



R6 Applications Development / Performance



D01

Agenda

- Performance overview
- Database options
- View options
- LotusScript methods
- LotusScript Tips

Performance Overview

- Every feature has a price
 - Time to develop
 - Trade-offs in other features
 - Time to maintain
 - Performance
- Our goal is to help you identify high value and low cost features, especially (today) in terms of performance

Performance Overview (cont.)

- We can
 - Compare performance with/out a feature enabled
 - Compare performance to R5
- We cannot
 - Give absolute numbers
 - "How many users can I have concurrently using my application?"
 - "How fast will my db/view/document open with feature xyz turned on/off?"

Database Options

- Set via Notes UI or Compact task
- Optimize document table map
 - -F will set this option ON
 - -f will set this option OFF
 - Domino Directory indexes about twice as fast!
 - Other applications return mixed results
 - Application 1: 2k docs, 60 views, 60mbs
 - Application 2: 20k docs, 140 views, 650mbs
 - In R5, this setting had no effect

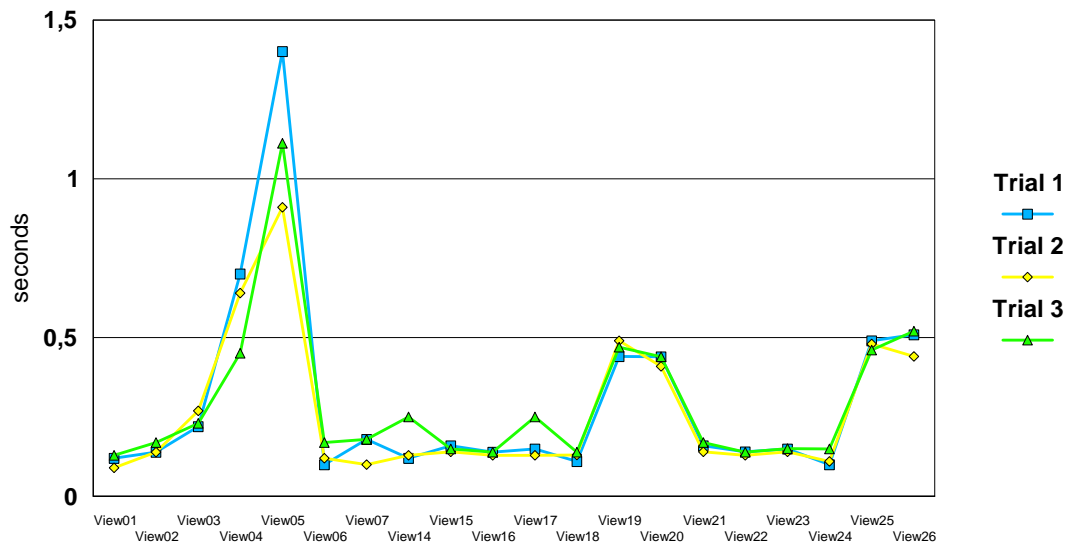
Database Options

- Unread Marks, Overwrite free space, Maintain LastAccessed, Maintain specialized response hierarchy
 - 5-10% faster indexing on both Application 1 and Application 2 with all settings optimized

