Java and C Performance Comparison on Palm OS PDA device

- Basic numerical computation

  example: Multidimensional matrix computation
- Memory management

  example: Java primitive type vs. Java Object
- SpecJVM98, SciMark and JkernelMark official benchmarks
- Measure in execution time and memory usage

Zhi-Kai Xin
zxin@cs.columbia.edu

Motivation

- Java is widely adopted in embedded system to develop various applications
  - PDA, wireless phone, and game console
- Java is easy to learn and powerful
  - Rich set of library
  - Platform independent
  - Easy integration

Zhi-Kai Xin
zxin@cs.columbia.edu

Motivation cont.

- Popular JVM for PDA or Wireless device
  - Sun Microsystems J2ME (KVM)
    - CLDC and MIDP for Palm OS 5+
  - IBM J9 VM
  - HP ChaiVM
  - iPAQ PersonalJava (Jeoda)
- C uses Cygwin with PRC-TOOL or CodeWarrior for PDA application development

Zhi-Kai Xin
zxin@cs.columbia.edu

Related works

- Sosnoski [1][2] Java and C/C++ performance comparison and analysis
  - Java object allocation wastes memory space and takes too long
  - But improvement can be made by using primitive types instead of Java object
  - Use array instead of java.util.Vector
  - Avoid creating new object in large software

Zhi-Kai Xin
zxin@cs.columbia.edu

Related works cont.

- Moreira et al. [3] numerical computation
  - Multidimensional matrix addition and multiplication
  - Java Array incurs overhead of runtime checking.
  - Java multidimensional array is array of arrays (slower indexing)
  - Java outperforms C if Java array runtime checking is disabled

Zhi-Kai Xin
zxin@cs.columbia.edu

Questions to be answered by end of this project

- Which language has better performance on Palm OS PDA device, Java or C?
- If C is better, how bad is Java's performance on PDA? Is it acceptable performance?
- What improvements can be made to Java to run better on PDA?

Zhi-Kai Xin
zxin@cs.columbia.edu

Development Environment

- Sony Clie (Palm OS Device)
  - 33MHz
  - 16MB RAM
- Java Development IDE
  - Java wireless toolkit
  - CodeWarrior

