JCOD Optimizing Technology

A project subsidized by Japan IPA Agency

Vania Joloboff
Groupe Silicomp
http://www.ri.silicomp.com/

Java To Native Compilation for Embedded Systems

- Improve performance by order of magnitude
- The only practical approach in embedded systems is to have compiler outside the device as a compiler hardly fits into devices such as a mobile phone, a set top box or a printer…
- *Flash compiler*: the device is already in the hands of the customer, who wants to download new applications, typically stored in flash

Java Compilation on the market today

- Very good performance improvement
  - 25 times faster for TurboJ on Caffeine
- Significant code expansion
  - requires up to 8 times more memory
- Linked with RTOS dependent and VM dependent code
  - application port is not free...

Optimizing Optimization

- Real-world applications are not CPU only, they do I/O's and involve garbage collection
- Consider an application spending
  - 80 in CPU, 10 in I/Os, 10 in other things such as garbage collection
  - A high performance compiler that would go 40 times faster would reduce to
    - 2 + 10 + 10 = 22 that is, performance gain 78%
  - A less optimizing compiler 8 times faster will reduce to
    - 10 + 10 + 10 = 30 that is, performance gain 70 %

Optimizing the trade-off

- But for much better cost !
  - Assuming memory cost represents 20 out of 100
  - With compiler requiring 4 times more memory
    - Device cost is 80 + (4*20) = 160
    - 78 % better performance for 60% cost increase
  - With JCOD type of technology
    - Device cost is 80 + (1.75*20) = 115
    - 70 % better performance for 15% cost increase

New Approach: JCOD Optimization

- Download application after device is shipped
- Ease application portability
- Ease application deployment
- Minimize memory cost
  - do not compile everything
  - generate small code

JCOD principles

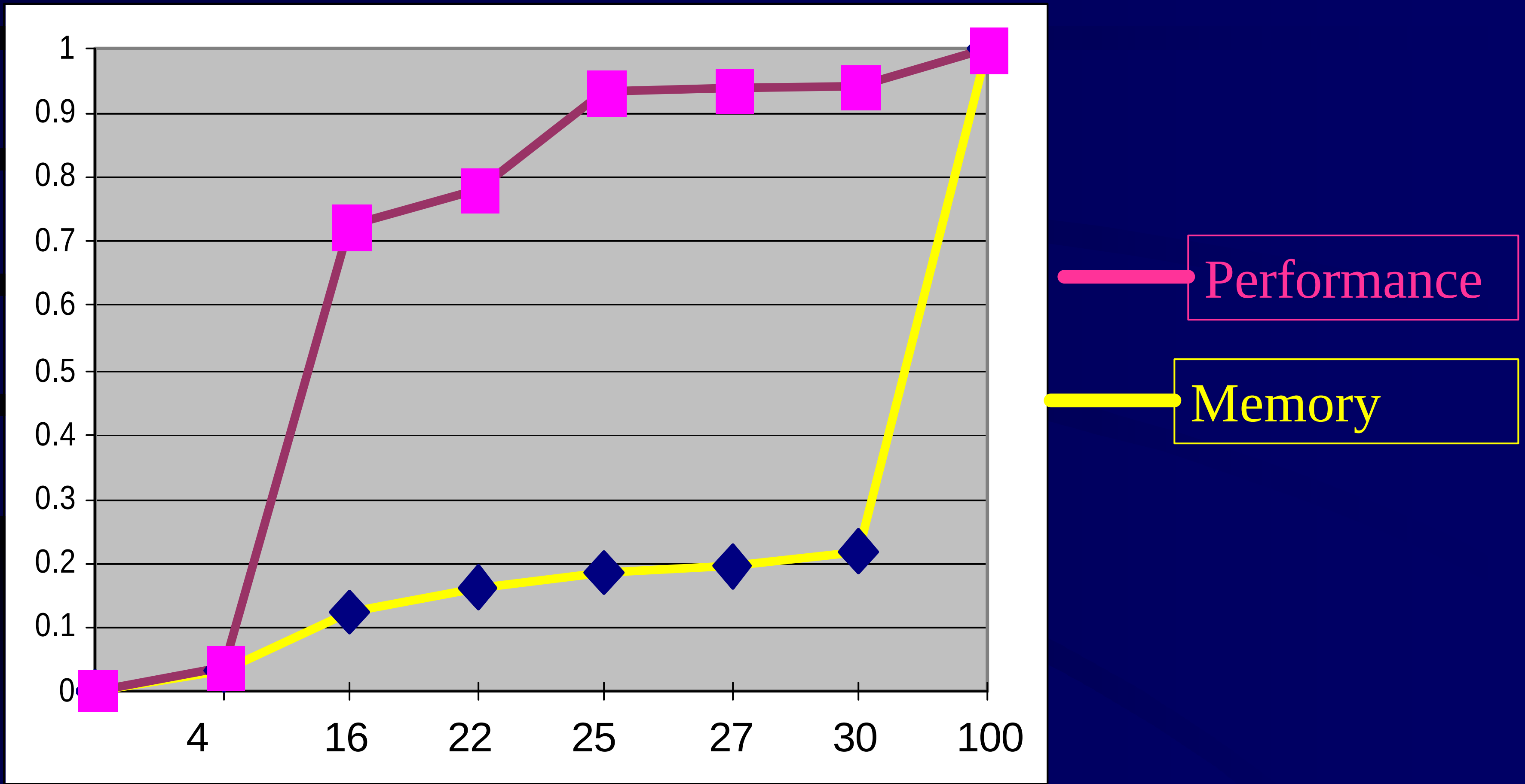
- Do not compile everything
  - Profile the application
    - at run time or before hand
  - Smart compiler to generate small code
- Ease application deployment
  - Use the .class file to store native code
- Ease application portability
  - Provide VM independence, RTOS independence
  - Independence between compiler version and VM version