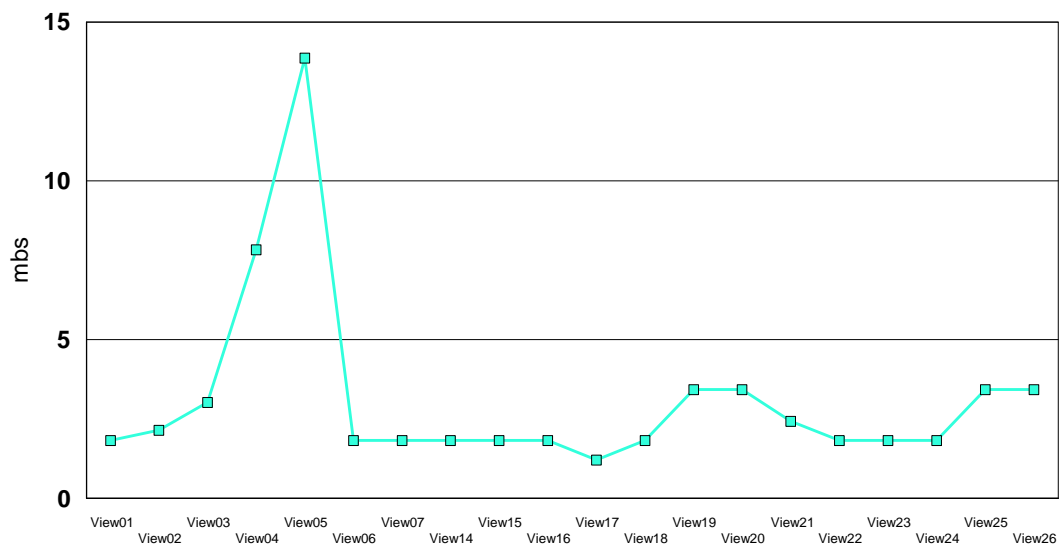
View Options

- View01: 5 columns, first one sorted
- View02: all columns sorted
- View03: click-sort on second column
- View04: click-sort on all columns
- View05: double click-sort on all columns
- View06: formulas instead of field names in all 5 columns
- View07: Selection formula instead of SIMPLE (same net result of all 10k documents)
- View08: @Today in Selection formula
- View09: @Today repeated in Selection formula
- View10: @Now in Selection formula
- View11: @Now repeated in Selection formula
- View12: @Today in column formula
- View13: @Now in column formula
- View14: alternating row colors, background image tiled, bolder left margin, blue grid, green simple header bars, allow 3 rows (shrink to fit), 1 1/4 row spacing, different bold/color/fonts for every column,
- View15: "create new documents at view level"
- View16: "Treat view contents as HTML"
- View17: "Generate unique keys in index"
- View18: "Evaluate actions for every document change"
- View19: categorize first two columns (expanded)
- View20: "don't show empty categories" with two columns categorized (expanded)
- View21: use value as color for a 6th column (at far left)
- View22: hide 4 out of 5 columns
- View23: 5th column is an "editable column"
- View24: "hide column if formula is true"
- View25: categorize first two columns (collapsed)
- View26: "don't show empty categories" with two columns categorized (collapsed)

View Indexing Times



View Index Sizes



Time/Date Sensitive Views

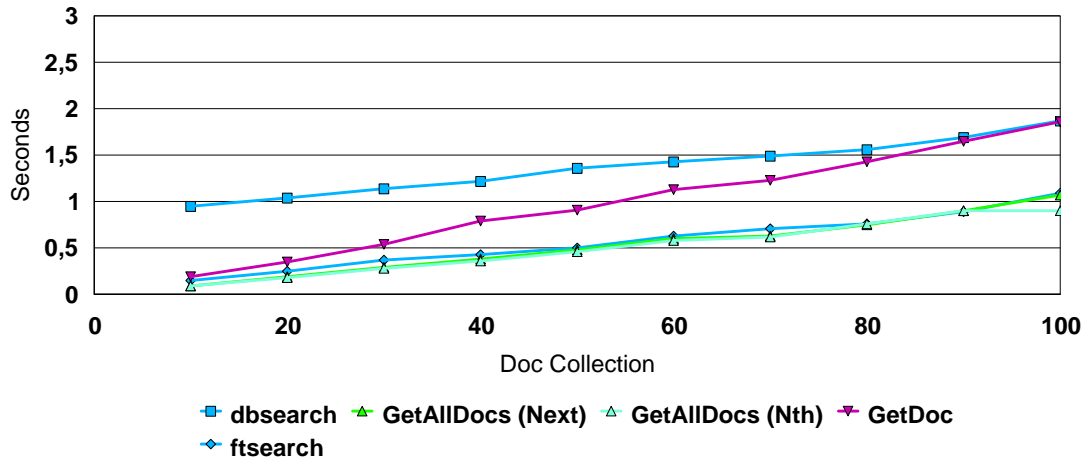
- @Now ~7 seconds
- @Today ~8 seconds
 - These numbers held true for selection formulas, column formulas, and repeated use of @Formula vs. setting a temp variable
 - Approximately 100 times as long as base view
- UPDATE task no longer tries to refresh these views!

