S67

IMS e-business Offering Performance Test Results

IMS e-business Performance

Sinmei DeGrange
sinmei@us.ibm.com

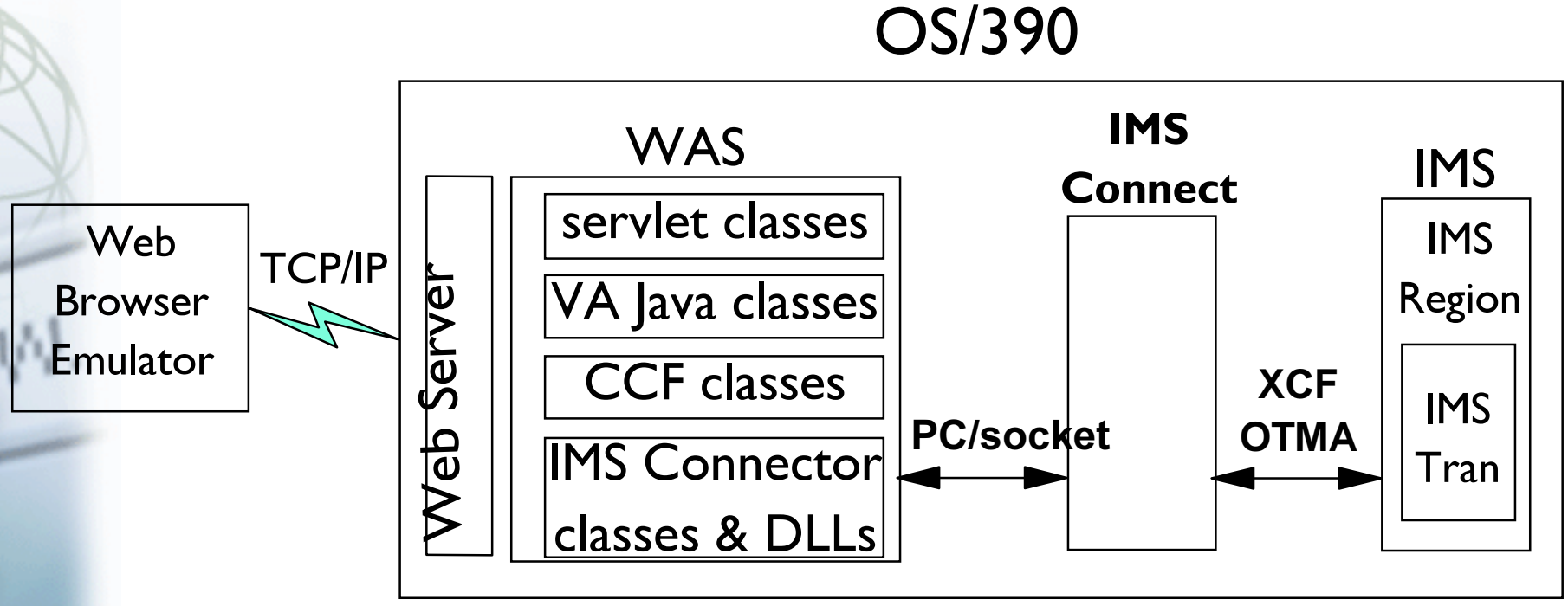
Miami Beach, FL October 22-25, 2001

RETURN TO INDEX

IMS Connector for Java

Large Message Proof of Concept

IMS Connector for Java



OS/390 V2R10
IBM HTTP Server V5R3
WAS 3.5
IMS V7

IMS Connector for Java

Large message

- Test environment
  - **IMS Fast Path Workload**
  - **A COBOL application had 12 Fast Path DEDB reads**
  - **The input message was 500 bytes and the average output message was 21-22K bytes**
    - The original input message was 42 bytes and the average output message was 429 bytes
    - Input and Output messages were padded with alpha numeric characters to inflate the message sizes
  - **The input and output messages were single segment**
    - IMS FP limits its message size to the size of the IMS log buffer size
    - Our log buffer size was 22K
    - Therefore, the output message size could not be larger than 22K

IMS Connector for Java

Small vs. Large message

- Test environment
  - Took advantage of the fastest available processor, 2064-116 16-way
  - Put the IMS LOG datasets, OLDS, SLDS and WADS on IBM Enterprise Storage Server 2105-E20 DASD by themselves
    - this turned out to be a key factor in achieving a better transaction rate
  - All databases were on RAMAC Virtual Array (RVA) 9393 DASD

IMS Connector for Java

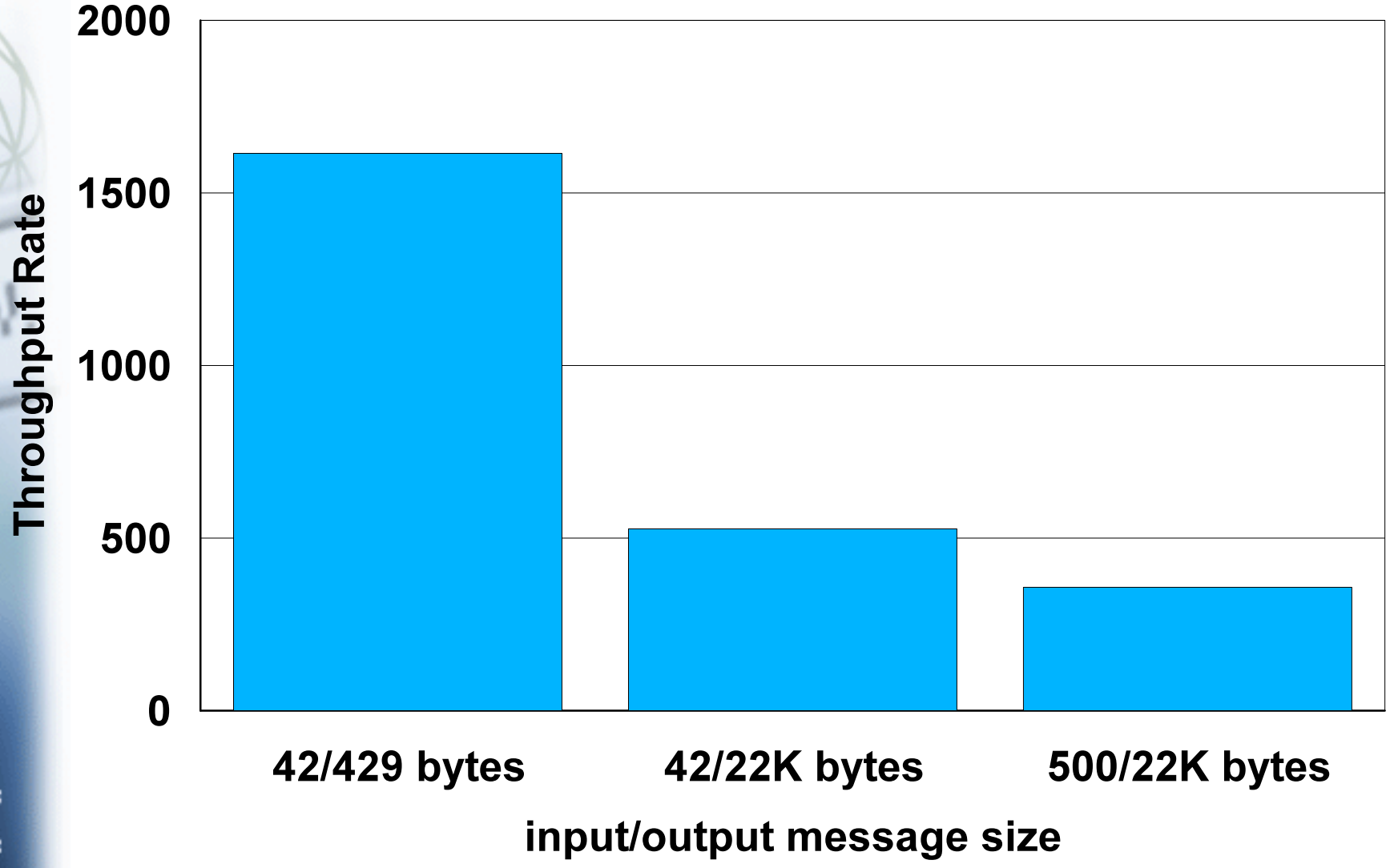
Small vs. Large message

- Test environment
  - OS/390 2.10, IBM HTTP Server 5.3, WebSphere 3.5, SDK 1.3.0 and VA Java 3.5.3
  - IMS Connector for Java **Local Option** was used
  - IMS, IMS Connect, HTTP Server and WebSphere were on the IBM 2064-116 single image 16-way server
  - AKStress, a WebSphere development tool, was used to simulate the Web Browser clients