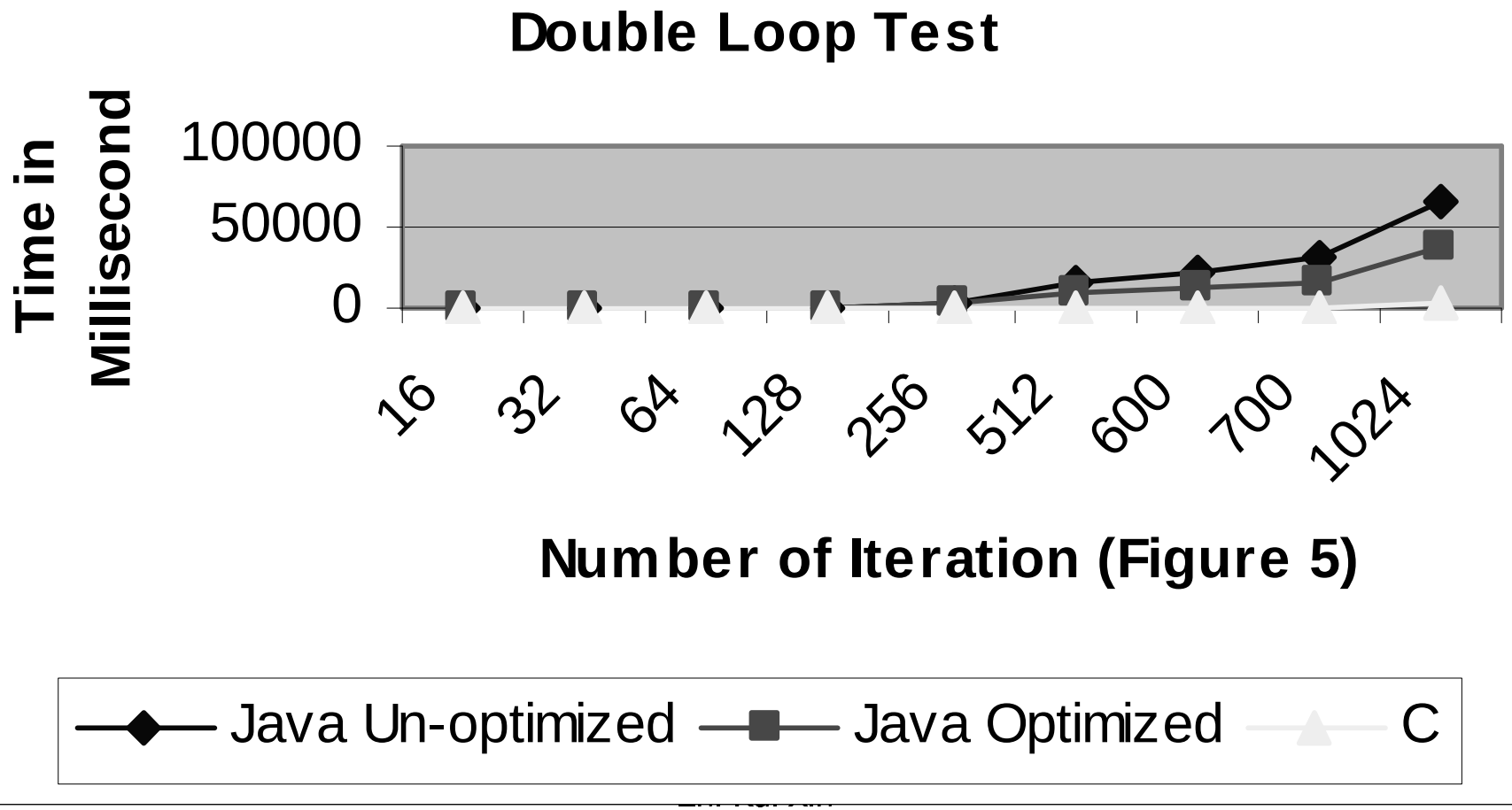
Zhi-Kai Xin
zxin@cs.columbia.edu

Results and Analysis

Columbia University
CS4995-2 Embedded
System Project
Presentation

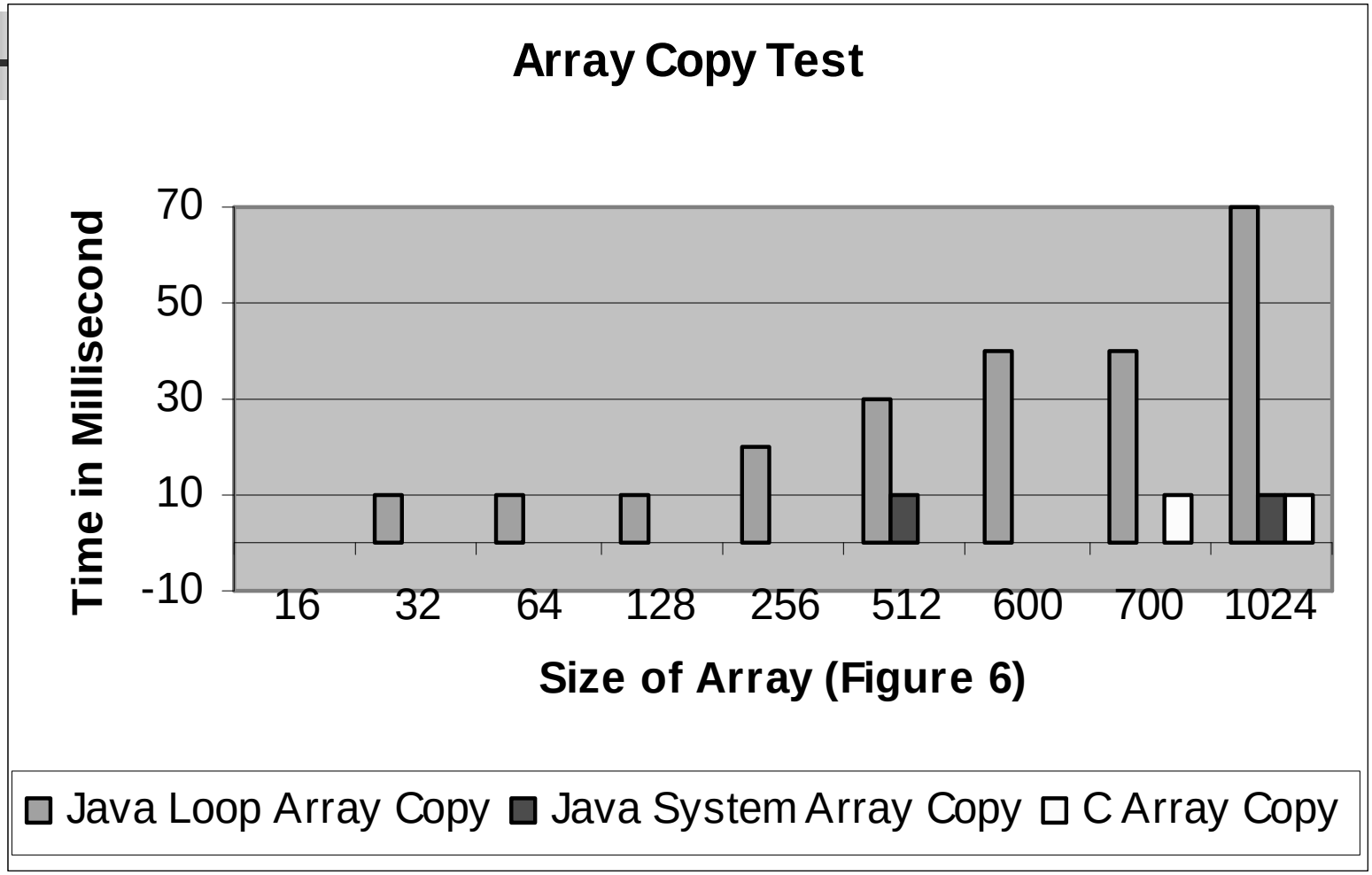
Zhi-Kai Xin

Looping



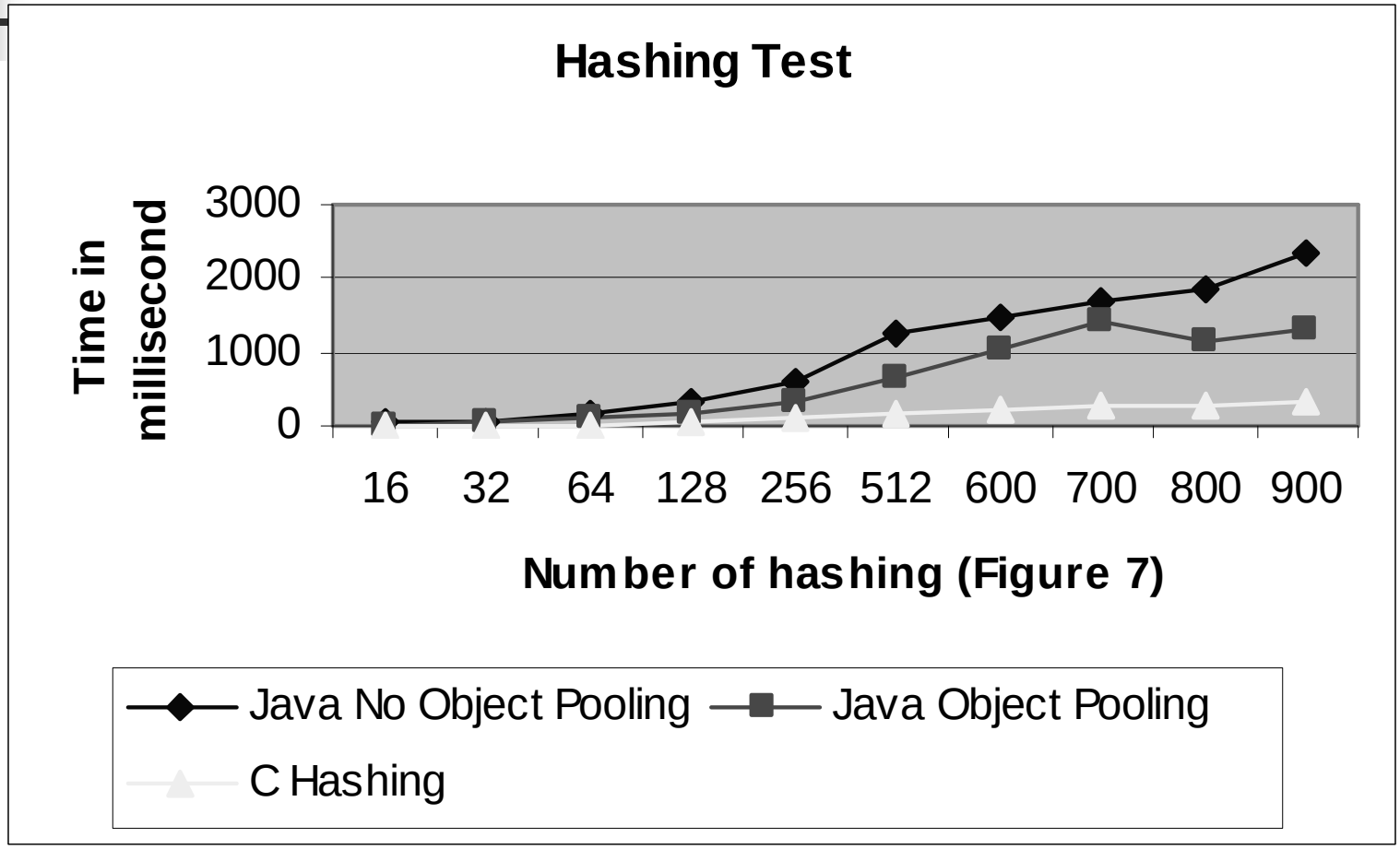
zxin@cs.columbia.edu

Array Copy



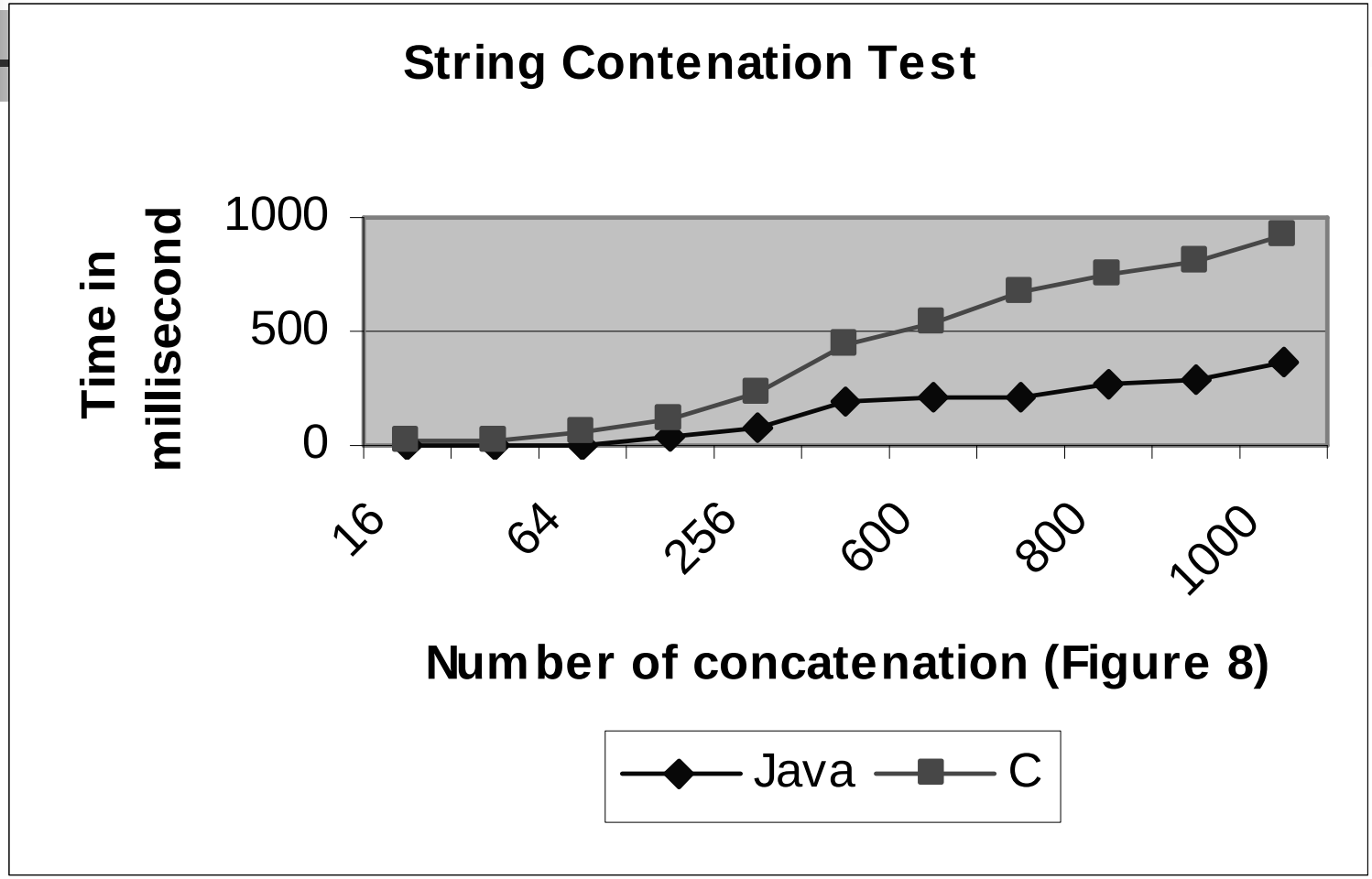
Zhi-Kai Xin
zxin@cs.columbia.edu

