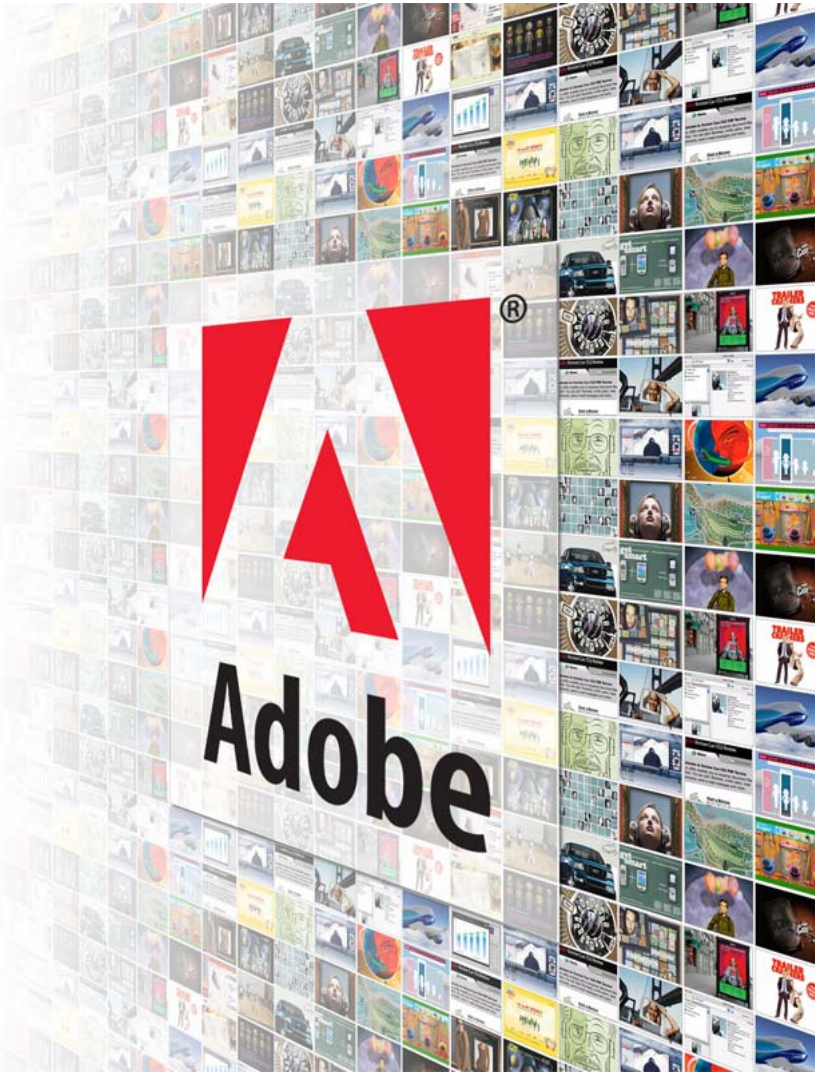
ActionScript 3 Performance Tuning

Matt Chotin

Product Manager

Adobe Systems



Adobe

Overview

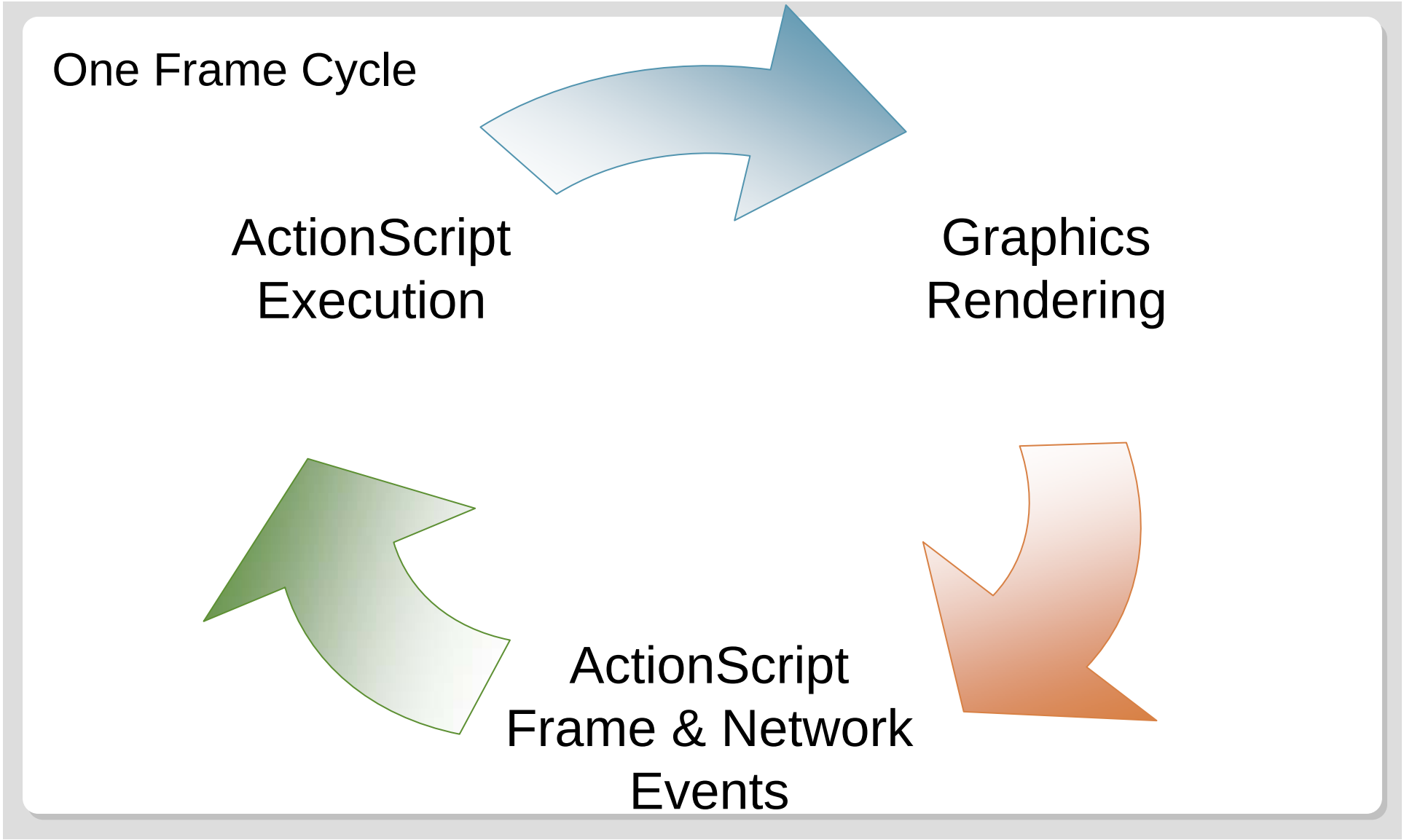
- Premature Optimization
- How Flash Player and ActionScript work
- Strong Typing
- Other random goodness
- Measuring performance