IMS Connector for Java

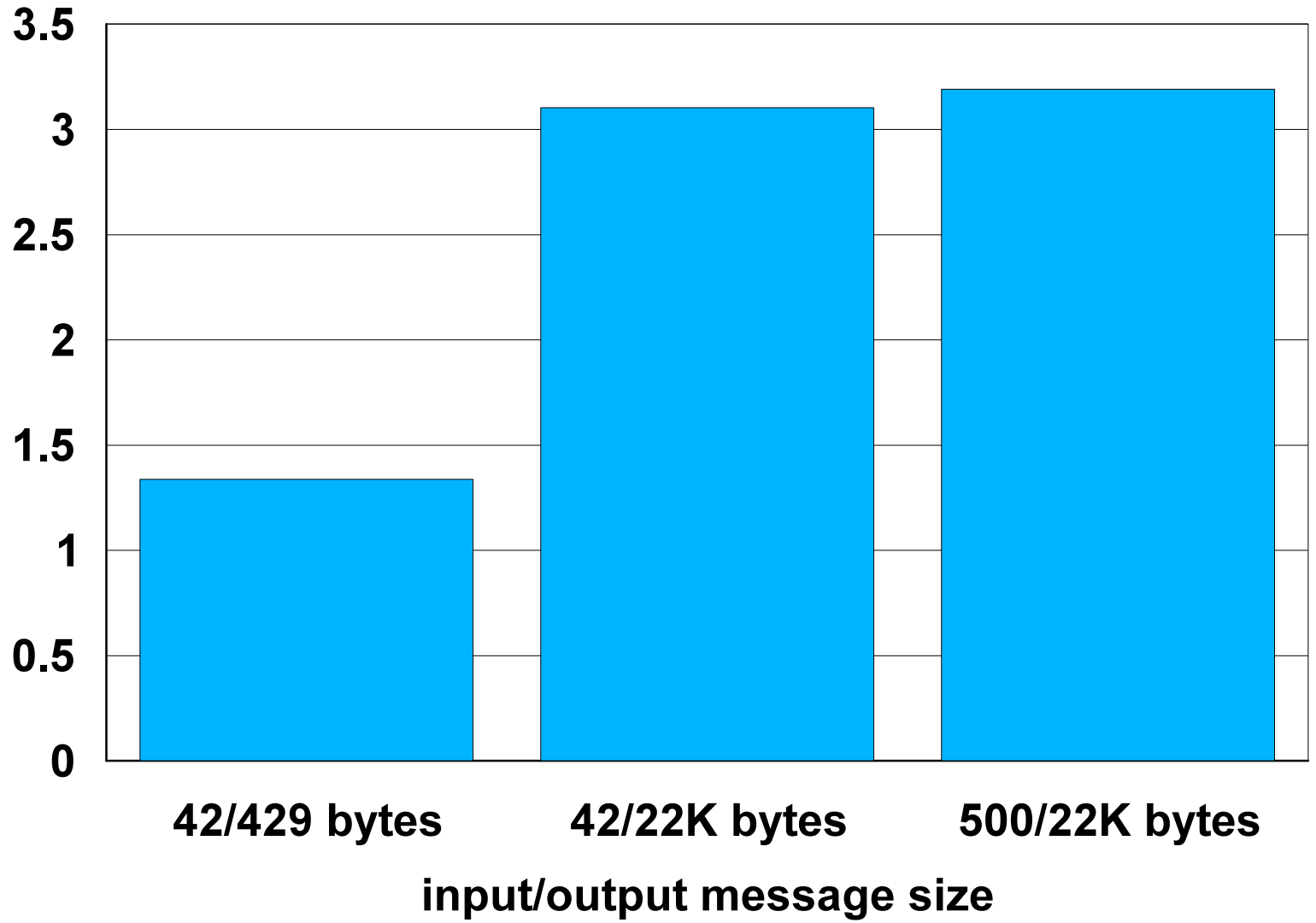
Throughput Rate

Small vs. Large Message



IMS Connector for Java
Pathlength

Small vs. Large Message



IMS Connector for Java

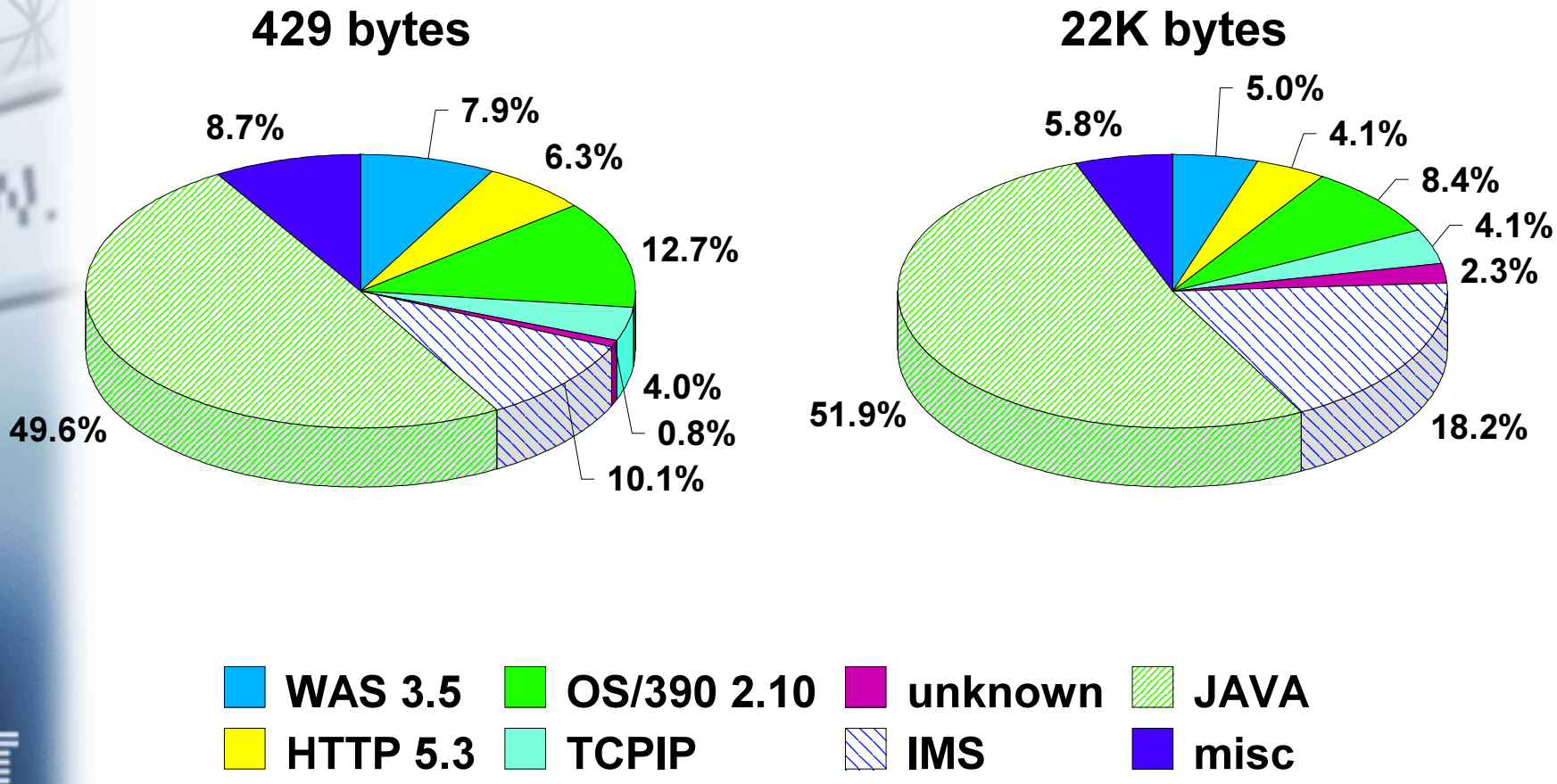
Throughput Drop Causes

- Java CPU use increase
  - CharToByteSingleByte.convert for input message
  - BytetoCharSingleByte.convert for output message
- I/O bandwidth limit for IMS Logging
- 100 Megabits Ethernet LAN bandwidth limit