Hashing



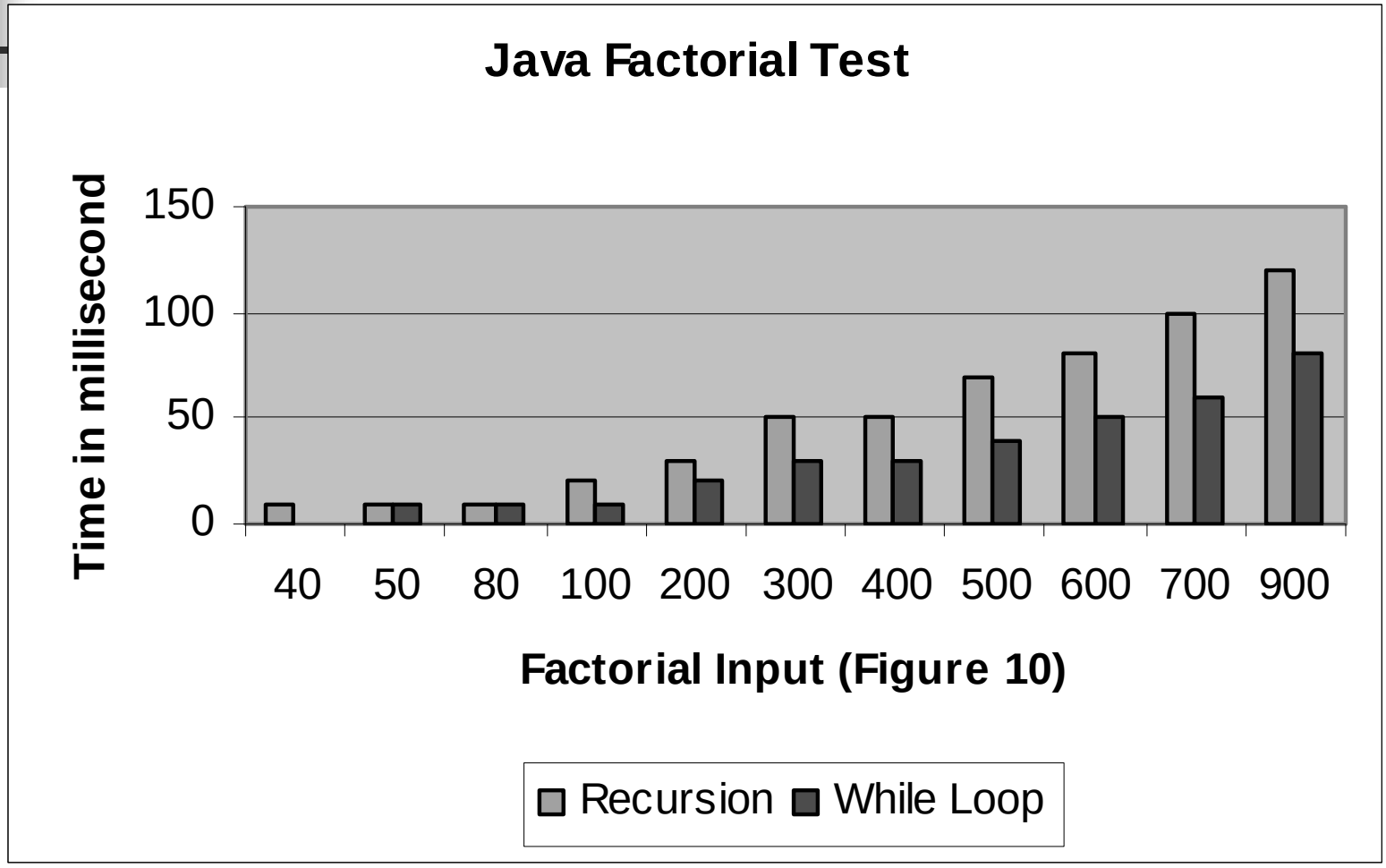
Zhi-Kai Xin
zxin@cs.columbia.edu

String concatenation



Zhi-Kai Xin
zxin@cs.columbia.edu

Factorial



Zhi-Kai Xin
zxin@cs.columbia.edu

Conclusion

- Java performance can be improved by fine tune Java program such as:
  - Avoid recursion
  - Avoid having array access within nested loops
