchmark (Math)

```
// Math test
$mspeed = SetRecursion (
While ( [ i = 30 ; r = 0 ] ; i ; [ mspeed =
      While ( [ j = 0 ; $mresult = 0 ; t = Get ( CurrentTimeUTCMicroseconds ) + 1000000
              ] ; Get ( CurrentTimeUTCMicroseconds ) < t ; [

              a = $mresult + 0,123456 ;
              b = $mresult - 0,987654 ;
              c = $mresult * 0,456789 ;
              d = $mresult / 0,654321 ;
              e = Average ( a ; b ) ;
              f = Average ( c ; d ) ;
              $mresult = Sum ( Min ( e ; f ) ; Max ( e ; f ) ) / 2 ;

              j = j + 1 ] ; j )
      ; $log = List ( $log ; i & ": Math result: " & $mresult ; i & ": Math speed: " & mspeed &
" cycles/second" )
      ; r = Max ( mspeed ; r ) ; i = i - 1 ] ; r )
; 100000000 ) ;
```

24U SOFTWARE

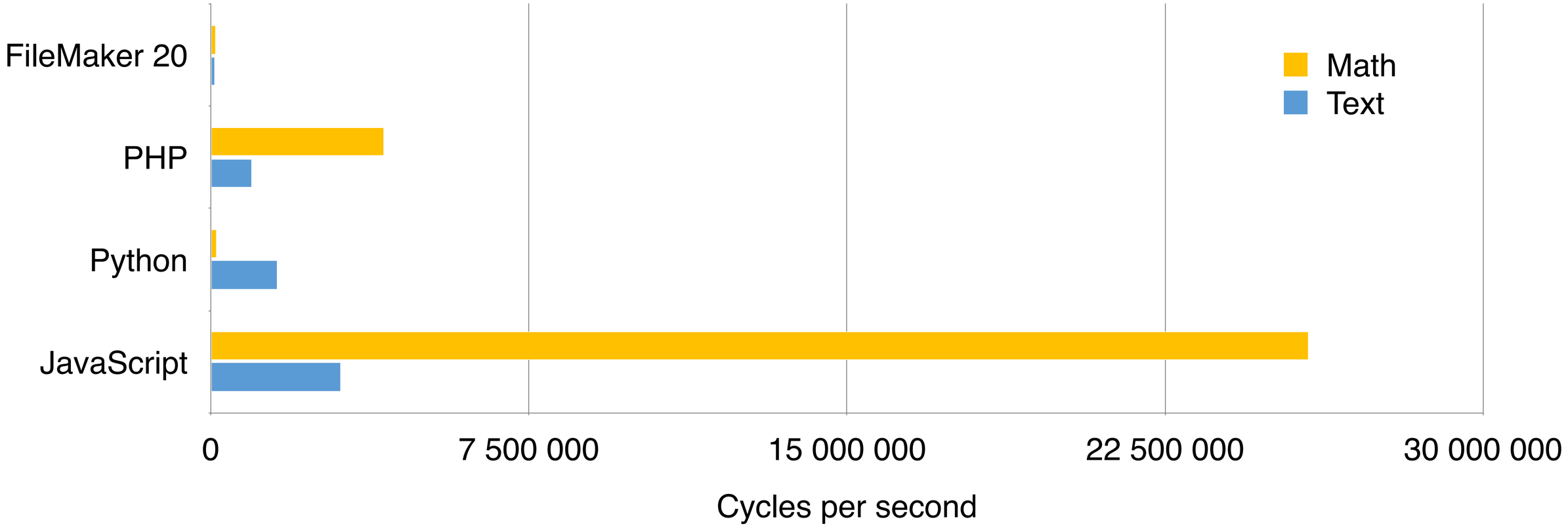
# Reference Benchmark (Text)

```
// Text test
$tspeed = SetRecursion (
While ( [ i = 30 ; r = 0 ] ; i ; [ tspeed =
      While ( [ j = 0 ; $tresult = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ěščřžýáíéúůďťň" ;
         t = Get ( CurrentTimeUTCMicroseconds ) + 1000000
              ] ; Get ( CurrentTimeUTCMicroseconds ) < t ; [

              a = Left ( $tresult ; 15 ) ;
              b = Right ( $tresult ; 15 ) ;
              c = Middle ( $tresult ; 16 ; 20 ) ;
              d = Lower ( a ) ;
              e = Upper ( b ) ;
              f = List ( e ; d ; c ) ;
              $tresult = Substitute ( f  ; ¶ ; "" ) ;

              j = j + 1 ] ; j )
      ; $log = List ( $log ; i & ": Text result: " & $tresult ; i & ": Text speed: " & tspeed &
" cycles/second" )
      ; r = Max ( tspeed ; r ) ; i = i - 1 ] ; r )
; 100000000 )
```

# Performance Comparison




24U SOFTWARE

# Workarounds

## What can we do now?

24U SOFTWARE

# Avoid unstored calculations

- Pros:
  - No expensive calculations over relationships
  - Every result calculated just once
  - Faster loading/displaying of data (sometimes)
- Cons:
  - Slower commits (more values to save, updating indexes)
  - More data to load/store per record