IMS Connector for Java

429 vs. 22K bytes Output Message

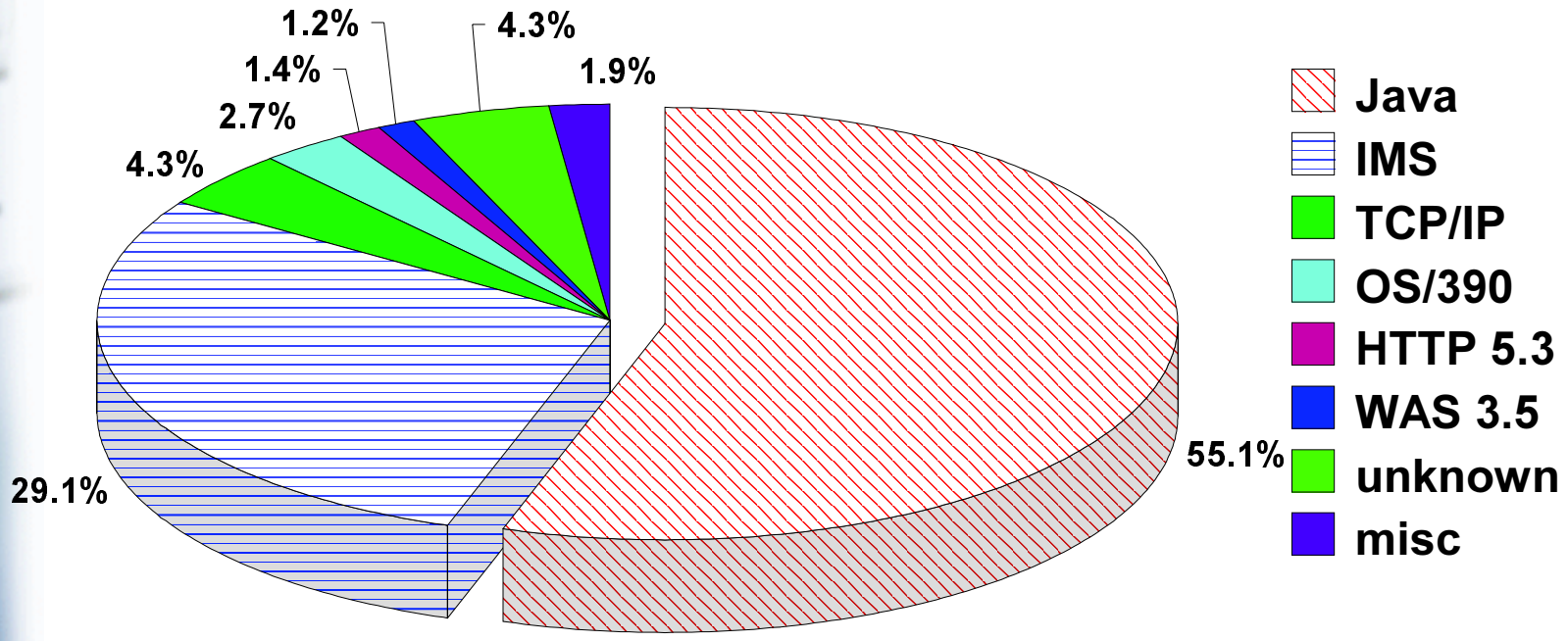
Product CPU Use Distribution



IMS Connector for Java

429 vs. 22K bytes Output Message

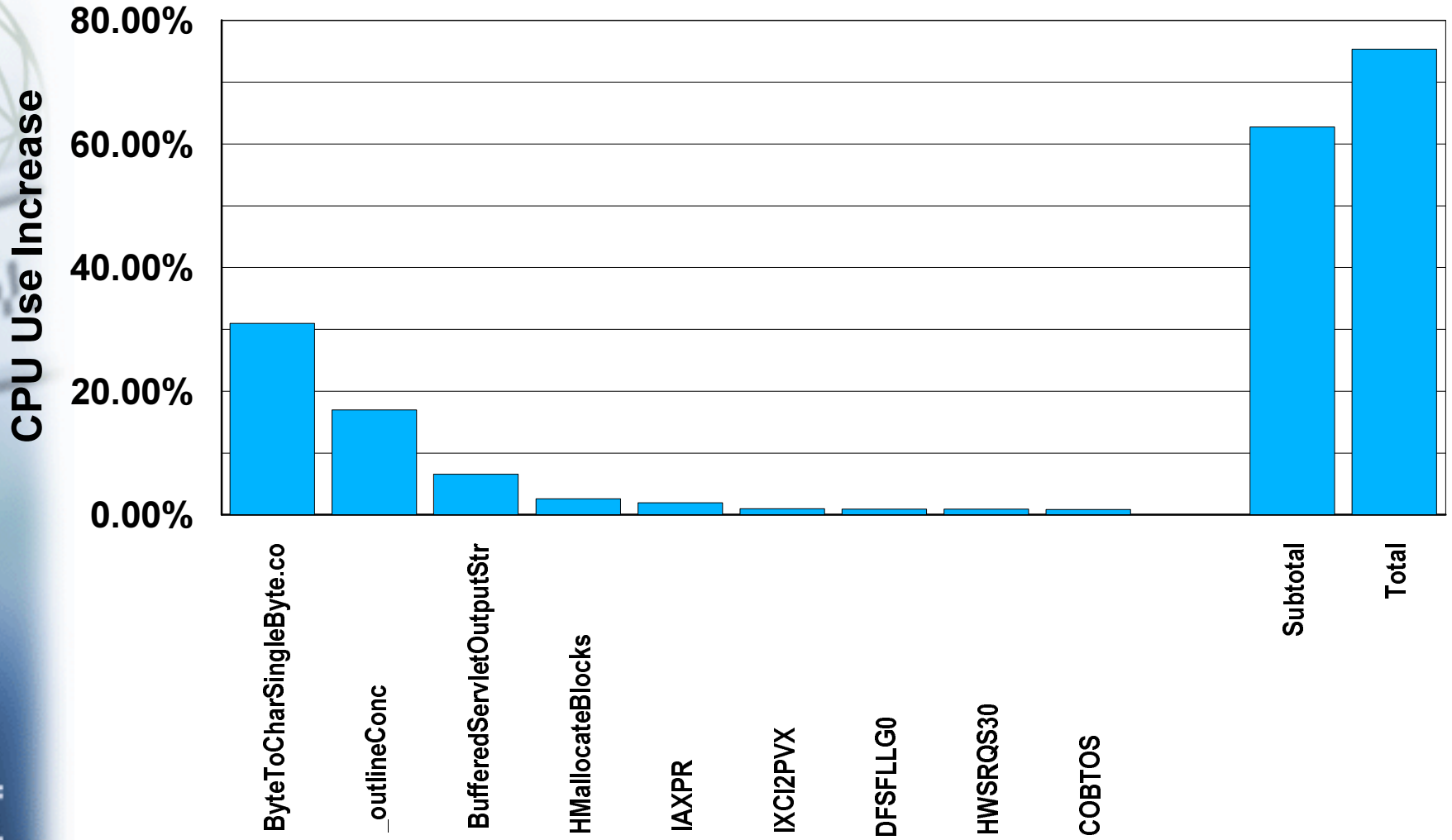
Product CPU Use Increase



IMS Connector for Java

429 vs. 22K bytes Output Message

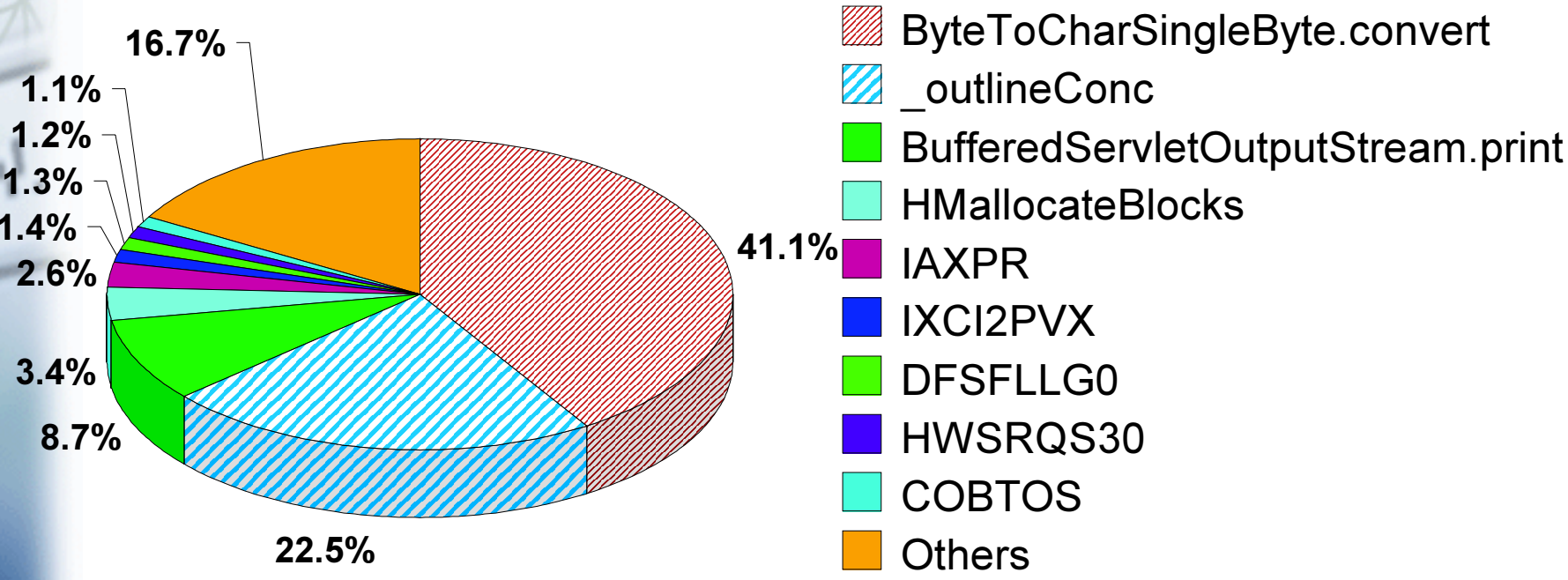
Top CPU Use Module/Method