  - Use object pooling, avoid create new object, thus avoid garbage collect, especially within loops
  - Try use Java's build in methods, avoid re-writing your own routines

Zhi-Kai Xin
zxin@cs.columbia.edu

Conclusion cont.

- Java has a rich set of APIs for fast development
- Definitely worth to use on PDA device software development

Future works

- Look into Java's IO and network performance on PDA devices

Zhi-Kai Xin
zxin@cs.columbia.edu

Source code

The project source code can be found at:

http://www.cs.columbia.edu/~zxin/cs4995-2/final/project

Zhi-Kai Xin
zxin@cs.columbia.edu

References

- [1] Dennis M. Sosnoski. Java performance programming, Part 1: Smart object-management saves the day. Java World, November 1999. http://www.javaworld.com/javaworld/jw-11-1999/jw-11-performance.html

[2] Dennis M. Sosnoski. Java Performance Comparison with C/C++. Sosnoski Software Solution, Inc. This report was presented in 1999 JavaOne conference. http://www.sosnoski.com/Java/Compare.html

[3] J. E. Moreira, S. P. Midkiff, and M. Gupta. A Comparison of Java, C/C++, and FORTRAN for Numerical Computing. IEEE Antennas and Propagation Magazine, Vol. 40, No. 5, October 1998.