Early Results

- On SH processor (16 bits instructions) without float support

% of app. Compiled	% of memory expansion	Performance increase
0	0%	0%
4.0%	12%	37%
16.0%	45%	560%
22.0%	58%	630%
25.0%	67%	730%
27.0%	70%	733%
30.0%	80%	733%
100.0%	264%	790%

Normalized Chart



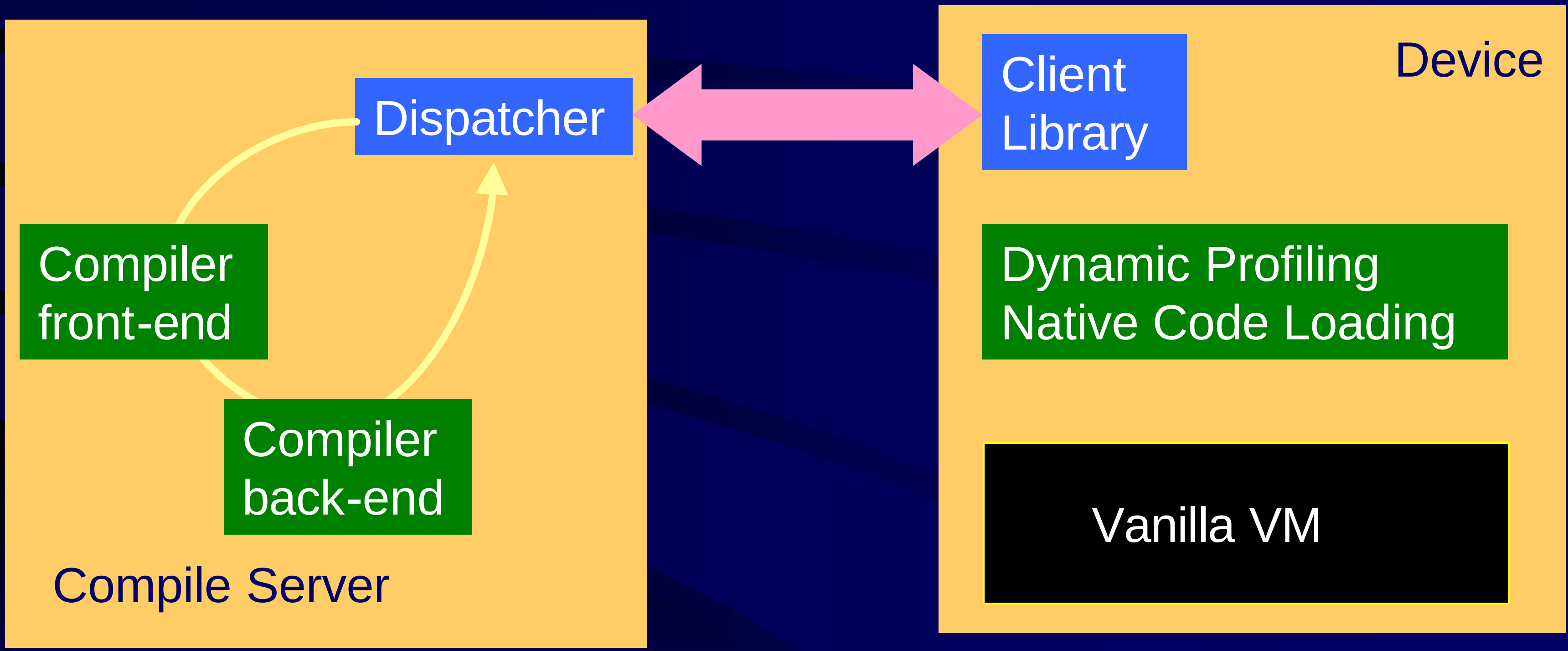
Dynamic Profiling

- Run the application
- Compute for every methods (or only some) a method cost