LotusScript

- Agents getting a collection of many documents
- Tips
 - Iterating through a collection
 - Updating documents

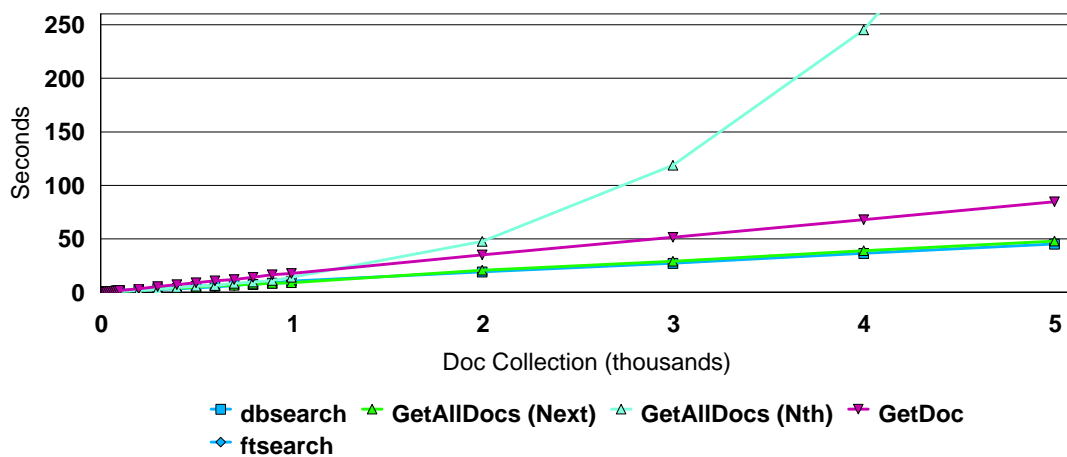
Agents

Getting collections in a 10k database



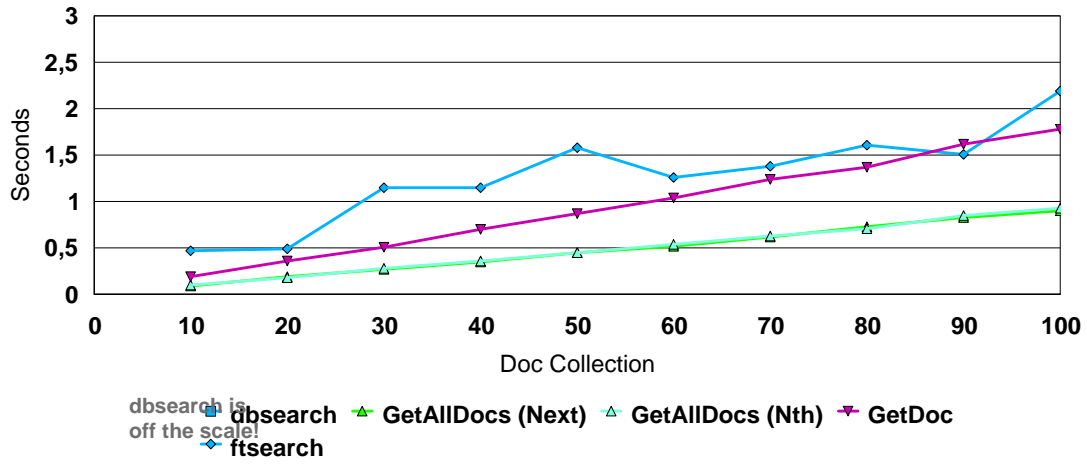
Agents

Getting collections in a 10k database



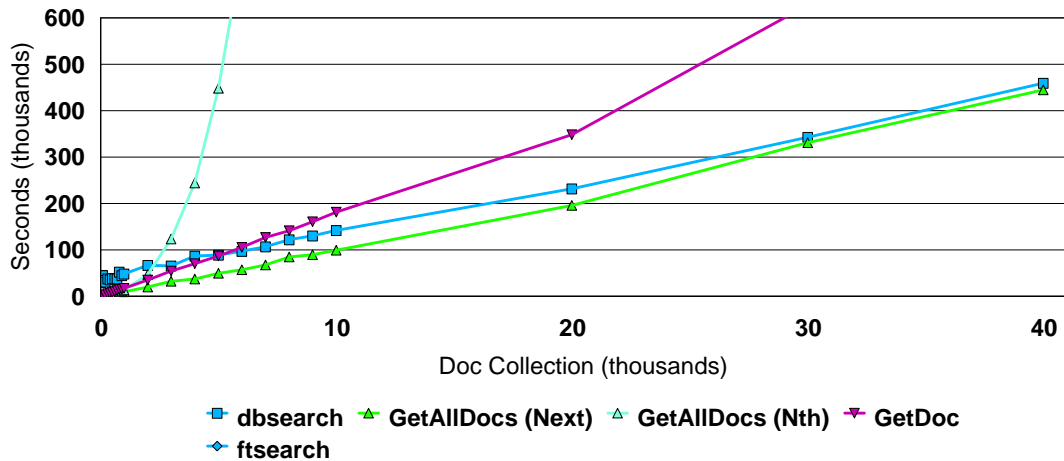
Agents

Getting collections in a 100k database



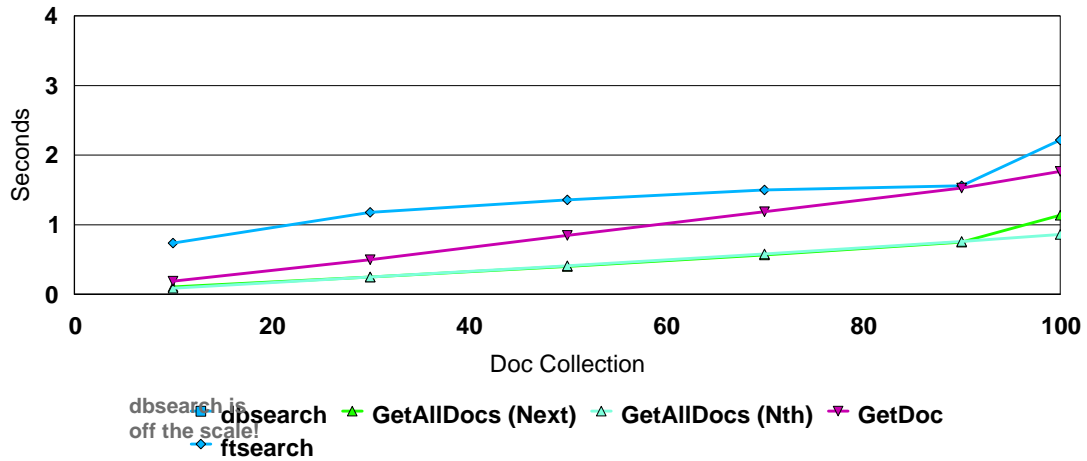
