

Hashes, Hashmaps, frames, structs, ...

Erlang User Conference Stockholm 2011

Björn-Egil Dahlberg, Kenneth Lundin



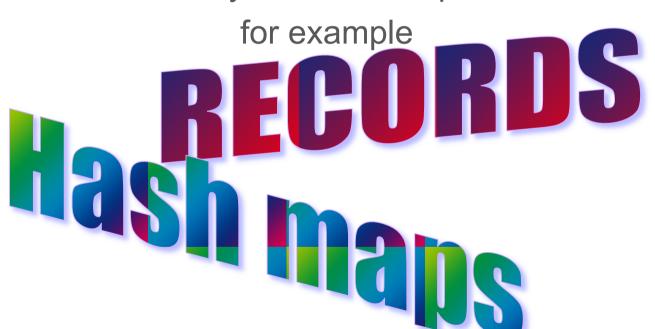


Background

Erlang/OTP makes the programmer productive

But!

there is always room for improvements



XYZ[\]

ŌÖרÙÚÛÜÝÞ ŸpÿĀāĂăąĆċĊċČ ŁŀŃńŊņŇňŌŐőŒ ŶŷŸŹźŻźŽžſŞş~ −≤≥fifl

ĠĠĠĠĠĠijijĬĶĶ ŞŞŢŢŤŤŪŪŮŮŰ

)

MHOΠΡCTYΦX



RECORDS

- > RECORDS are very frequently used
- Natural choice for representation of structured data
 - -few named fields
 - possibly many instances
- > Records are declared,
 - -have named data fields, with optional default values,
 - -can be matched
 - -special syntax for records.

XYZ[\] £¤¥¦§¨©ª«¬®¯°

iOÓרÜÜÜÜYÞ ĭþÿĀāÄăąĆćĊĊČ ĿŀŃńŊŋŇňŌŐŒŒ ŶŷŸŹźŻźŽžĬſŞş~ ∽≤≥fifl

sşşŢŢŤŤŪŪŮŮŰ

(ЛМНОПРСТУФ ІМНОПРСТУФХ



Records continued

```
-record(Person, {firstname, lastname, age}).
P1 = #person{firstname="John", lastname="Doe",age=45},
P2 = P1#person{firstname="Carl"},
FirstName = P2#person.firstname,
P3 = P2#person{age = P2#person.age+1},
myfunction(Person = #person{firstname = Firstname}) ->
```

BUT!

- not a distinct datatype (tuples)
- > declared in include files (compile time dependencies)
- > problems with upgrade
- > somewhat inefficient syntax
- > not suitable for large number of fields

XYZ[\] £¤¥¦§¨©ª«¬®¯°

ŐÖרÙÚÛÜÝÞ þÿĀāĂăąĆċĊċČ ĿŀŇńŊņŇňŌŐŏŒ ŶŷŸŹźŻźŽžƒŞş~ ∽≤≥fifl

ŞŞŢŢŤŤŪŪŮŮŰ

)



Hash maps

Associative arrays, Known from other languages:

- -Perl ,hashes,
- -Python dictionaries,
- -Clojure, maps struct maps

Typically many elements (keys)

Keys can typically be strings not so many instances

In Erlang we have proplists, dict, gb trees, gb sets etc.



Hash maps continued

proplists

```
P1 = [ {firstname, "John"}, {lastname, "Doe"}, {age,
    45}],

P2 = [{firstname, "Carl"}|proplists:delete
    (firstname, P1)],

FirstName = dict:fetch(firstname, P2),

P3 = dict:update(age, fun(V) -> V+1 end, P2),

BUT!
```

Inefficient memory wise

Bad performance for lookup when many keys.

Can not be matched, is not a data type of its own.



Hash Maps continued

dict

```
P1 = dict:from_list([ {firstname, "John"}, {lastname,
   "Doe"}, {age, 45}]),
P2 = dict:store(firstname, "Carl", P1),
FirstName = dict:fetch(firstname, P2),
P3 = dict:update(age, fun(V) -> V+1 end, P2),
```

BUT!

can not be matched,

rather inefficient (but better than proplists) to access for large number of keys.