Adobe

Premature Optimization

- "More Computing sins are committed in the name of efficiency than any other reason including blind stupidity" – W.A. Wulf
- "Premature Optimization is the root of all evil." - Hoare and Knuth
- "Bottlenecks occur in surprising places, so don't try to second guess and put in a speed hack until you have proven that's where the bottleneck is." - Rob Pike

How Flash Player works

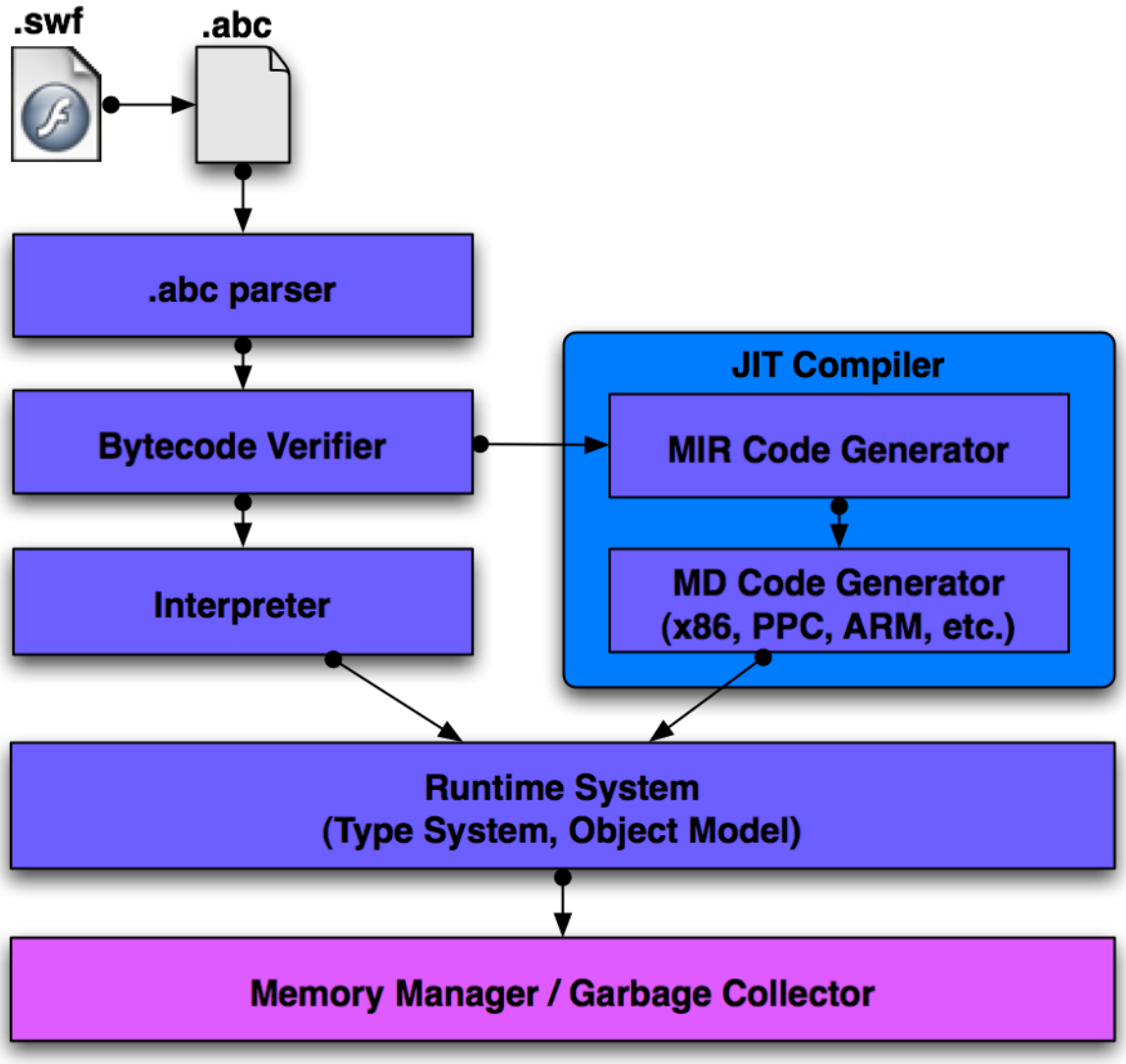


Adobe

How Flash Player works

- Flash Player presents a single thread of execution
- Flash Player is multi-threaded!!!
- Rendering and ActionScript are executed sequentially.
- The player will do everything you ask it to do.
- Do to much in one frame cycle a script-timeout occurs.
- Defer AS execution to next frame cycle with:
    - mx.core.UIComponent.callLater()

AVM2 Architecture



Adobe

AVM2 Just-in-Time Compiler (JIT)

- First pass
  - intermediate representation (MIR) produced concurrent with verification
  - early binding
  - constant folding
  - copy and constant propagation
  - common sub-expression elimination (CSE)
- Second pass
  - native code generation
  - instruction selection
  - register allocation
  - dead code elimination (DCE)

Strong Typing improves productivity

- Simplified debugging
- Compiler will throw compile-time errors
- Code hinting in Flex Builder is based on strong typing

Adobe

Typing optimizes bytecode!

- Typing optimizes bytecode!
- Typing optimizes bytecode!
- Typing optimizes bytecode!

Adobe

Typing optimizes bytecode!

- Typing affects member access
- Strong references are equivalent to C++ member access
  - Must use dot operator for member
  - Must use strong typing
- Weak references are a hash lookup
  - No typing
  - Use of ["member"] access

Adobe

Member Access Examples

- instance.x
- instance[“x”]

- “This stuff happens a lot!!!”

Adobe

Type everything!

- All variables, member and local
- All function return values
- All parameter types in functions or error handlers
- All loop counters