IMS Connector for Java

429 vs. 22K bytes Output Message

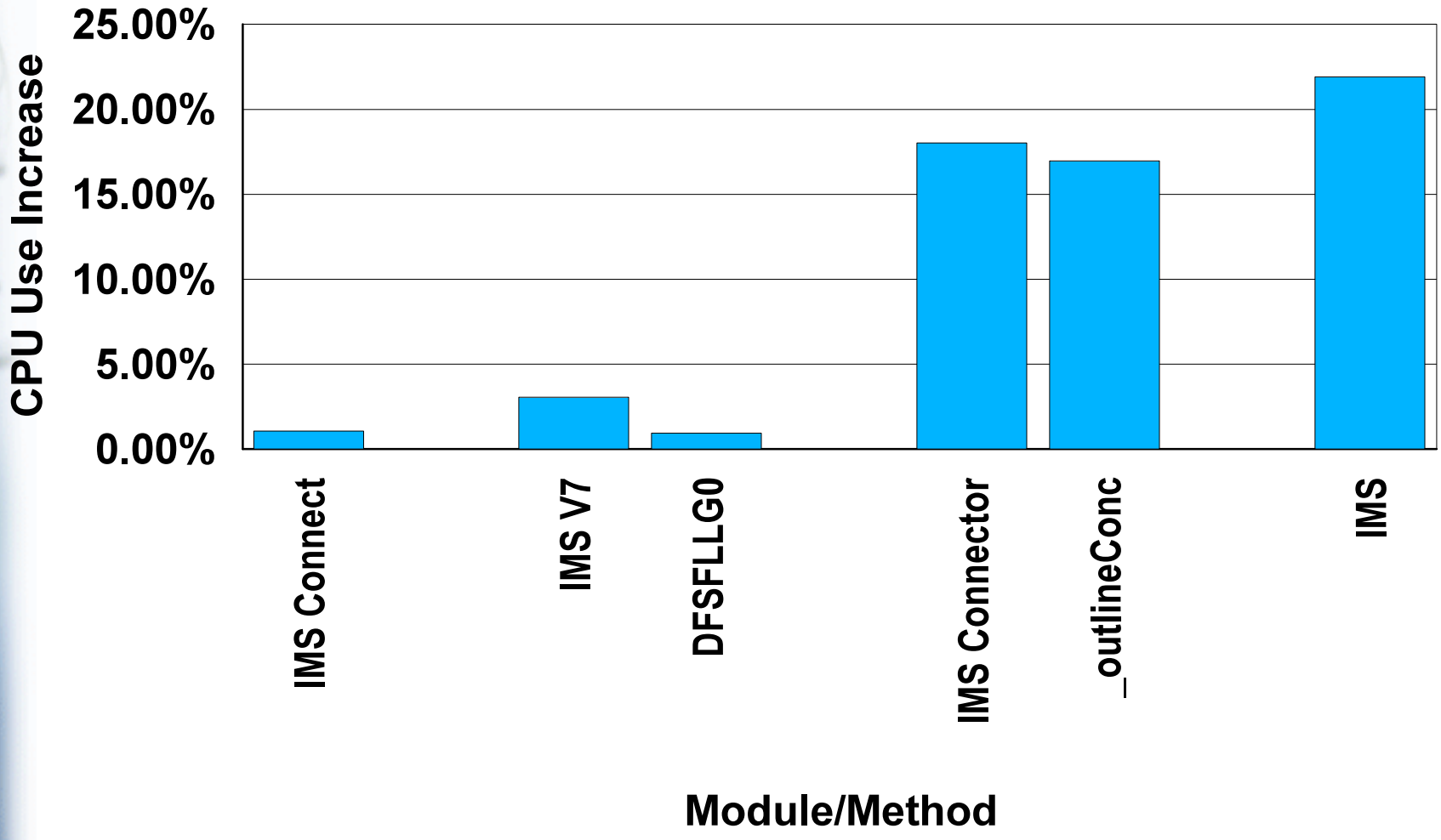
Top CPU Use Increase Modules/Methods



IMS Connector for Java

429 vs. 22K bytes Output Message

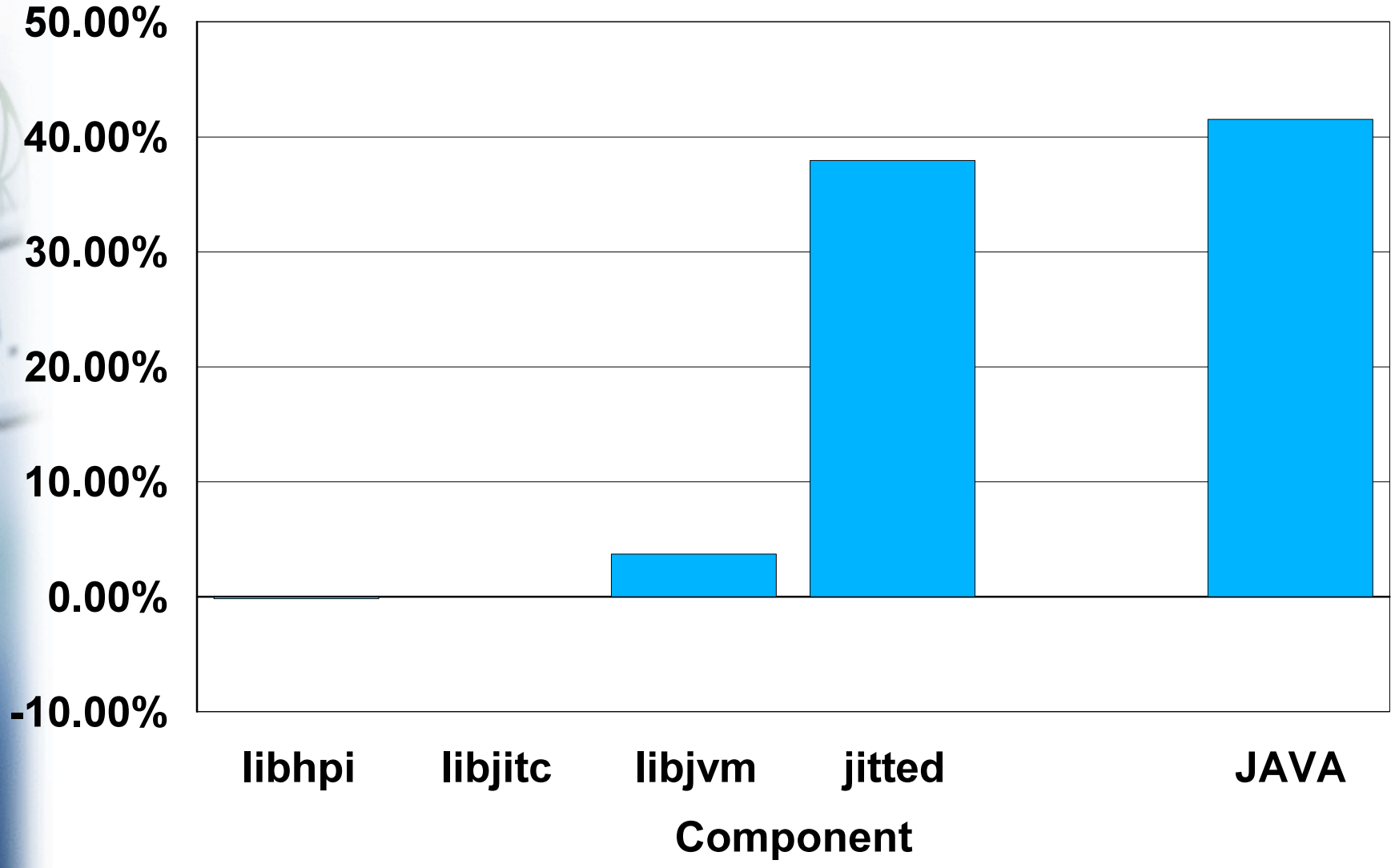
IMS Top CPU Use Increase



IMS Connector for Java

429 vs. 22K bytes Output Message

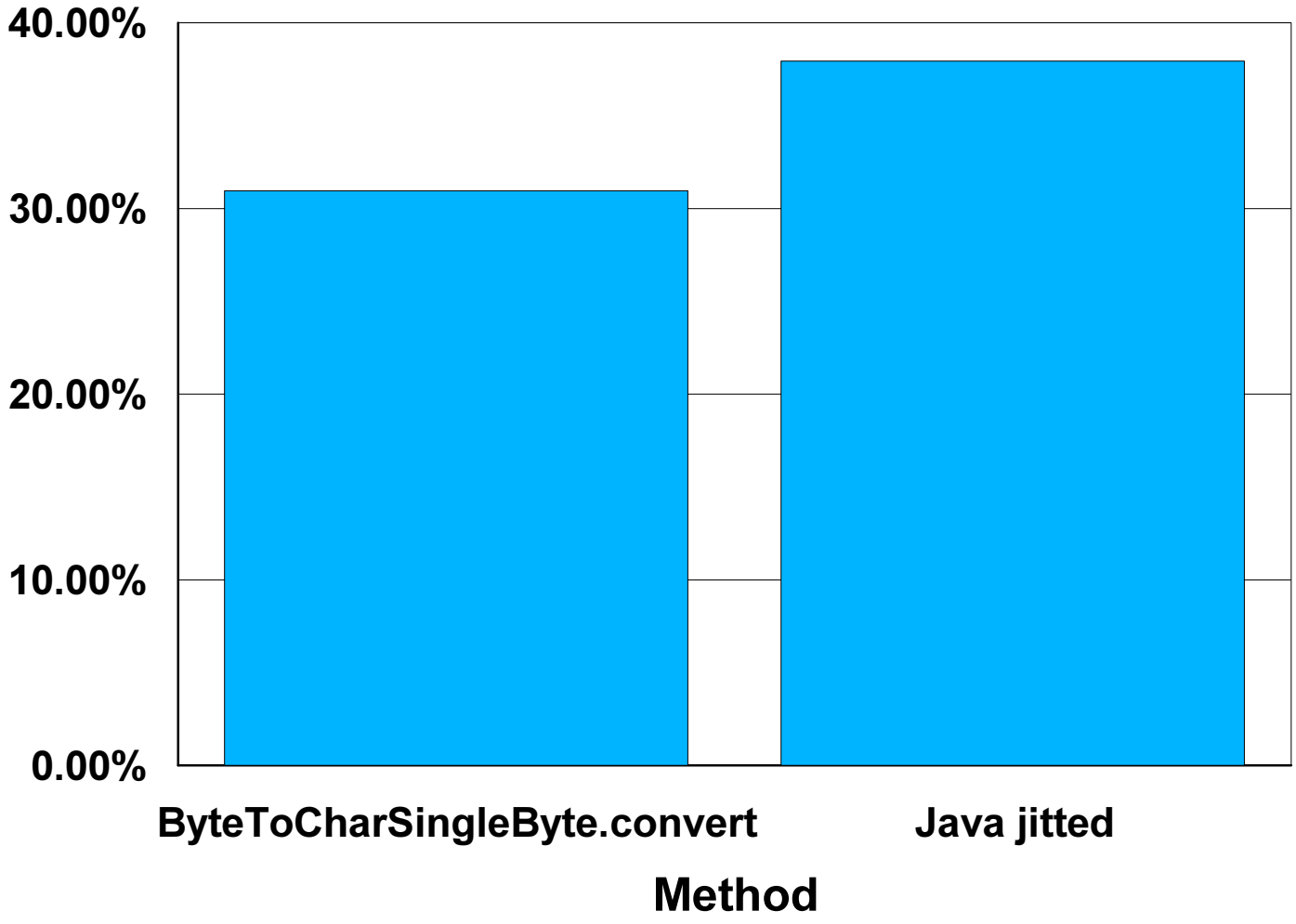
Java CPU Use Increase