Requirements/Goals HashMaps

Persistence (non destructive update)

Have determinable matching

Upgradeable in Applications

No compile time dependencies

Consume less memory than property lists (< 5n)

Ordering and equality

Complete info in runtime

Faster than gb_trees and dict

Ideally fast enough to be used as better records as well.

Focus performance on read, modify and write cycle.

Should be possible to declare each key in a named hashmap?

Have two types, named and anonymous?

'XYZ[\] £¤¥!\$"©ª«¬®¯°

OÓרÜÜÜÜYÞ þÿĀāĂăąĆćĊĊČ ĿľŃńŊņŇňŌŐőŒ ŶŷŸŹźŻźŽźſŞş~ ∽≤≥fifl

ŞŞŢŢŤŤŪŪŮŮŰ

СЛМНОПРСТУФ ЛМНОПРСТУФХ



Input

- > Richard O'Keefes frames proposal
 - -record replacement
- Joe Armstrongs struct proposal
 - -record replacement
- "Erlson" by Anton Lavrik (https://github.com/alavrik/erlson)
 - -syntax similar to records, using dict as representation
- > Phil Bagwell hash trees and vlists
 - -internal representations

/XYZ[\] £¤¥¦§¨©ª«¬®¯°

OOרUUUUYÞ þÿĀāÄăąĆćĊċČ ĿľŇńŊņŇňŌŐŏŒ ŶŷŸŹźŻżŽžfŞş^{~~} ∽≤≥fifl

ŞŞŢŢŤŤŪŪŮŮŰ

2 (ЛМНОПРСТУФ 1МНОПРСТУФХ



Syntax (some potential solutions)

Similar to records but without the type name (from Erlson)



Evaluation of Data structures

We have implemented prototypes and performed measurements using the Vlists and HAMT datastructures internally in the Erlang VM.

- > Records (=tuples)
- Vlists, a linked list of increasingly larger hashtables
- > HAMT, Hash Array Mapped Tries
- Hash table with exception lists