  - Method cost based on loop cost and method calls cost
- Compile only methods with a very high method cost

Application Deployment

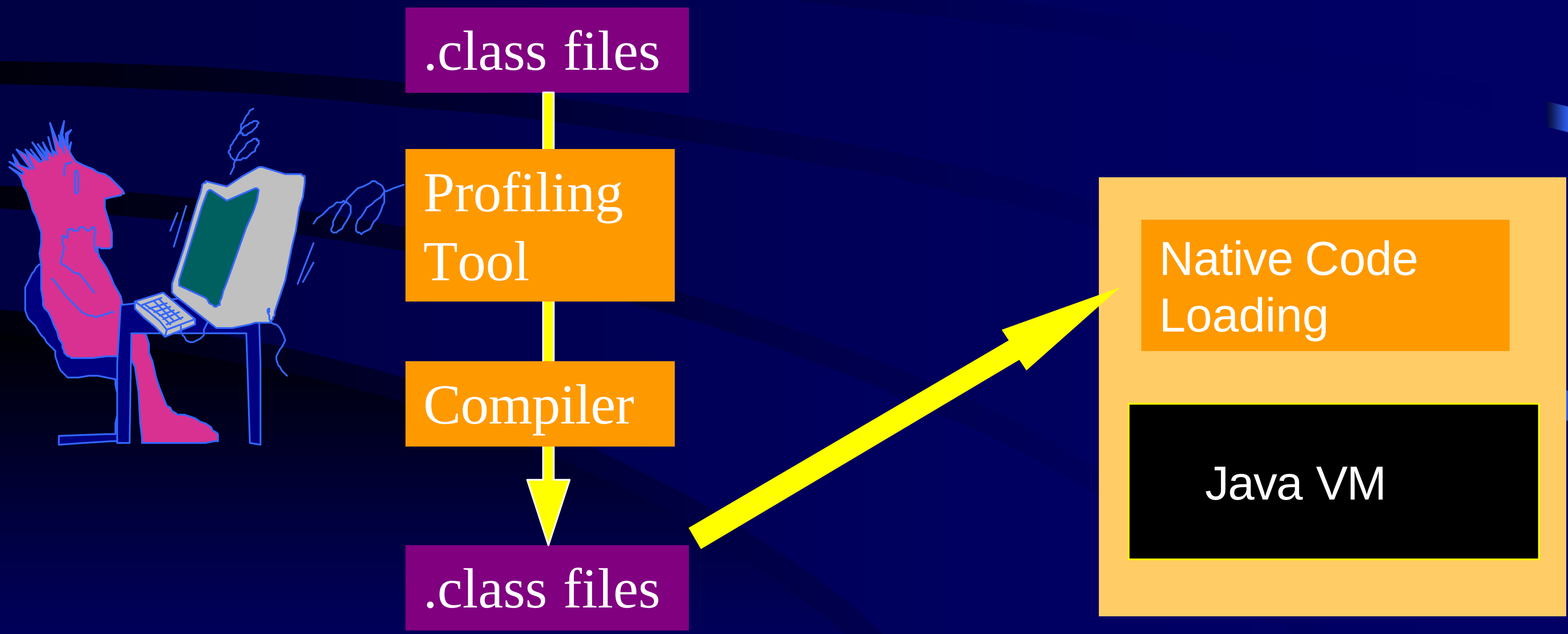
- Two modes:
  - Mostly connected appliances: Dynamic Mode
    - Use a network compile server that is available
  - Occasionally connected devices: Static Mode
    - Do not use compile server. Compile in advance

