# IPJVES Application Note 18:

# Performance profile

SV TEHS SIA

# IPJV-ES Application Note 18: Performance profile

**V 1.0**



Ruses 14-24 • LV1029 • Riga • Latvia
Phone: +371-9223895 • Fax: +371-7332773
Email: info@svtehs.com • Web: http://www.svtehs.com

Chapter 1

# Introduction

*IPJV-ES Development Board can be used in different applications.*

The IPJV-ES Development Board with embedded virtual machine for Java™ offers an Ethernet based connection to the Internet and numerous interface possibilities to other equipment, include serial RS-232 DTE interface, serializer module with UART, SPI, GPSI and 10BASE-T Ethernet support, 6-channel 10-bit A/D inputs, analog comparator and 16 I/O pins.

The IPJVM virtual machine for Java is a clean room implementation, that has been specially optimized to run on device with limited amount of internal memory and designed for Java™ 2 Platform, Micro Edition (J2ME™) Connected Device Configuration (CDC) Foundation Profile.

A complete development toolkit available for application development with IPJVM platform. The IPJVM platform provide system designers and software developers simple, flexible and cost-effective solution for embedded Internet application rapid development and prototyping. The platform is combination of Ubicom IP2022 Internet Processor and a Java programmable runtime environment.

The IPJV-ES Development Board based on Ubicom IP2022 Internet Processor, optimized for Internet-edge applications. It handles protocol processing in software instead of in hard-wired logic, making the whole solution more adaptable to evolving standards and allow designer to use the same solution across a wide variety of internet-edge products simply by changing the software, thereby significantly reducing nonrecurring engineering (NRE) costs.

Typical IPJV-ES applications include Includes HTTP/FTP/SMTP/SNMP/Telnet servers, PPP support on embedded UARTs, encryption, security and authentication tools, reporting and alarming via e-mail, remote monitoring, control, management and maintenance.

## Updates

New versions of the IPJV-ES software and applications can be obtained from the manufacturer's web site at:
`http://www.svtehs.com/ipjv.htm`

Chapter 2

# Performance profile

*How to measure application performance.*

This application demonstrate the procedure used to time the performance of some code snippets in different applications. It also provides some insight into issues that should be considered to accurately profile portions of your application

The `System.currentTimeMillis()` method, described in the AN16, can be used for timing operations that execute over a period of a few milliseconds to several minutes or hours with one ms accuracy. To keep the overhead of the profiling code minimal, the timing measurements are taking in-line by capturing two times – before and after profiling code portion.

```
Class Someclass{
. . .
  Void SomeMethod{
  . . .
     begin_time = System.currentTimeMillis();
     // Profiling code portion
     . . .
     end_time = System.currentTimeMillis();
     total_time = end_time - begin_time;
     out.println("total time: "+String.valueOf(total_time)+"ms");
  }
}
```

As the example of the profiling code this apllication use Dhrystone benchmark. Developed in 1984 by R.P. Wecker, Dhrystone is a benchmark program that tests a system's integer performance. The program is CPU bound, performing no I/O functions or operating system

calls. Dhrystones per second is the metric used to measure the number of times the program can run in a second.

The original Dhrystone benchmark is still widely used to measure CPU performance in industry under various versions or variants. The benchmark is designed to contain a representative sample of types of operations, mostly numerical, used by applications. Unfortunately this does not always represent a true real-life performance, but is useful to compare the speed of various CPUs.

The result is determined by measuring the time it takes to perform some sequences of instructions. Due to various changes, the result is not directly comparable with other Dhrystone benchmarks. However the MIPS (Million Instructions Per Second) should be the same for the same system (+5-10% variation) between benchmarks.

# Table of Contents