|XYZ[\] |£¤¥¦§"©ª«¬®¯°

ŌÖרÙÚÛÜŸÞ þÿĀāĂăąĆċĊċČ ŀŀŃńŊŋŇňŌŐŏŒ ŶŷŸŹźŻźŽžƒŞş~ -≤≥fifl

ĠĞĠĠĢĢĪĪĮĮĬĶĶ ŞŞŢŢŤŤŪŪŮŮŰ

) (ЛМНОПРСТУФ ЈИНОПРСТУФУ

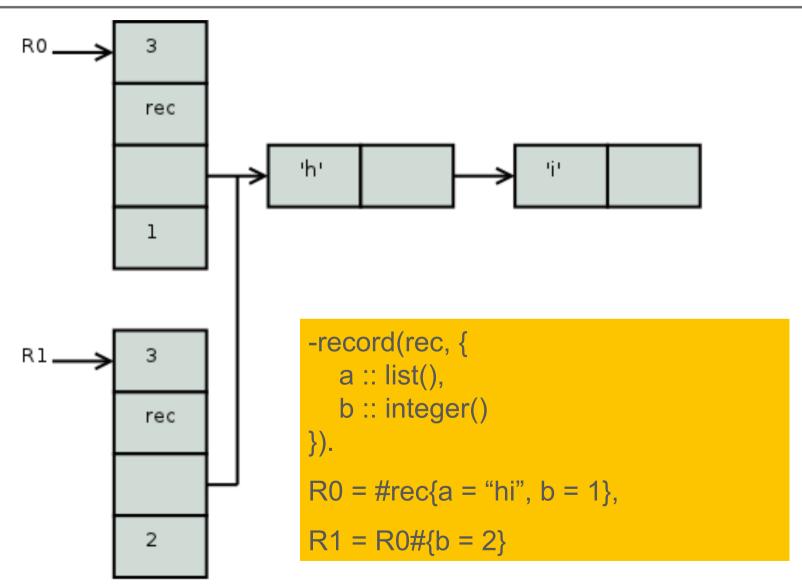


Tuples

VXYZ[\] ¢£¤¥¦§"©ª«¬®¯°

OŌÖרÙÚÛÜŸÞ ýÞÿĀāĂăąĆċĊċČ ŁŀŃńŊņŇňŌŐőŒ ŶŷŸŹźŻźžžſŞş¨ '∽≤≥fifl

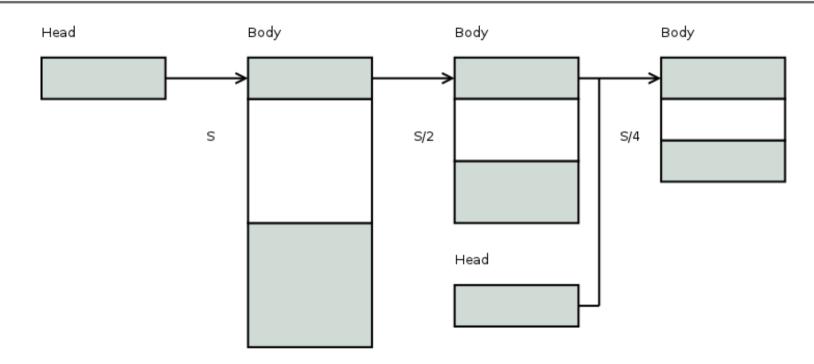
ĔĞĞĞĞÇÇÎIIIİKK SŞŞŢŢŤŪŪŮÜÜ ΣΤΥΦΧΨΪŸΆΈΗΙ Ω KIMHOΠΡCTYΦ IMHOΠΡCTYΦ VŲЪЪΘΘVVTҐ°2



Latest News from the Erlang group at Ericsson | Public | © Ericsson AB 2011 | 2011-06-10 | Page 12



VLISTS



- > A vhash list is a linked list of increasingly larger hashtables
- > Can be seen as a property list with very fast lookups (O(1))
- > Update and delete is expensive

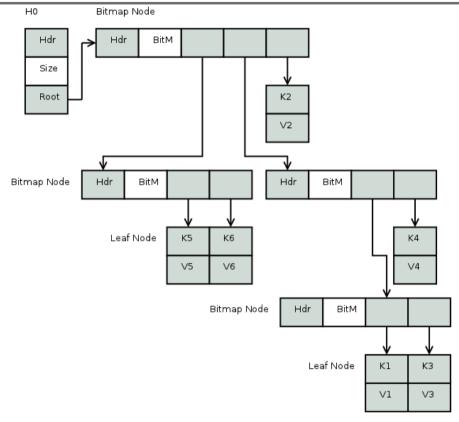
/ÿYZżZżZżƒŞş^{...} -≤≥fifl

エンホンロボジルニュョ

КЛМНОПРСТУФ ЛМНОПРСТУФХ ЎЦѣѣӨӨVVҐҐә



Hamt (Hash Array Mapped Tries)



- A shallow and memory compact tree (thanks to the bitmaps)
- > bitcount instruction in the processor is very useful for this.
- > O(log32) for lookup and updates, (almost as fast as a hash table for reasonable sizes)
- > Used for hashmaps in Clojure

XYZ[\] E¤¥¦§¨©ª«¬®¯°

ŌÖרÙÚÛÜÝÞ þÿĀāĂăąĆćĊĊČ ĿŀŃńŊņŇňŌŐőŒ ŶŷŸŹźŻźŽžĬţŞş[~] ∽≤≥fifl

ĠĠĠĠĠijijĬĶĬ ŖŢŢŤŤŪŪŮŮĹ

ΙΥΦΧΨΙΥΑΕΗ !

