

# Úskalí práce s IPv6 v ekosystému Javy

Adam Kalisz

[adam.kalisz@notnullmakers.com](mailto:adam.kalisz@notnullmakers.com)



NOT NULL  
Makers



Úskalí práce s IPv6 v ekosystému Javy

# Miliardy zařízení s JVM, bez IPv6

- Jak funguje Java s IPv4, IPv6 a oběma dohromady?
- Jak funguje překlad jmen pomocí DNS?
- Jak efektivní je práce s IP adresami pomocí `java.net`?

# Prostředí

- GNU/ Linux – Debian 13
- OpenJDK 21
- dual-stack

# Dual-Stack Ready

`-Djava.net.preferIPv4Stack=false`

- Pokud je IPv6 dostupná, bude použit nativní IPv6 socket. `::ffff:127.0.0.1` IPv4-mapovaná IPv6 adresa.
- Pokud se nastaví při startu na `true`, nebude aplikace komunikovat po IPv6.

# Dual-Stack Ready pokračování

`-Djava.net.preferIPv6Addresses=false`

- Pokud je IPv6 dostupná, bude použit nativní IPv6 socket avšak stále IPv4-mapované IPv6 adresy. `::ffff:127.0.0.1`
- Pokud se nastaví při startu na `true`, bude použita IPv6 varianta `::1` a budou preferovány IPv6 adresy.

# Překvapující chování

- Přepínat můžete nastavení preference IPv4 a IPv6 i za chodu, ale kontroluje se jen při startu. 🙄
- Standardně zůstávají pozitivní překlady DNS jmen v keši do zastavení JVM. Negativní překlady zůstávají 10 sekund.
- Java neumí poslat ICMP echo v metodě `InetAddress.isReachable` bez dalšího oprávnění protože vyžaduje na Linuxu zbytečně `SOCK_RAW`.

# Překvapující chování pokračování

- Nastavení IPv4/ IPv6 ovlivňuje překlad jmen. `system` zachová pořadí z resolveru.
- Pokud nemá žádné síťové rozhraní IPv6 adresu v `/proc/net/if_inet6`, systém bude předpokládat, že není IPv6 aktivní.
- `setcap cap_net_raw=eip `realpath /usr/bin/java`` pro schopnost ICMP echo v Javě bez plných oprávnění root.



# Obecné poznámky

- Kontrola `/etc/hosts` a `/etc/resolv.conf` na správné nastavení pro IPv4 a IPv6
- Kešování překladu jmen lze upravit ⚡ :  
`java.security.Security`  
`.setProperty("networkaddress.cache.ttl","0")` což by cache pro testování vyplo.
- `System.setProperty("sun.net.inetaddr.ttl","0")`

# Obecné poznámky pokračování

`java.net.InetAddress.getAllByName("example.com")`

- Debugging časováním dotazů, `tcpdump -nni any port 53`, `ss -lnpt`, REPL, jshell (OpenJDK 9+)
- Aplikace jako Elasticsearch, Websphere můžou mít jiné nastavení keší v jiných souborech.
- `$JAVA_HOME/conf/security/java.security` (OpenJDK 11+)
- `networkaddress.cache.stale.ttl` (OpenJDK 21+)

# Happy Eyeballs v1, v2, v3

# Paměťová (ne)efektivita IPv4 v Javě

- Pro `Inet4Address` používá Java dohromady 56 B místo 32 b (4 B), tedy skoro 15x režie
- `Inet4Address` objekt zabírá 24 B
- `InetAddressHolder` 32 B a může obsahovat hostname a typ adresy (IPv4 = 1, IPv6 = 2)
- Možnost snížit na 28 B celkem

# Paměťová (ne)efektivita IPv6 v Javě

- Pro `Inet6Address` používá Java dohromady 120 B místo 128 b (16 B), tedy skoro 7,5x režie
- `Inet6Address` obsahuje `InetAddressHolder` zděděný od `InetAddress` a přidává `Inet6AddressHolder`, který navíc přidává rozsah platnosti a pole bytů pro adresu samotnou.
- Možnost inspirace implementací `java.util.UUID`, kdy se využívají 2x long

# Rozložení objektu

Inet@25618e91d footprint:

Table is sorted by "SUM".

Printing first 30 lines. Use -DprintFirst=# to override.