IMS Connector for Java

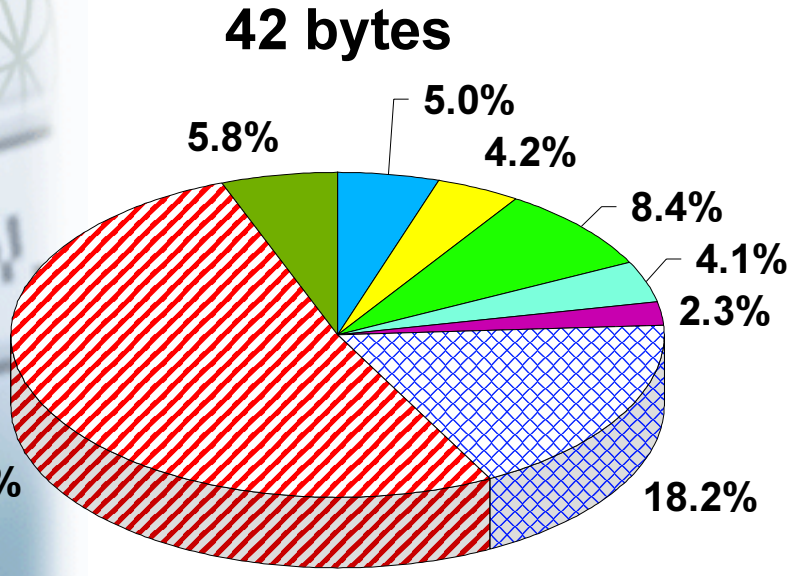
429 vs. 22K bytes Output Message

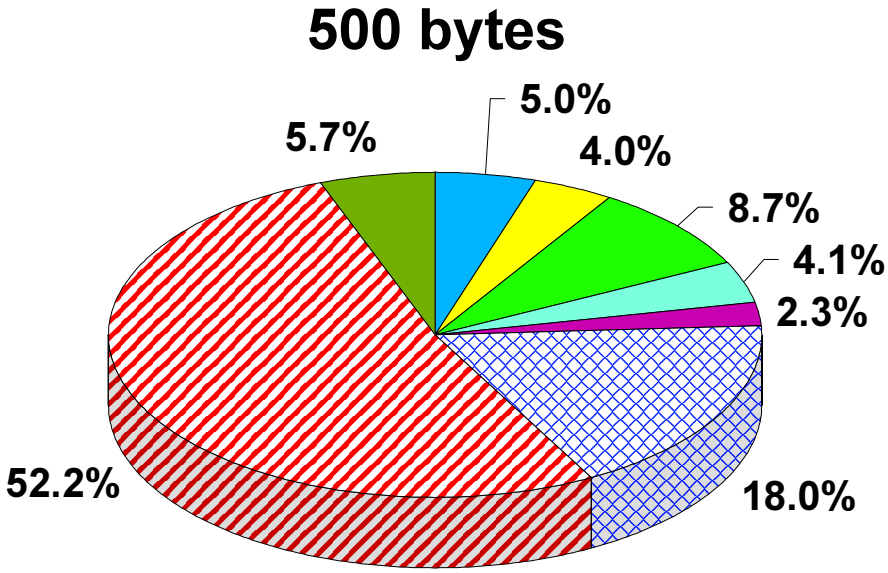
Java Top CPU Use Increase



IMS Connector for Java

42 vs. 500 bytes Input Message
22K Output Message
Product CPU Use Distribution





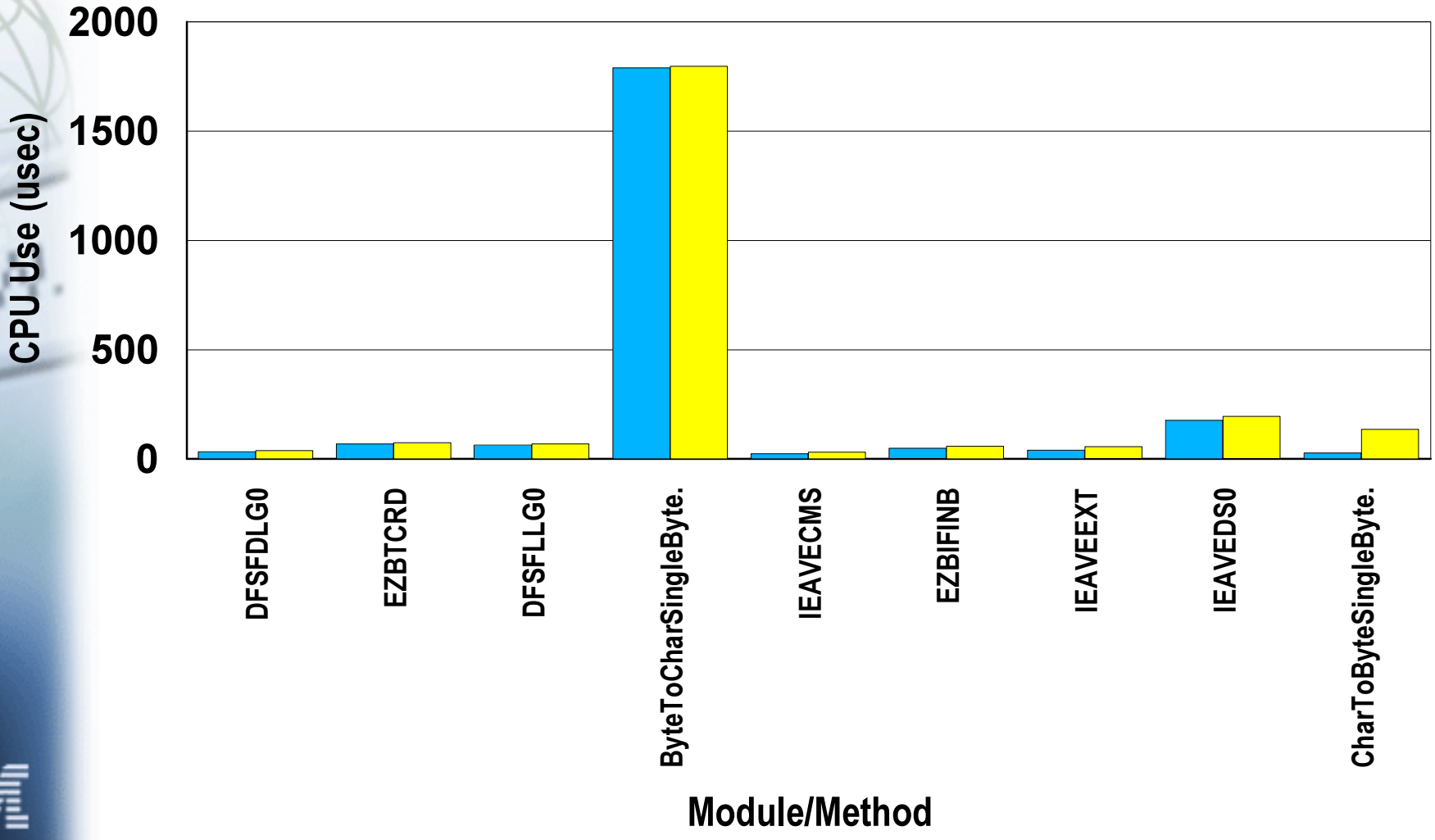
WAS 3.5 | OS/390 2.10 | unknown | JAVA
HTTP 5.3 | TCPIP | IMS | misc