КЛМНОПРСТУФ ЛМНОПРСТУФХ ЎЏѢѢӨӨVVҐҐә



Hash table with Exception list

Arity nil Hashtable nil H0 nil Size nil Root

VXYZ[\] ¢£¤¥¦§¨©ª«¬®¯°

ÕÖרÙÚÛÜÜÝÞ þÿĀāĂăąĆċĊċČ ĿľŃńŊņŇňŌŐŏŒ ŶŷŸŹźŻżŽžƒŞş¨ ∽≤≥fifl

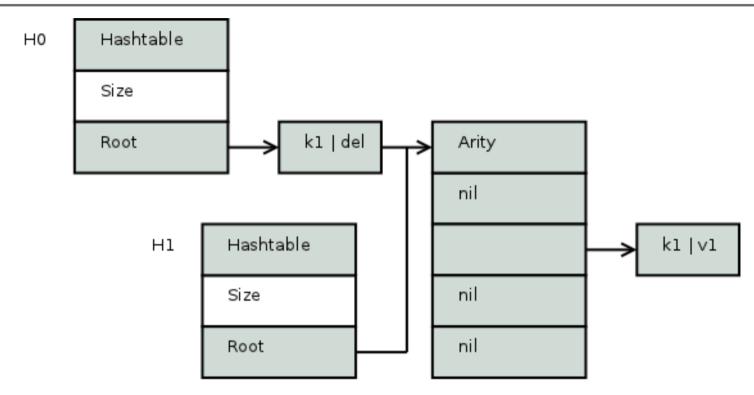
ĞĞĠĠĢĢĪĪĮĮİŀ ŞŞŢŢŤŤŨŨŮŮ

ТҮФХѰӀ҅ѶѦ҅ӔҺ

КЛМНОПРСТУФ ЛМНОПРСТУФХ ЎЏѢѢӨӨVVҐҐә



Hash table with Exception list update



- A hash table which prioritize the latest version
- Older versions gets an additional cost when accessing (traversing the exception lists)

XYZ[\] £¤¥¦§"©ª«¬®¯°

OOרUUUUYÞ þÿĀāĂăąĆćĊĊČ ĿľŃńŊņŇňŌŐőŒ ŶŷŸŹźŹżŽžƒŞş~ ∽≤≥fifl

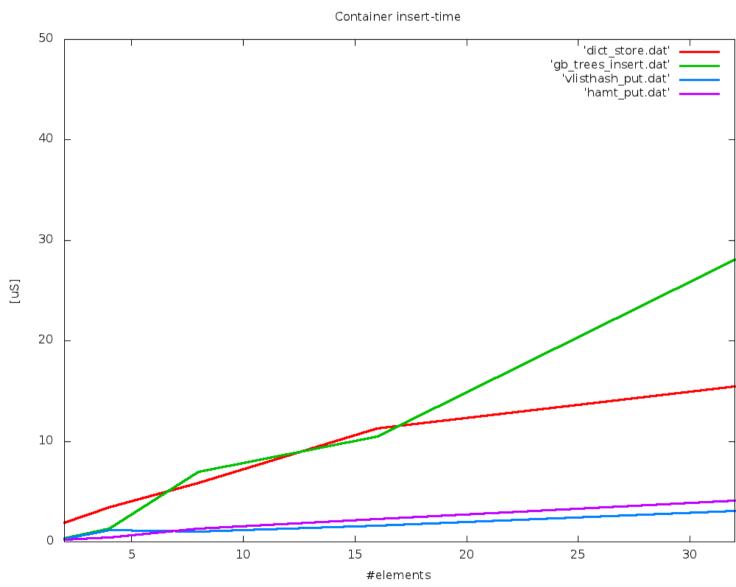
ĠĠĠĠĠĠijijijĶĬ ŞŞŢŢŤŤŪŪŮŮŰ

ГҮФХЧІҮА'ЕН :

MHOULLCLY MHOLLCLY MHOLLCLY MHOLLCLY MHOLLCLY MHOLLCLY MHOLLCLY



Measurements (insert, small)