Agents

Getting collections in a 100k database



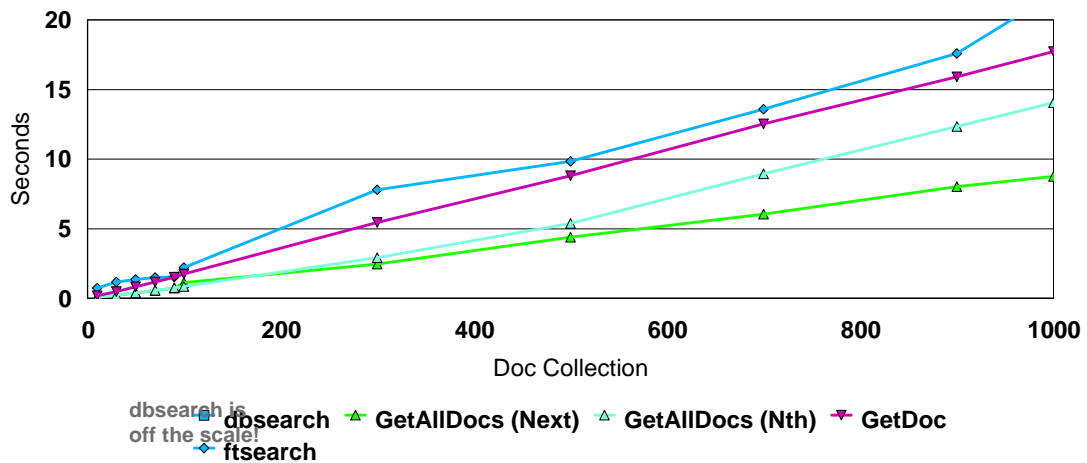
Agents

Getting collections in a 1million database



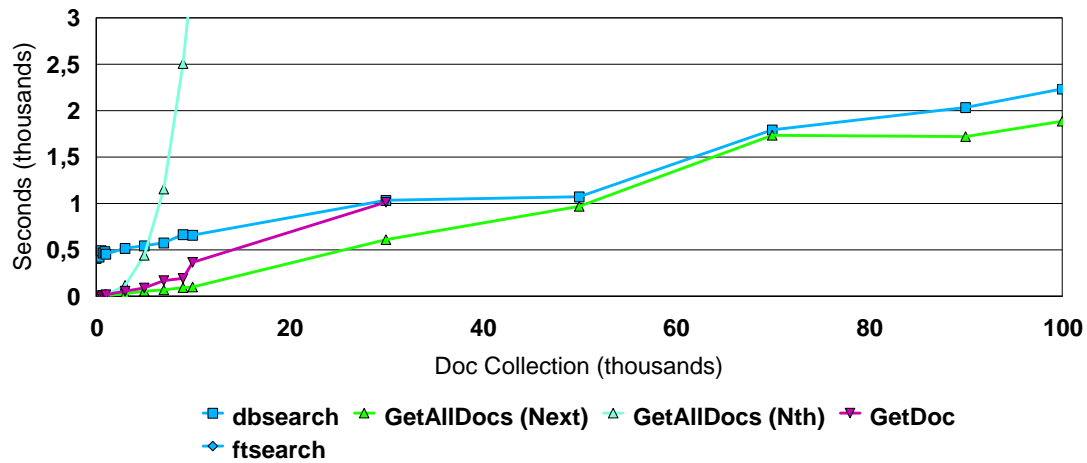
Agents

Getting collections in a 1million database



Agents

Getting collections in a 1million database



Agents - Some History

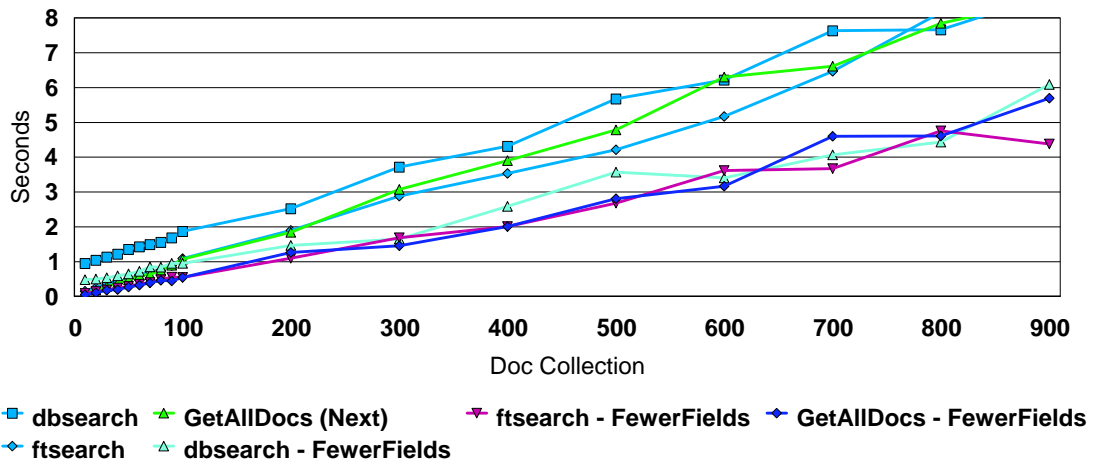
- R4
 - view.GetAllDocs (Next) best < 1%
 - db.search best between 1% and 15%
- R5
 - view.GetAllDocs (Next) best < 5-10%
 - db.search best between 5-10% and 30%
- Domino6
 - view.GetAllDocs (Next) always best
 - db.search almost as good ... larger collections, smaller databases

Agents - ftsearch

- R5
 - Fastest method, where applicable
 - Limited to 5000 results until FT_MAX_SEARCH_RESULT
- Domino6
 - Not quite as fast as view.GetAllDocs
 - Limited to 1000 results until ...