Dynamic Mode



Static Mode

- User or Developer runs profiling tool to determine what to compile



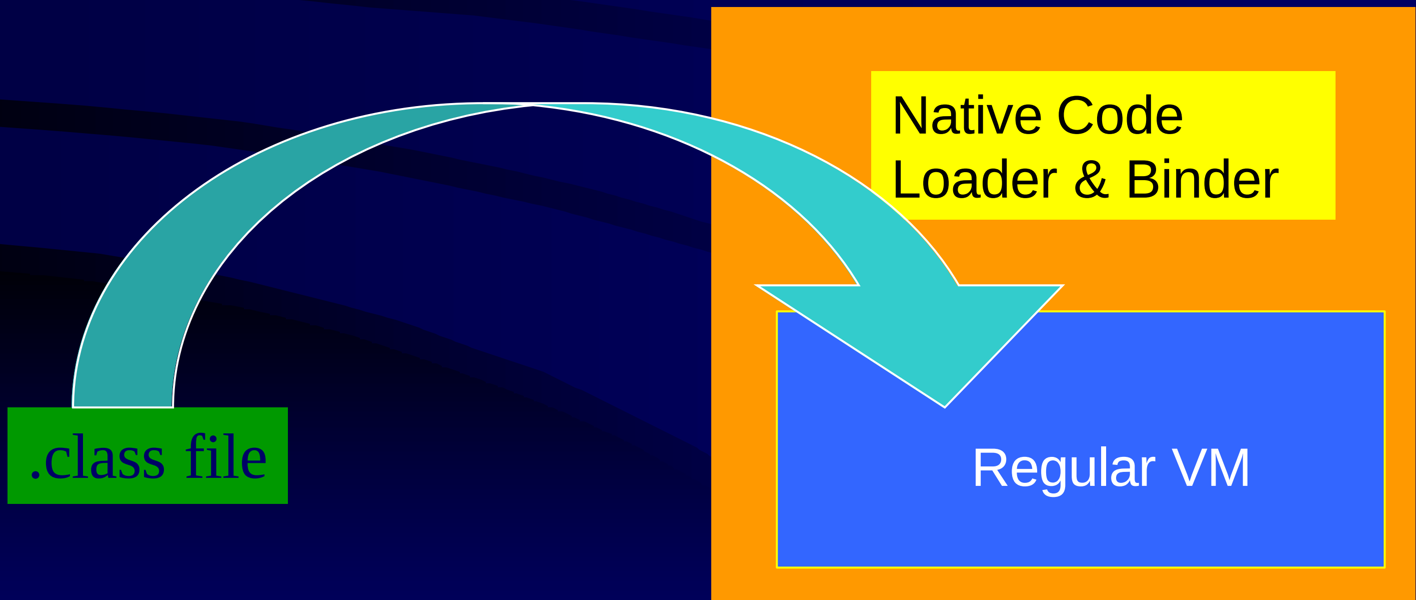
Target Independence

- Define an object code format which is
  - independent from the RTOS
  - independent from the VM
- Idea:
  - an object format with late binding
  - the compiler generates processor dependent code stored back into the class file

.class file → Flash compiler → new .class file

Target Independence

- The code generated by the compiler is RTOS independent and VM independent and loaded by an extended native loader



Flash Compiler Technology Advantages

- Optimize memory/performance trade-off
  - 10 times faster for twice as much memory is feasible
- Same delivery mechanism as vanilla Java: class file
  - Decision to compile can be postponed up to the last moment (users can compile, not only software vendors) including the VM itself
- Applications run on any VM, just faster with those supporting compiled code loader
  - Users or Developers don't have to worry upon VM or RTOS dependence

Japanese Contact

- Junkyo Fujieda
  REGIS Inc
  TEL: +81-44-201-5210
  FAX: +81-44-200-7091
  E-mail: jack@re-gis.com