# Avoid stored calculations

- Pros:
  - Less data to load/store
  - Faster commits
  - Not calculated until needed
  - No load on server when all source data already cached
- Cons:
  - Calculations involving related records much slower on client
  - Difficult to avoid re-calculating the same value multiple times

# Off-load to server

- Pros:
  - All data on the same machine -> faster complex calculations
  - Client does not have to wait
- Cons:
  - More load on server (easy to overload)
  - More difficult to debug
  - More exceptions to handle

# Off-load to later (off-peak)

- Pros:
  - Better distribution of sever load
  - Processing more changes at once (recalculating less often)
- Cons:
  - Not suitable for 24/7 businesses
  - Less time slots available for off-peak maintenance
  - Users have to wait for the results, often hours

24U SOFTWARE

# Virtual List (cr. Bruce Robertson)

- Pros:
  - More control over data transfers
  - Lower load on server
  - Great for browsing
- Cons:
  - More development time
  - Often misunderstood/misimplemented
  - Not so great for editing

# Local File Editing (cr. Vince Menanno)

- Pros:
  - Less data transferred between client and server
  - Lower server load to save changes
  - Minimum time to keep record locked
- Cons:
  - Optimistic (save can fail)
  - Far more development time
  - Higher developer skills required

# Off-load to another technology

- Pros:
  - Hundreds or thousands times faster
- Cons:
  - More development time
  - More difficult server-side
  - First step to move away from FileMaker

# All have one in common

- Trying to trigger FileMaker calculation engine less often
- Trying to manage where it is triggered (client/server)
- Trying to manage when it is triggered (defer or pre-calculate)
- When a value is needed, we simply do have to calculate it somewhere, at some time, at least once

# What can Claris do?

## Make the calculation engine faster

# Objections

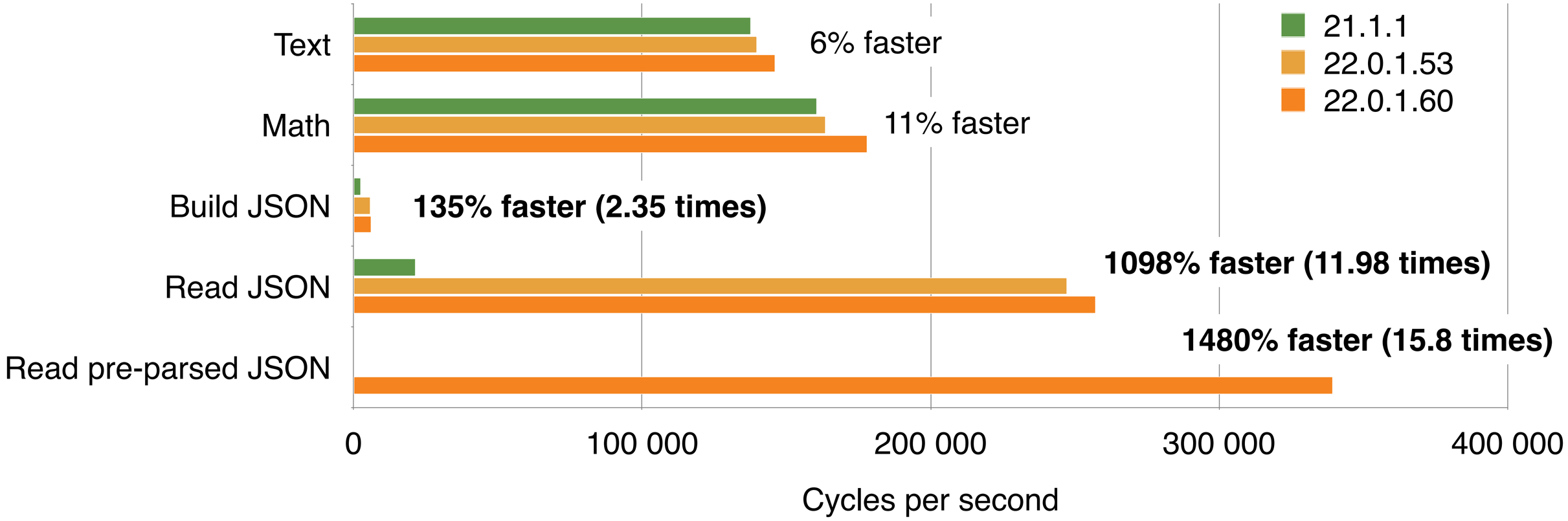
- We don't want to break existing solutions
- FileMaker's math is made for up to 400 digits precision
- FileMaker text functions are fully Unicode savvy
- It's a lot of work and we're not sure it's worth the investment

# My answers

- Nothing have to break
- 1% users who need this do accept the penalty.
  Why should the other 99% pay it as well?
- We're here to help
- New features attract new customers, optimization prevents existing ones from leaving the platform

24U SOFTWARE

# FileMaker 22 is the first step




24U SOFTWARE

# FileMaker Calculations Faster for Everyone

Vote here

**24usw.com/fastcalc**

3640 out of 6620

298 votes (2980 points) to reach target

Learn why

**24usw.com/i24**

24U SOFTWARE