IMS Connector for Java

42 vs. 500 bytes Input Message

22K Output Message

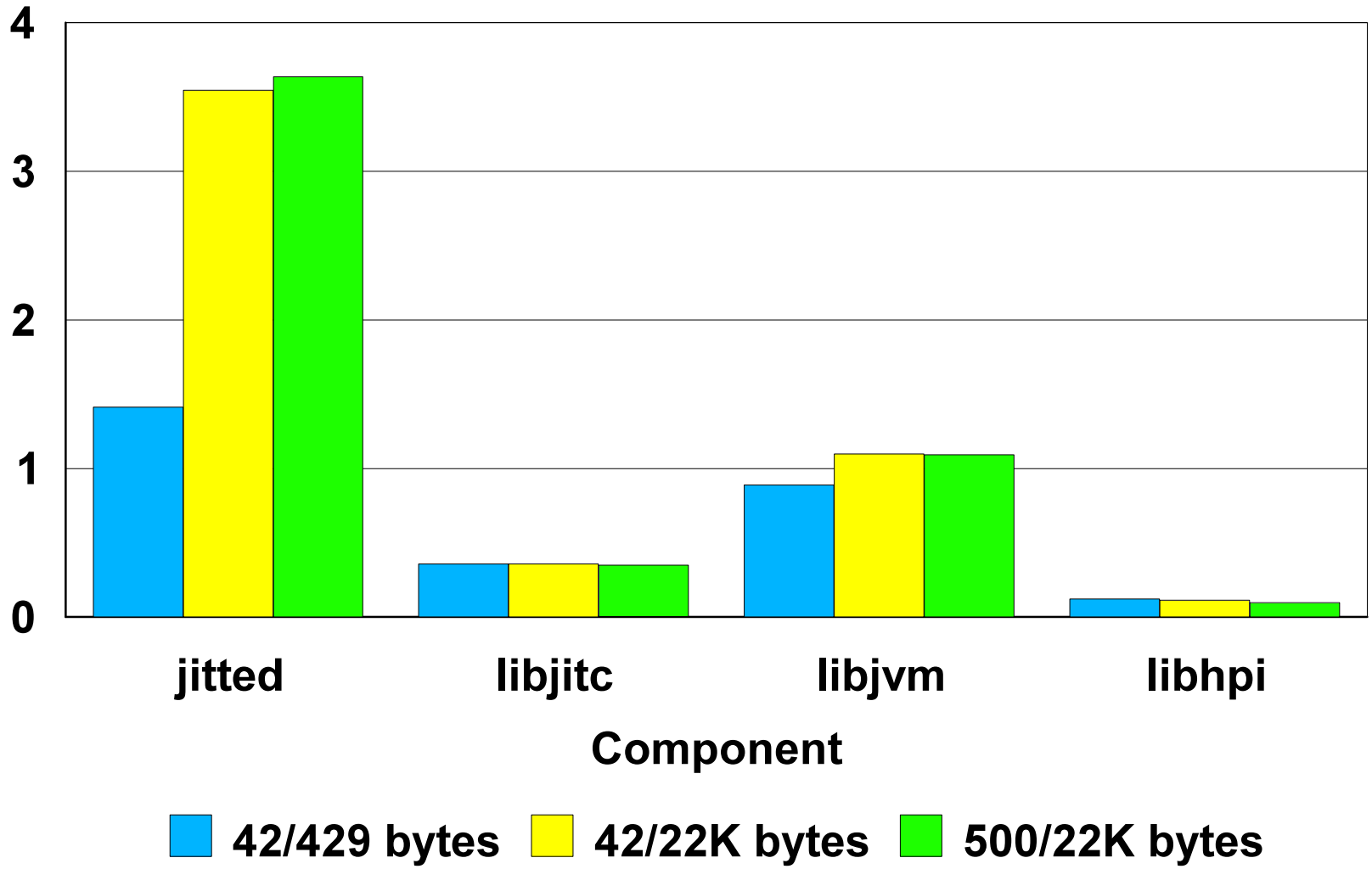
Top 10 CPU Use Modules/Methods



IMS Connector for Java

Small vs. Large Message

Java CPU Use Comparison



IMS Connector for Java

Large message IMS logging

- IMS logger wrote a lot of more data per message to DASD
  - 22K vs. 429 bytes output message
  - 500 vs. 42 bytes input message
- The average response time for OLDSs increased from 4 msec to 19 msec along with the message size increase on a 2105-E20

IMS Connector for Java

IMS logging experiments 42/22K message

42 bytes input 22K bytes output	2105-F20	2105-E20	9393-002
Throughput Rate	534.16	526.81	349.13
In Q time	0 msec	131 msec	389 msec
Transit time	18 msec	149 msec	407 msec
BLKSIZE	22K	22K	22K
OLDS I/O response time	9 msec	19 msec	28 msec

IMS Connector for Java

IMS logging experiments- 500/22K message

500 bytes input 22K bytes output	2105-F20(1)	2105-F20(2)	2105-E20
Throughput Rate	534.34	422.08	357.53
In Q time	10 msec	224 msec	378 msec
Transit time	29 msec	242 msec	397 msec
BLKSIZE	26K	22K	22K
OLDS I/O response time	18 msec	16 msec	19 msec

IMS Connector for Java

Performance Tips

- PUT OLDS/SLDS on a fast device
  - Improved I/O bandwidth, response time ===> improved throughput rate
  - 2105-F20 throughput result was limited by:
    - 100 Megabits Ethernet LAN
    - ESCON channel speed
- Select a good IMS logger BUFNO of OLDSDEF statement
  - avoid waiting for buffers

IMS Connector for Java

Performance Tips

- Issue /DISPLAY POOL IMS Command
  - Choose proper PSBW
    - IMS loader will wait for the PSBW storage to be freed up before a new dependent region can be scheduled
    - No error warning message
- Select good ECSA size in the IEASYSxx parm
  - PSBW and EMHB pools can be large
  - These pools are fixed storage