XYZ[\] E¤¥¦§"©ª«¬®¯°

⁄þÿAāAăąCćCċC ĿŧŃńŊņŇňŌŐőŒ ŶŷŸŹźŻżŽžſŞş[~] −≤≥fifl

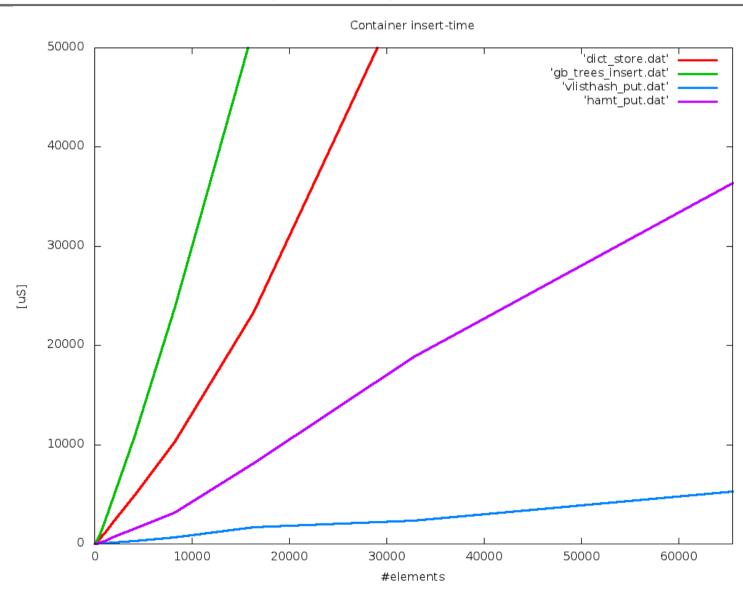
5511110000

КЛМНОПРСТУФ ЛМНОПРСТУФХ ЎЏѣѣӨӨVVҐҐә

Latest News from the Erlang group at Ericsson | Public | © Ericsson AB 2011 | 2011-06-10 | Page 17



Measurements (insert)



XYZ[\] @¤¥¦§"©ª«¬®¯°

ooxøddddip pyĀāĂăąĆćĊĊČ ĿľŃńŊņŇňŌŐőŒ ŶŷŸŹźŻżŽžĬţŞş[™] ∽≤≥fifl

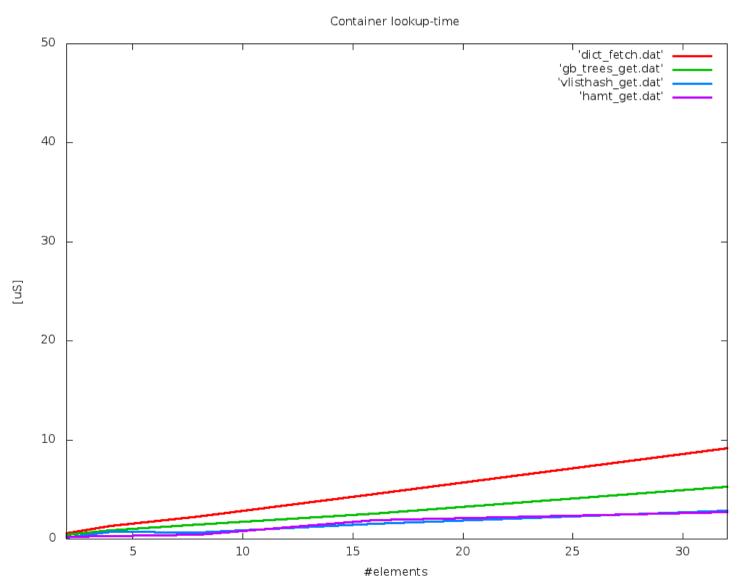
şşţŢŤŤŪŪŮŮ

Ω

КЛМНОПРСТУФ ЛМНОПРСТУФХ ЎЏѣѣӨӨVVҐҐә



Measurements (Lookup small)



