

ALTREP and Other Things

Luke Tierney

Department of Statistics & Actuarial Science
University of Iowa

July 3, 2017





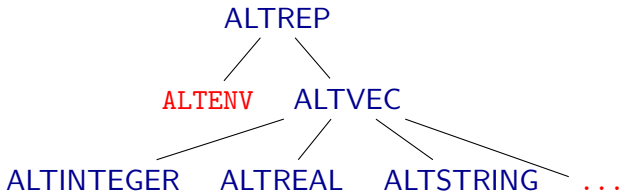
ALTREP: Alternate Representations for R Objects

- Joint work with Gabe Becker, Tomas Kalibera.
- The C level R implementation works with a fixed set of data types, e.g. [INTSXP](#), [REALSXP](#), [ENVSXP](#).
- Contents are accessed through a function/macro abstraction.
- [ALTREP](#) allows for alternate representations of these data types.
- To existing C code these look like ordinary R objects.
- Some of the goals:
 - allow vector data to be in a memory-mapped file or distributed;
 - allow compact representation of arithmetic sequences;
 - allow adding meta-data to objects;
 - allow computations/allocations to be deferred;
 - support alternative representations of environments.
- Current state is available in the [ALTREP SVN](#) branch.
- More details are available in [ALTREP.md](#) at the branch root.



Abstract Classes

- A set of abstract classes for R data types:



- The most specific classes correspond to R data types.
- Concrete classes specialize one of these.



Methods

General Methods

- **ALTREP** object methods:
 - Duplicate
 - Coerce
 - Length
 - Inspect
- The standard macros defer to these methods for **ALTREP** objects.
- **Duplicate** and **Coerce** methods can return **NULL** to fall back to the default behavior.



Methods

Vector Methods

- **ALTVEC** methods:
 - `Dataptr`
 - `Dataptr_or_null`
 - `Extract_subset`
- `Dataptr` may need to allocate memory; for now GC is suspended when calling the method.
- `Dataptr_or_null` will not allocate.
- `Dataptr_or_null` and `Extract_subset` can be used to avoid fully allocating an object



Methods

Specific Vector Methods

- Specific vector methods (patterned after JNI):
 - Elt
 - Set_elt
 - Get_region
 - No_NA
 - Is_sorted
 - and several others.
- Some numeric vector methods:
 - Min
 - Max
 - Sum
 - Prod



Changes to Existing Functions

- Some functions modified to avoid using `DATAPTR`:
 - `mean`
 - `min`
 - `max`
 - `sum`
 - `prod`.
- These use `Get_region` to process data in chunks.
- Many more functions could be modified along these lines.
- Subsetting has also been modified to avoid using `DATAPTR`.
- This means `head`, `sample`, for example, may avoid allocation.



Serialization and Package Support

- Classes can provide custom serialization by defining methods for
 - `Serialized_state`
 - `Unserialize`
- Packages can register `ALTREP` classes.
- Serialization records the package and class name.
- Unserializing loads the package namespace and looks up the registered class.
- A sample package implementing a memory mapped vector object is available on GitHub.



Sample Class Implementations

Compact Integer Vectors

- Vectors created by `n1:n2`, `seq_along` or `seq_len` can be represented compactly.

- In R 3.3.x (or 3.4.0 with JIT disabled)

```
> system.time(for (i in 1:1e9) break)
  user  system elapsed
0.258  1.141   1.400
> x <- 1:1e10
Error: cannot allocate vector of size 74.5 Gb
```

- In the ALTREP branch:

```
> system.time(for (i in 1:1e9) break)
  user  system elapsed
    0      0         0
> x <- 1:1e10
> length(x)
[1] 1e+10
```



Sample Class Implementations

Deferred String Conversions

- Converting integers or reals to strings is expensive.
- In `lm` and `glm` default row labels on design matrices are created but rarely used.
- The `ALTREP` branch
 - modifies the internal `coerce` function to return a deferred string conversion object;
 - this class has a `subset` method that returns another deferred conversion object.
- For `lm` or `glm` with $n = 10^7$ and $p = 2$ this produces a 5 to 10 fold speedup.
- Deferred evaluation could be useful in many other settings as well.



Sample Class Implementations

Memory Mapped Vectors

- The **ALTREP** branch includes sample classes for memory mapped integer and real vectors.
- The file can be opened for reading and writing or in read-only mode.
- When used by **ALTREP**-aware code these will not result in allocating memory for holding all the data.
- Using non-aware functions may result in attempts to allocate large objects.
- The class provides an option for signaling an error when the raw data pointer is requested.



Sample Class Implementations

Wrapper Objects

- Currently changing an attribute on a shared vector requires a copy of the vector data.
- Wrapper can hold the new attribute value and a reference to the original object to access its data.
- Wrapper objects can also be used to attach meta-data, such as
 - is the vector sorted;
 - are there no **NA** values.
- The **sort** function returns a wrapper that records that the vector is sorted.



Some Implementation Details

- **ALTREP** objects are allocated as **CONS** cells with an **altrep** header bit set.
- Standard macros, like **LENGTH** look at this bit to decide whether to dispatch.
- To allow efficient scalar identification there is also a **scalar** bit,
- With the ALTREP changes operations like **DATAPTR**, **STRING_ELT**, and **SET_STRING_ELT** now might cause allocation.
- Eventually code should be rewritten to allow for this.
- For now, GC is suspended in these allocations.



- Deferred evaluations/allocations are very useful, but:
 - allocation failures can be delayed and come at unexpected times;
 - operations may produce unexpected large allocations, e.g. `log(1:1e10)`;
 - some situations can lead to repeated evaluations.
 - Memory mapping issues:
 - serialization failure when the file is not available;
 - some settings might need a conversion layer (e.g. a file of 8-bit integers).
 - Length and data address consistency; can these change during object lifetime?
- Deferred edits might be useful for improving complex assignment performance.



Changes Needed in R-devel

- `ALTREP` needs one or two new header bits.
- This requires a binary-incompatible header change.
- Because of alignment issues, adding 32 bits to the header does not increase object sizes on (most if not all) 64-bit platforms.
- This also allows room for a reasonable size reference count.
- This does seem like a good opportunity to also reserve 64 bits for the vector length fields (which does increase vector object sizes).
- There is now a mechanism in place (`R_INTERNALS_UUID`) that prevents loading packages with compiled code created by a binary-incompatible R.
- It would be good to make this change fairly soon; if there are other header adjustments needed these could happen now also.



Transition Stages for R-devel

- Rough order of steps:
 - Header changes.
 - Add support for basic framework, packages.
 - Modify some functions to take advantage of support.
 - Create **ALTREP** object within R-devel.
- Header change will be most disruptive; best to do it soon.
- Will need to check against CRAN, Bioconductor at each stage.



- Reference counting:
 - more maintainable;
 - allow less duplicating;
 - may help improving complex assignment performance.
- Compilation:
 - reduce remaining interpreted/compiled differences;
 - pre-compile packages by default;
 - more optimization opportunities.
- Integer and logical sum:
 - Currently `sum(x > 0)` can return `NA` for a long vector.
 - Allow `sum(x)` to return a double?