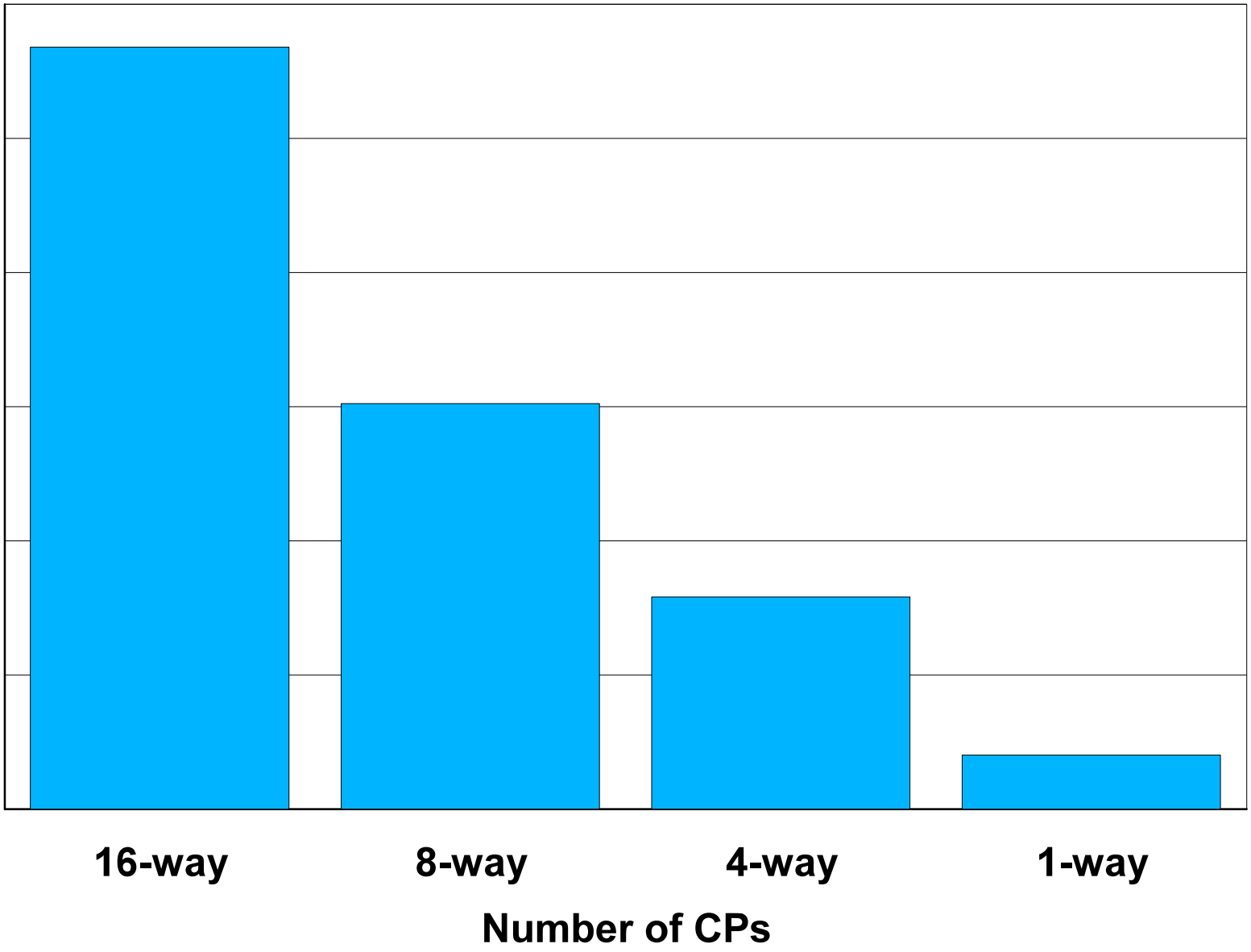
IMS Connector for Java

Over 1600 transactions per second throughput rate

Scalable

IMS Connector for Java

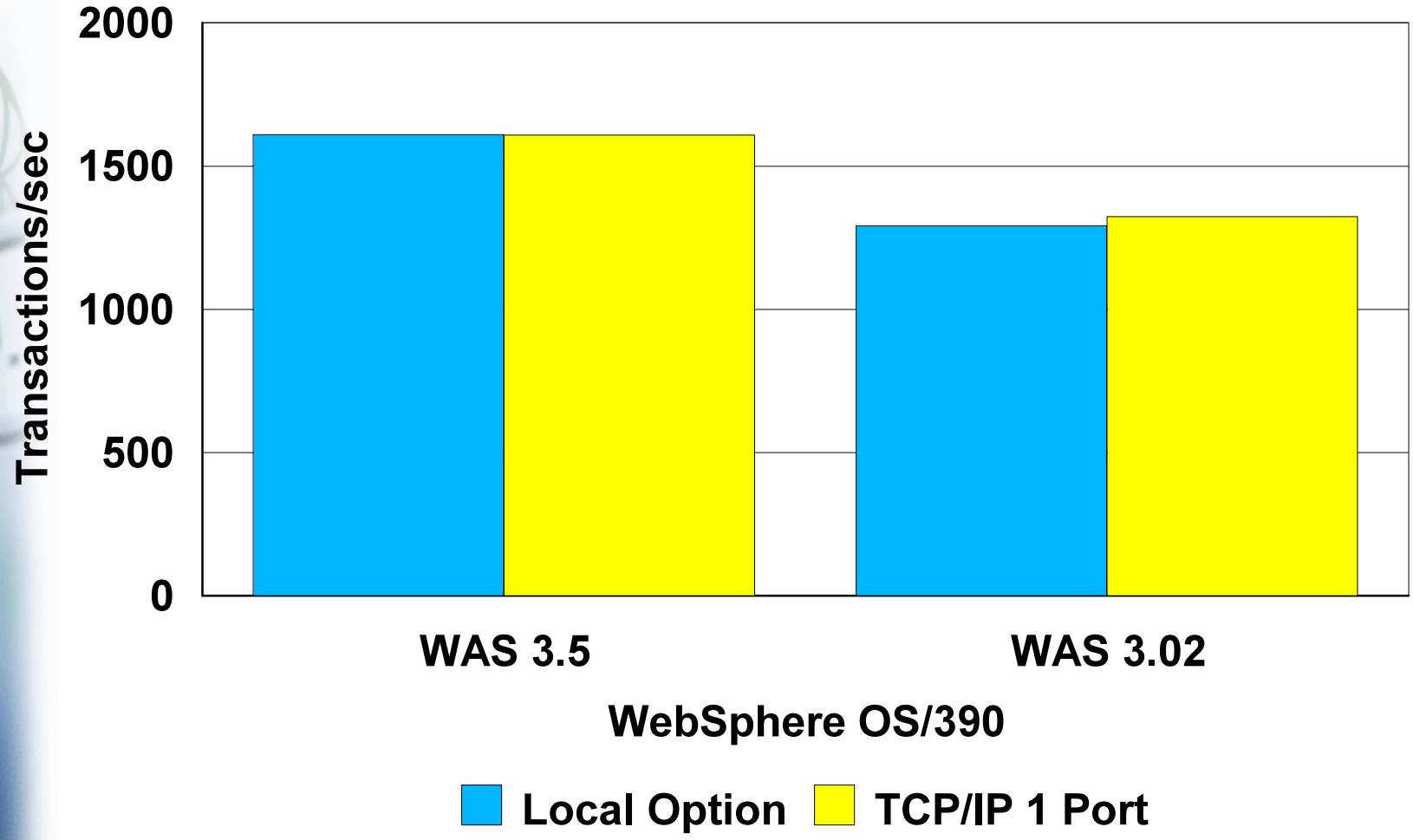
Local Option Scalability



IMS Connector for Java

Throughput Increased by over 20%

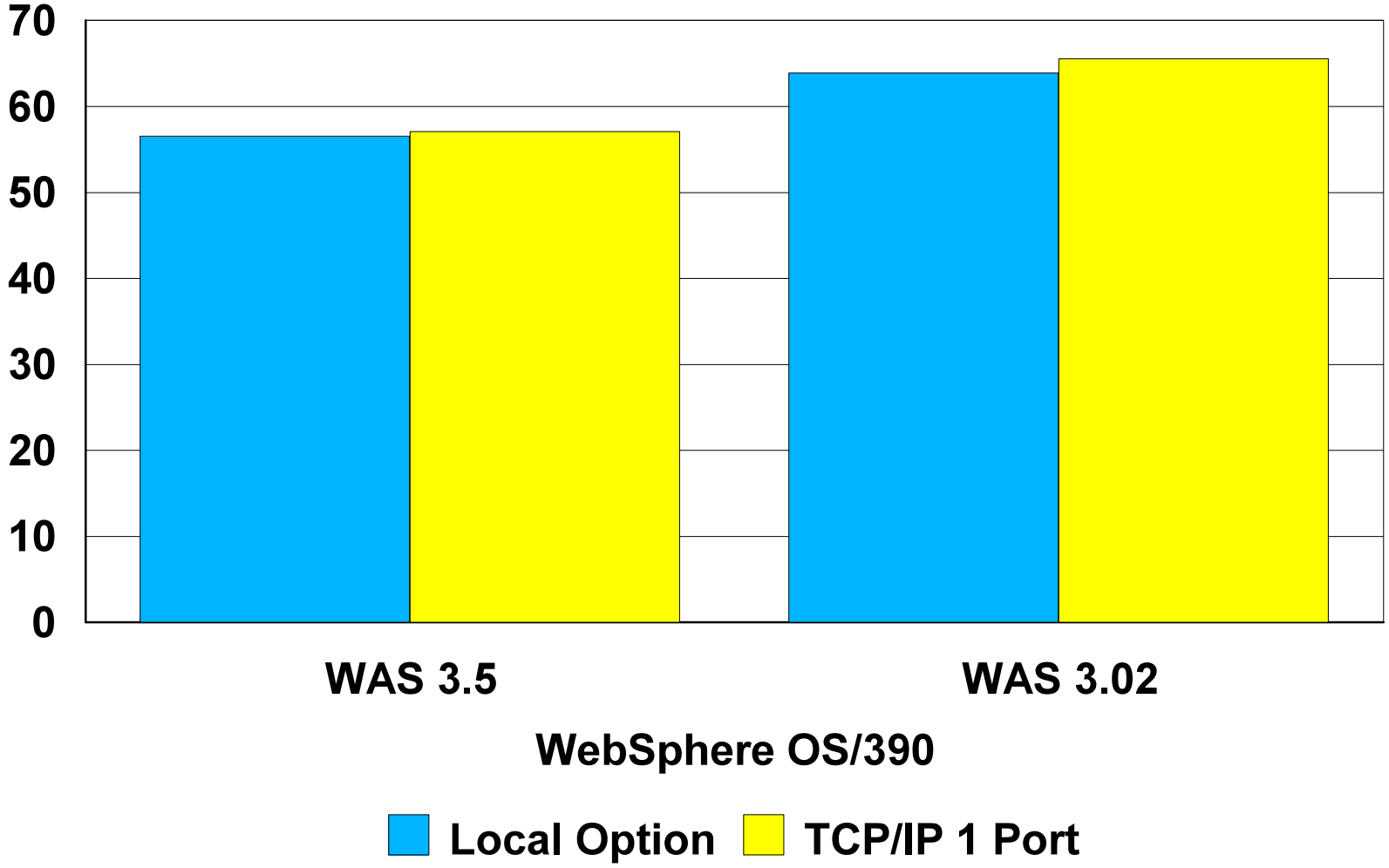
WAS 3.5 vs. WAS 3.02



IMS Connector for Java

CPU Use Decreased by 10%

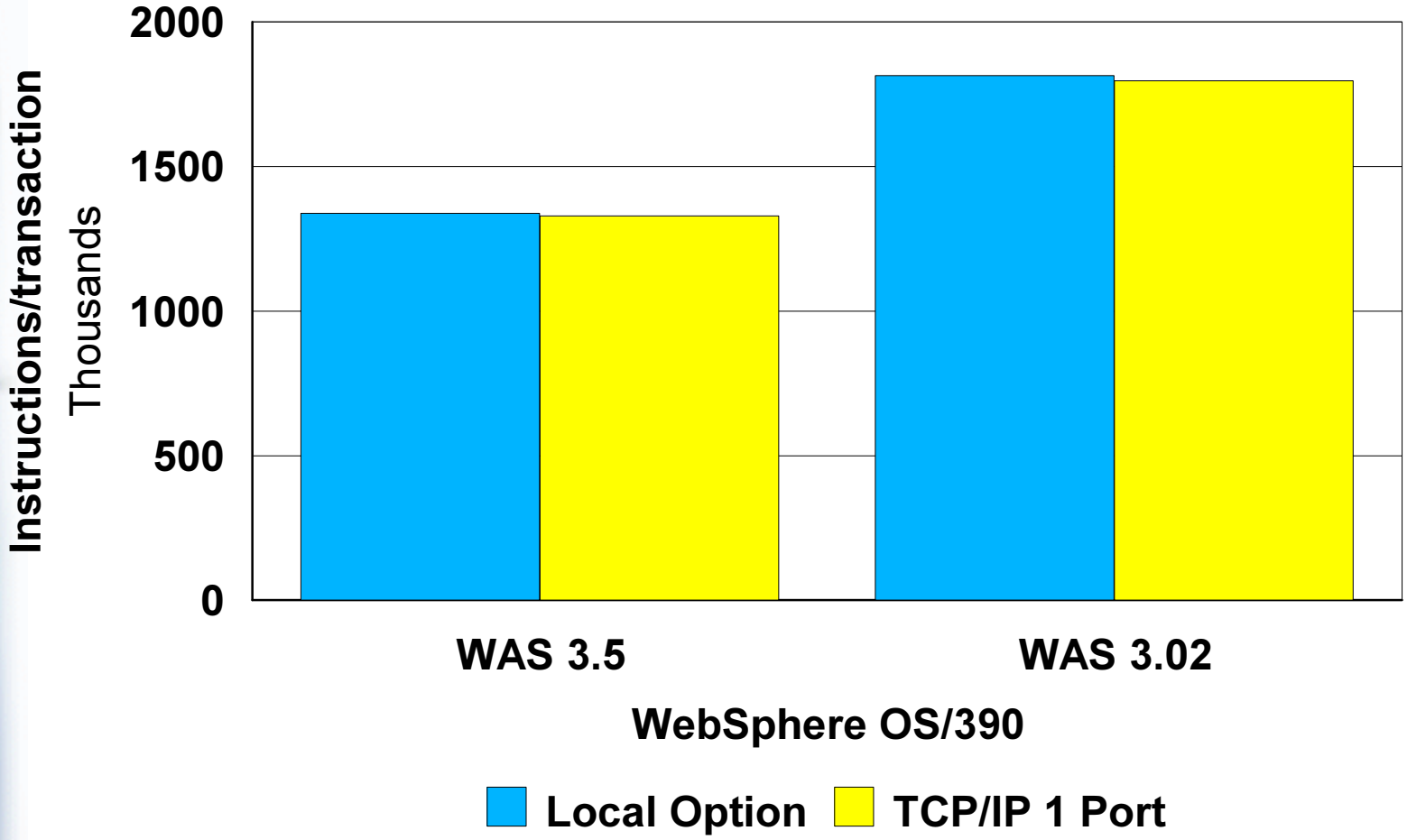
WAS 3.5 vs. WAS 3.02



IMS Connector for Java

Pathlength Reduced by over 20%

WAS 3.5 vs. WAS 3.02



IMS Connector for Java

Performance Tips

- WAS 3.5 was.conf
  - appserver.nodetach=true
    - avoid JVM attach/detach
- Jvmpropertiesfile
  - appserver.product.java.jvmconfig.jit=jitc
    - Our measurements showed 11 million instructions/second pathlength if jvmconfig.jit=on (default) was used
  - appserver.product.java.jvmconfig.mx and appserver.product.java.jvmconfig.ms
    - Choose the right Heap size
    - Make the Maximum Heap size the same as the minimum Heap size

IMS Connector for Java

Performance Tips

- Must be at SDK 1.3.0 SR 10 or SDK 1.3.1 level
  - A must have record conversion performance fix is available only in SDK 1.3.0 SR 10 and SDK 1.3.1
  - SDK 1.3.0 without the fix gave us **40%** more pathlength
- TCPIP
  - NODELAYACKS
  - SOMAXCONN
- Network capacity/performance
  - Ethernet LAN performance
  - 100 Megabits vs. 1 Gigabits Ethernet LAN

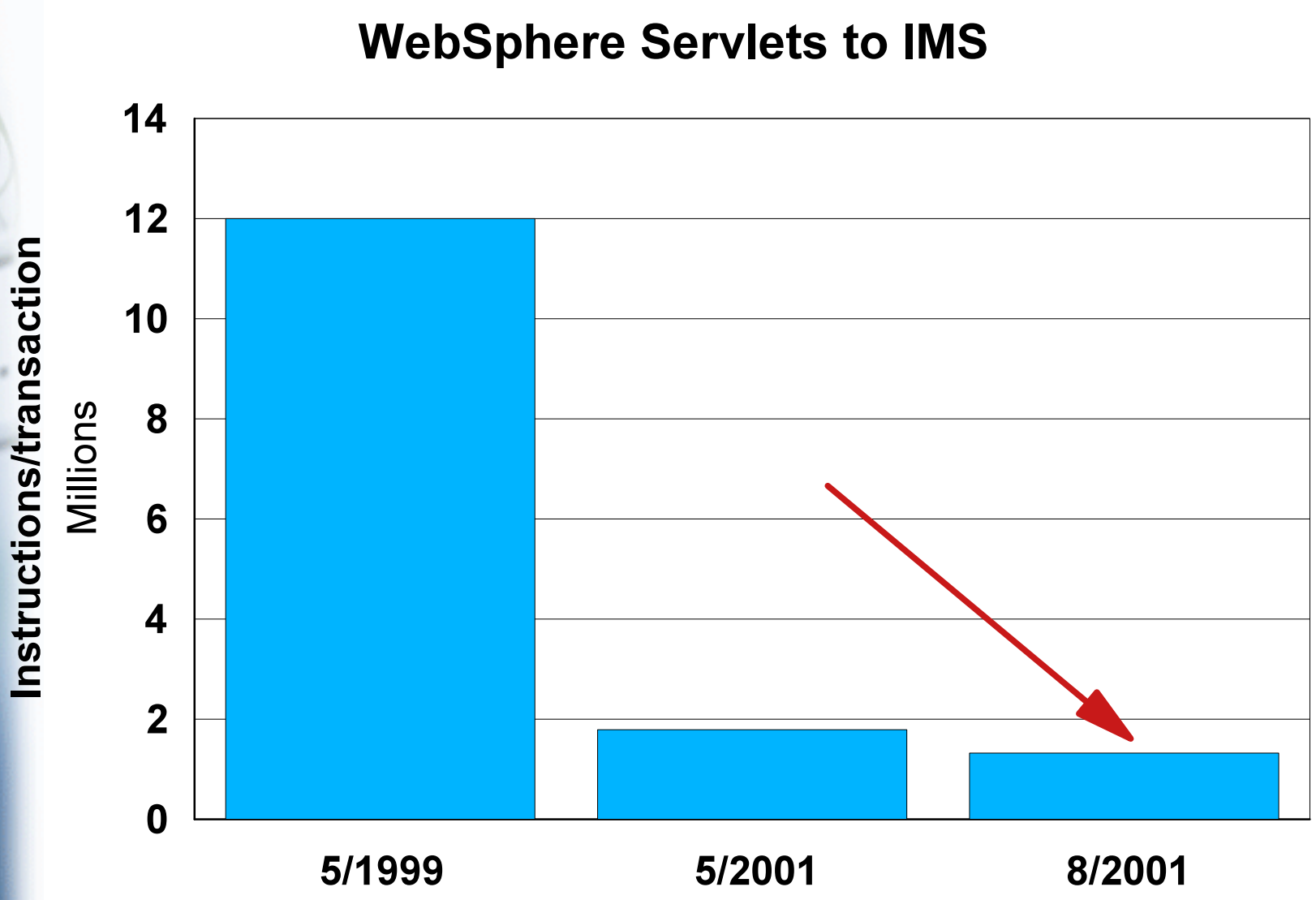
IMS Connector for Java

Performance Tips

- Upgrade to WebSphere 3.5
- Move to SDK 1.3.0 SR 10 or SDK 1.3.1

IMS Connector for Java

Performance Improvement Continues