- If you see “var”, better see “:<type>”
- Even if your class is dynamic
  - VM can take advantage of slots for members that are pre-compiled
  - New properties will require the hash lookup

If type is unknown, check before using

- Sometimes you receive a variable where you don't know the type
- Try coercing the variable to the type you want before accessing
- An VM error can be 10x slower than the time it takes to execute the test or coercion

Array Member Access

- Arrays include fast path for dense portion
- Member access is as fast as C++ in dense portion
  - var a:Array = [1,2,3,4,5];
  - a[1000] = 2010;
  - a[1001] = 2011;
  - a[2]; //FAST PATH
  - a[1000]; //SLOW PATH
- A miss in an array lookup is slow

Demo

- Let’s see some exaggerated running times

Adobe

Typing with int, uint, Number

- Avoid implicit type conversion
- Math operations can induce type conversion
- Sometimes just using Number is more efficient

Adobe

Leverage local variables

- If you intend to use a variable more than a few times, store it in a local
- var person:Person;
- if (person.firstName == “Matt” || person.firstName == “Ted” || person.firstName == “Alex”)…
- var fname:String = person.firstName;
  if (fname == “Matt” || fname == “Ted” || fname == “Alex”)
- Imagine if that query is in a loop, every dot represents a lookup

Data Binding (Flex)

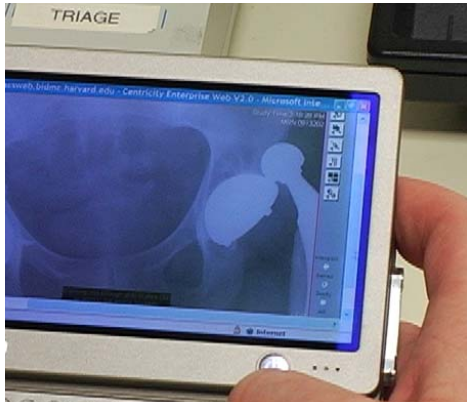
- Binding expressions take up memory and can slow down application startup as they are created
- Bindings are general case so they cannot optimize execution paths (for example knowing that the first part of a binding expression won't change and therefore doesn't need to be watched)
- For one-time use it may be more efficient to do an assignment in code rather than binding
- Binding makes your development faster, but can take up space in your application

Measuring performance

- flash.utils.getTimer():int
- New Flex 3 Performance Profiler

Adobe's Mission:

To revolutionize how the world engages with ideas and information



Adobe