XYZ[\] @¤¥¦§"©ª«¬®[—]°

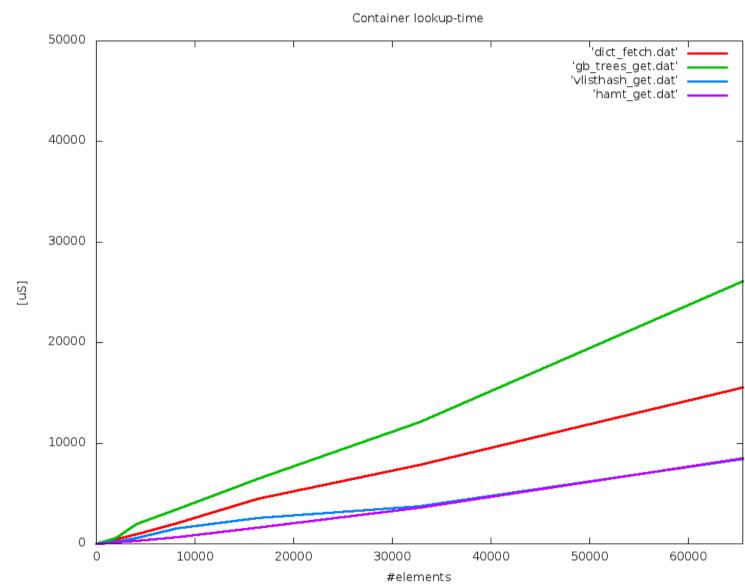
bÿĀāĂăąĆćĊĊČ ŀŇńŊņŇňŌŐσŒ ŶŷŸŹźŻżŽžďŞş° -≤≥fifl

ŞŞŢŢĦŪŪŪŪ

КЛМНОПРСТУФ ЛМНОПРСТУФХ ЎЏѣѣӨӨVVҐҐә



Measurements (Lookup)



XYZ[\] E¤¥¦§¨©ª«¬®¯°

i⁄þÿĀāĂăąĆćĊċČ ĿľŃńŊņŇňŌŐőŒ ŶŷŸŹźŻźŽžĬŞş[™] −≤≥fifl

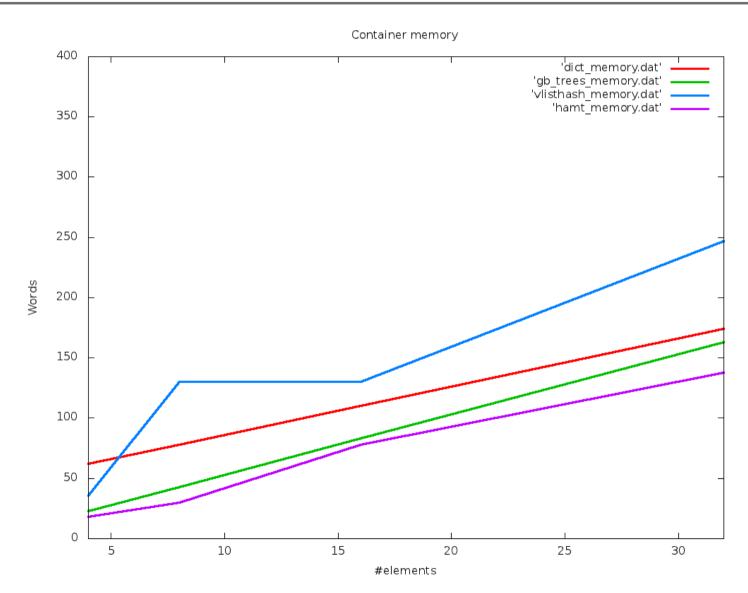
19911110000

КЛМНОПРСТУФ ЛМНОПРСТУФХ ЎЏѢѢӨӨVVҐҐә

Latest News from the Erlang group at Ericsson | Public | © Ericsson AB 2011 | 2011-06-10 | Page 20



Measurements (memory small)