IMS Connect

Almost 6000 transactions per second throughput rate

IMS Connect

- Test environment
  - IMS Fast Path Workload
  - A COBOL application had 12 Fast Path DEDB reads
  - The average input message was 42 bytes. The average output message was 429 bytes
  - IMS V7 and IMS Connect ran on an IBM 2064-116 single image 16-way server
  - TPNS was the driver to simulate the TCP/IP clients

IMS Connect

- A performance jump
  - IMS Connect for OS/390 V1 performance improved significantly when used with IMS V7
    - performance improvements: persistent socket, Unix System Services latch elimination, and an OTMA latch contention reduction
  - We reached a steady rate of almost **6000** transactions per second with a single IMS
    - The CPU usage was 52.3%
    - TPNS driver was on the same processor, 10-15% of the total CPU use

IMS Connect

Performance Tips

- IMS Connect
  - ECB=Y
  - Persistent Socket
- TCP/IP
  - SOMAXCONN
  - NODELAYACKS
- Network capacity/performance

Useful Information

- IMS Connect documentation
  - http://www-4.ibm.com/software/data/db2imstools/alpha.html
- IMS Connector for Java documentation, coding hints and tips, and many more ...
  - http://www-4.ibm.com/software/data/ims/jitoc.html
- Links with related information for Java on OS/390
  - http://www-1.ibm.com/servers/eserver/zseries/software/java/related.html

Conclusion

- IMS e-business offerings provide:
  - high throughput rate
  - truly industrial strength
- We will **continue** working with other development labs to improve the performance of our offerings
  - There is more to come

RETURN TO INDEX