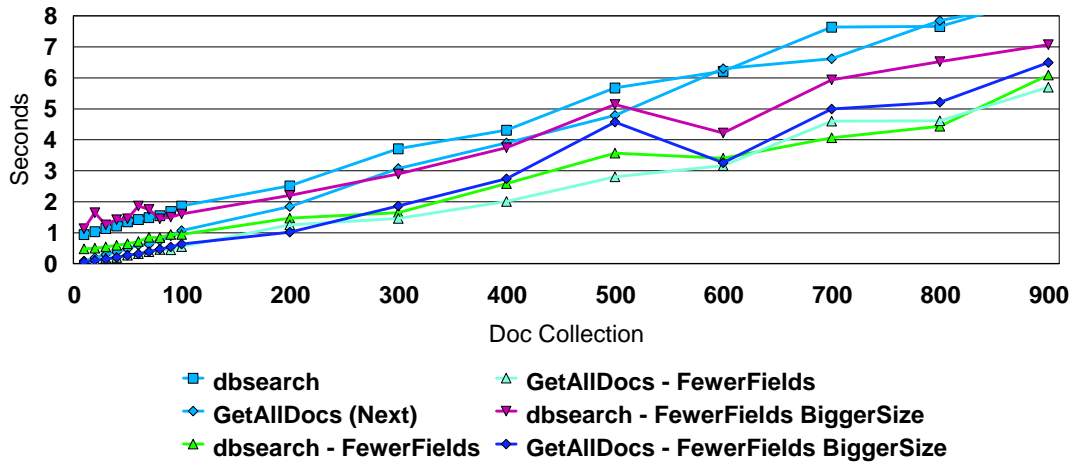
Fewer Fields

Getting collections in a 10k database



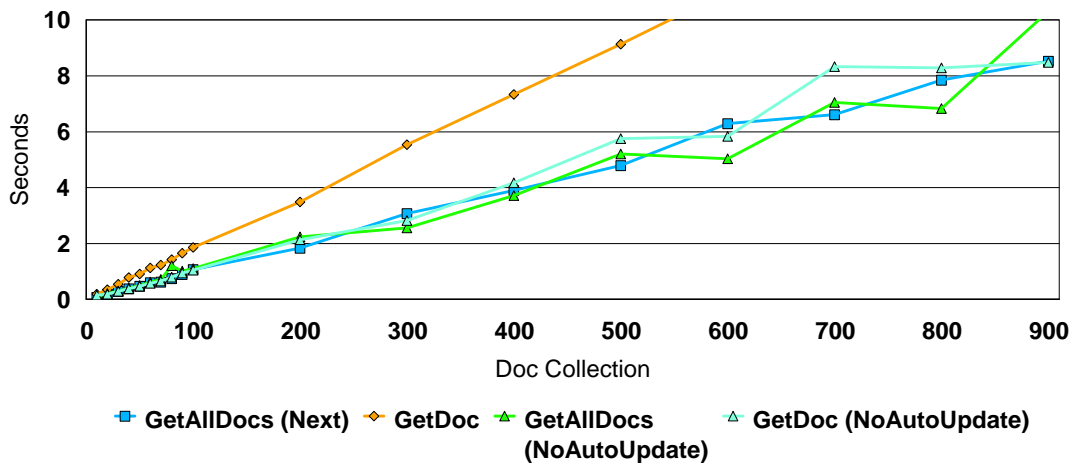
Fewer Fields - Bigger Documents

Getting collections in a 10k database



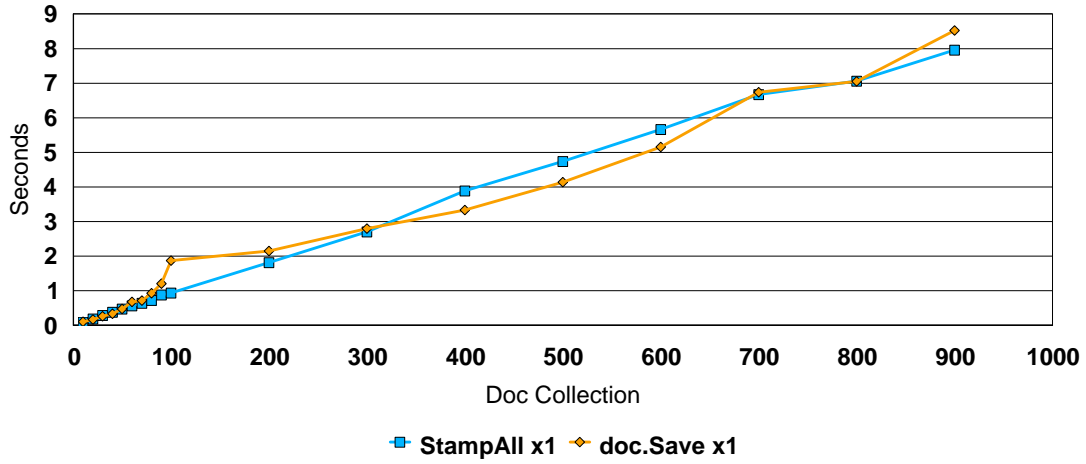
AutoUpdate = False

Getting collections in a 10k database



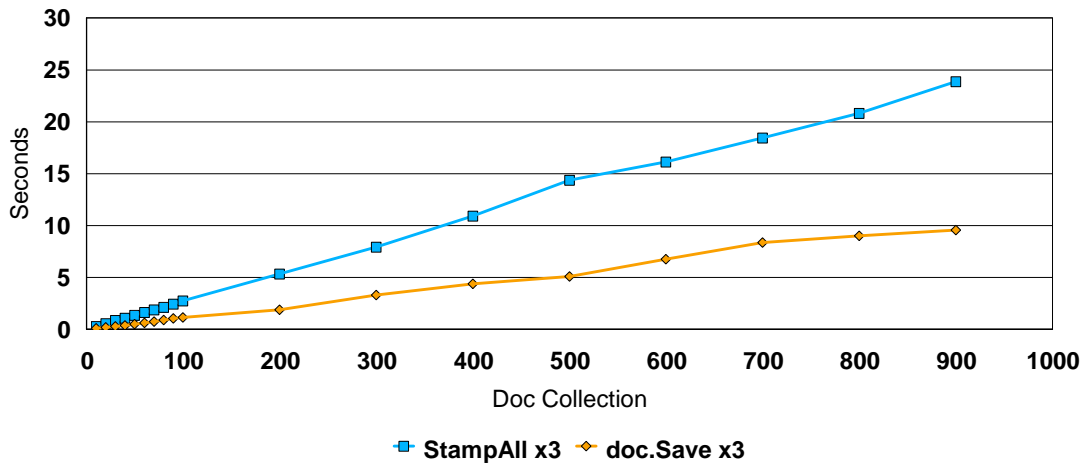
dc.StampAll vs. doc.Save

Getting collections in a 10k database



dc.StampAll vs. doc.Save

Getting collections in a 10k database



Document Selection

- No effect on any methods
 - ftindex updated
 - ftindex deleted
- But ... 4 1/2 minutes before the Initialize Event is kicked off if the ftindex is deleted (10k database)

tmpVAL vs. doc.Field

- Wrote code that references a single document
- Takes value from a single field
- Compares that value to 1000 strings in an array
 - Using doc.FieldName (0) in ForAll loop is 10 times more time consuming than setting tmpVAL = doc.FieldName (0)

ForAll vs. DoWhile vs. DoNext

- Wrote code that sorts values in an array with 1000 elements
 - ForAll 4.0 seconds
 - ForNext 6.1 - 6.6
 - DoWhile 7.8 - 8.2 seconds
- Note that if we have only 100 elements, times are approximately 1/10th of the figures above

Redim Preserve Array

- Wrote code that creates an array of 1000 elements
- Using redim preserve adds 50% (.3 vs. .2 seconds)

```
dim A () as string
do while i <= 999
  redim preserve A ( i )
  A ( i ) = SomeValue
  i = i + 1
loop
```

```
dim A ( 0 to 999 ) as string
do while i <= 999
  A ( i ) = SomeValue
  i = i + 1
loop
```

Next Steps

- Rewrite LotusScript code in Java
- More Tips
- 10million document database
- Forms testing (@Db formulas)
- Web App Dev testing