XYZ[\] C¤¥¦§"©ª«¬®¯°

OÓרÜÜÜÜYÞ þÿĀāĂăąĆćĊċČ ĿŀŃńŊņŇňŌŐőŒ ŶŷŸŹźŻźŽžƒŞş^ −≤≥fifl

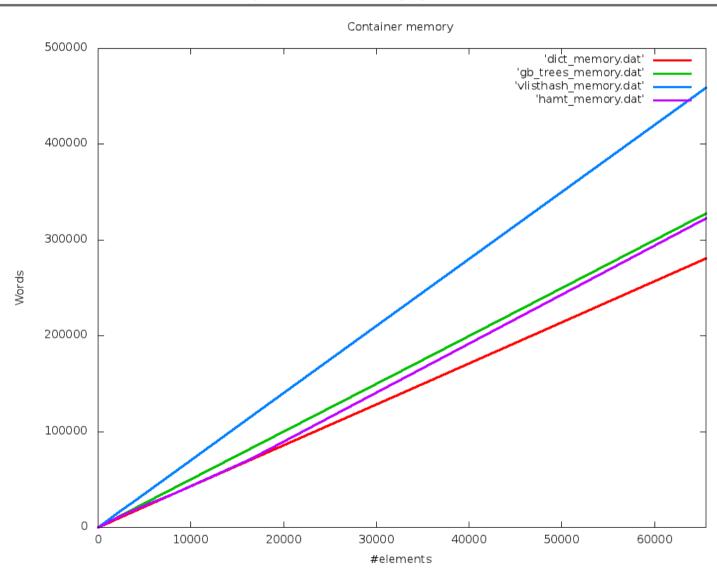
:GGGGĞĞII[[||Ķ ŞŞŢŢŤŤŪŪŮŮI

ΣΤΙΦΧΦΙΤΑΕΗ Ω КЛМНОПРСТУФ ЛМНОПРСТУФХ

ЎЏѣѣѲѲѴѴҐҐә



Measurements (memory)



XYZ[\] @¤¥¦§"©ª«¬®[—]°

OOרUUUUYP ⁄þÿĀāĂăąĆćĊċČ ĿŀŃńŊŋŇňŌŐŏŒ ŶŷŸŻźŻżŽžƒŞş~ ∽≤≥fifl

=GGGGĞĞII[[IK SŞŞŢŢŤŤŪŪŮŮI

ΤΥΦΧΨΪΫΆΈΉ Ω

КЛМНОПРСТУФ ЛМНОПРСТУФХ ЎЏѢѢӨӨVVҐҐә



Comparision of requirement fulfillment

Records gb_trees dict VList-Hash HAMT HWEL

Persistent
No Compile Time
Guardable
No Write-Barrier

Lookup
Insert
Replace

VXYZ[\] ¢£¤¥¦§¨©ª«¬®¯°

ŎŎרÙŰÛÜÝÞ ýþÿĀāĂăąĆċĊċČ ŁŀŇńŊņŇňŌŐŒ ŶŷŸŹźŻźžžſŞş¨ I∽≤≥fifl

ĔĞĞĠĠĢĢĪŢĮĮĬĶĶ ĠŞŞŢŢŤŤŪŪŮŮŰ

ΤΥΦΧΨΪΫΆΈΉ

КЛМНОПРСТУФ ЛМНОПРСТУФХ ЎЏѢѢӨӨVVҐҐә

Memory



Conclusions and way forward

- New better records
 - optionally named, possibly declared
 - few named fields
 - many instances
- > ----- and -----
-) Hashmaps
 - many "keys"
 - few instances
- > are two different things (but similar)
- > we want both
- > HAMT looks really promising for Hash Maps
- > but it might be hard to combine with one single representation
- > We have to decide what to address first
- > Probably some experimental implementation released during 2012

OOרUUUUYP þÿĀāĂăąĆċĊċČ ĿŀŇńŊņŇňŌŐŒ ŶŷŸŹźŻźŽžfŞş~ -≤≥fifl

\$\$}]]110000

(ЛМНОПРСТУФ ЛМНОПРСТУФХ (ПѣѣӨӨVVҐҐа



ERICSSON