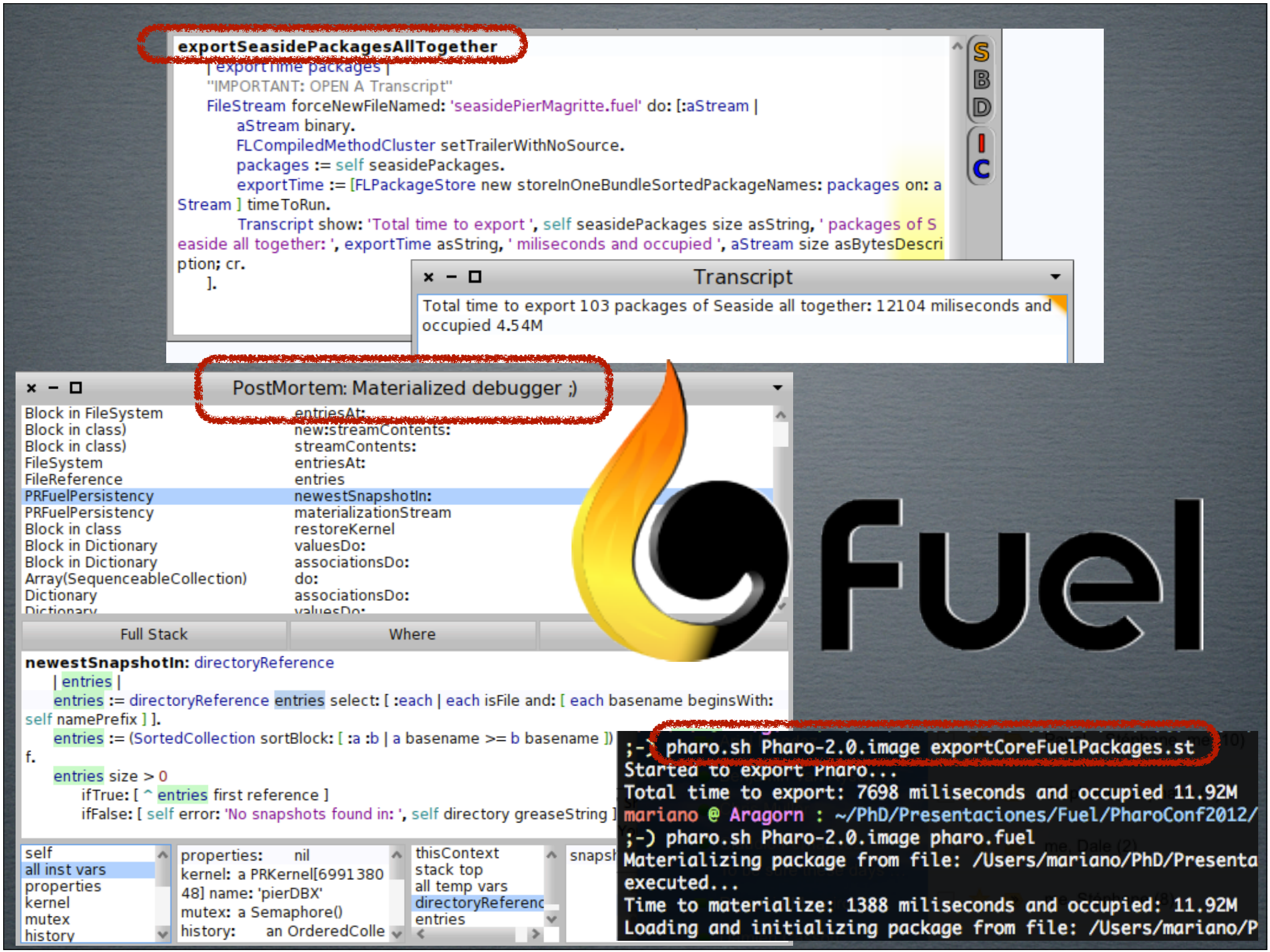
exportSeasidePackagesAllTogether
| exportTime packages |
"IMPORTANT: OPEN A Transcript"
FileStream forceNewFileNamed: 'seasidePierMagritte.fuel' do: [:aStream |
aStream binary.
FLCompiledMethodCluster setTrailerWithNoSource.
packages := self seasidePackages.
exportTime := [FLPackageStore new storeInOneBundleSortedPackageNames: packages on: aStream] timeToRun.
Transcript show: 'Total time to export ', self seasidePackages size asString, ' packages of Seaside all together: ', exportTime asString, ' miliseconds and occupied ', aStream size asBytesDescription; cr.
].
S
B
D
I
C
Transcript
Total time to export 103 packages of Seaside all together: 12104 miliseconds and occupied 4.54M
PostMortem: Materialized debugger ;)
Block in FileSystem entriesAt:
Block in class) new:streamContents:
Block in class) streamContents:
FileSystem entriesAt:
FileReference entries
PRFuelPersistency newestSnapshotIn:
PRFuelPersistency materializationStream
Block in class restoreKernel
Block in Dictionary valuesDo:
Block in Dictionary associationsDo:
Array(SequenceableCollection) do:
Dictionary associationsDo:
Full Stack
Where
newestSnapshotIn: directoryReference
| entries |
entries := directoryReference entries select: [:each | each isFile and: [each basename beginsWith: self namePrefix]].
entries := (SortedCollection sortBlock: [:a :b | a basename >= b basename]) f.
entries size > 0
ifTrue: [^ entries first reference]
ifFalse: [self error: 'No snapshots found in: ', self directory greaseString]
self
all inst vars
properties
kernel
mutex
history
properties: nil
kernel: a PRKernel[6991380 48] name: 'pierDBX'
mutex: a Semaphore()
history: an OrderedColle
thisContext
stack top
all temp vars
directoryReferenc
entries
Fuel
;-) pharo.sh Pharo-2.0.image exportCoreFuelPackages.st
Started to export Pharo...
Total time to export: 7698 miliseconds and occupied 11.92M
mariano @ Aragorn : ~/PhD/Presentaciones/Fuel/PharoConf2012/
;-) pharo.sh Pharo-2.0.image pharo.fuel
Materializing package from file: /Users/mariano/PhD/Presenta
executed...
Time to materialize: 1388 miliseconds and occupied: 11.92M
Loading and initializing package from file: /Users/mariano/P

- Is a fast, well-designed, concrete, general-purpose and flexible binary serializer.
- Can serialize/materialize not only plain objects but also classes, traits, methods, closures, contexts, packages, etc.
- Support for global references.
- Large number of hooks: ignore certain instance variables, substitute objects by others, post and pre serialization and materialization actions.
- Supports class rename and class reshape.
- 90% (approx. 500 unit tests) of test coverage.
- Large suite of benchmarks.
- Object-Oriented design.
- No VM support needed.
- Modular (clear division of packages).

Some numbers...

- Time to serialize all 2138 classes of Pharo: 7.7 seconds generating a file of 7.6 MB. Time to materialize that: 1,1 second.
- Time to serialize all 1489 classes of Seaside/Pier/Magritte: 12 seconds generating a file of 4.5 MB. Time to materialize that: 1,1 seconds.