COUNT	AVG	SUM	DESCRIPTION
2	32	64	java.net.InetAddress.InetAddressHolder
1	32	32	byte[]
1	32	32	java.net.Inet6Address.Inet6AddressHolder
1	24	24	Inet
1	24	24	java.net.Inet4Address
1	24	24	java.net.Inet6Address
7	168	200	<total>

# Spotřeba paměti u Inet4Address

Name	Live Bytes	Live Objects
java.net.InetAddress\$InetAddressHolder	1,600,001,120 B (52.9%)	50,000,035 (49.9%)
java.net.Inet4Address	1,200,000,432 B (39.6%)	50,000,018 (49.9%)
java.net.Inet4Address[]	200,000,016 B (6.6%)	1 (0%)

$$32 \text{ B} + 24 \text{ B} = 56 \text{ B}$$

# Spotřeba paměti u Inet6Address

Name	Live Bytes	Live Objects
byte[]	1,604,062,224 B (25.8%)	50,052,790 (25%)
java.net.InetAddress\$InetAddressHolder	1,600,000,672 B (25.7%)	50,000,021 (25%)
java.net.Inet6Address\$Inet6AddressHolder	1,600,000,288 B (25.7%)	50,000,009 (25%)
java.net.Inet6Address	1,200,000,216 B (19.3%)	50,000,009 (25%)
java.net.Inet6Address[]	200,000,016 B (3.2%)	1 (0%)

$$3 \times 32 \text{ B} + 24 \text{ B} = 120 \text{ B}$$

```

Clojure 1.12.0
(ns user
  (:import (java.net InetAddress Inet6Address Inet4Address)))

(declare from-inet-address)

(defrecord inet4-address [^int address]

  Object
  (toString [inet4]
    (.toString (from-inet-address inet4))))

;; Should also do reader literals
(defrecord inet6-address [^long msb ^long lsb]

  Object
  (toString [inet6]
    (.toString (from-inet-address inet6))))

(defn bytes->num
  [ba index upper-bound]
  (loop [i index acc 0]
    (if (< i upper-bound)
      (recur (inc i) (+ (bit-shift-left acc 8) (bit-and (aget ba i) 0xff)))
      acc)))

(defn num->bytes
  [ba num index upper-bound]
  (loop [i index shift (* (- (dec upper-bound) index) 8)] ; 56 for long ; 24 for int presumably
    (if (< i upper-bound)
      (do (aset ba i (unchecked-byte (bit-and (bit-shift-right num shift) 0xff)))
          (recur (inc i) (- shift 8)))
      ba)))

(defn ->inet-address
  [^String inet-str]
  (let [inet (InetAddress/getByName inet-str)]
    (cond (instance? Inet6Address inet)
      (let [^bytes inet6-bytes (.getAddress ^Inet6Address inet)]
        (inet6-address. (bytes->num inet6-bytes 0 8)
          (bytes->num inet6-bytes 8 16)))
      (instance? Inet4Address inet)
      (let [^bytes inet4-bytes (.getAddress ^Inet4Address inet)]
        (inet4-address. (unchecked-int (bytes->num inet4-bytes 0 4)))))))

(defn from-inet-address
  [inet]
  (cond (instance? inet6-address inet)
    (-> ^bytes (make-array Byte/TYPE 16)
      (num->bytes (.msb inet) 0 8) ; get the first half of the address as long
      (num->bytes (.lsb inet) 8 16) ; get the second half of the address as long
      (Inet6Address/getByAddress))
    (instance? inet4-address inet)
    (-> ^bytes (make-array Byte/TYPE 4)
      (num->bytes (.address inet) 0 4)
      (Inet4Address/getByAddress))))

=> nil

```

# Prototyp v Clojure



# Zdroje

<https://docs.oracle.com/en/java/javase/21/core/java-networking.html#GUID-540A1A77-3890-4303-A1EA-B51CCE58463B>

<https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/jvm-ttl-dns.html>

<https://github.com/netty/netty/issues/13400>

<https://bugs.openjdk.org/browse/JDK-8170568>

<https://bugs.openjdk.org/browse/JDK-8257235>

<https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/net/InetAddress.html>

<https://www.elastic.co/docs/deploy-manage/deploy/self-managed/networkaddress-cache-ttl>

<https://github.com/openjdk/jdk/blob/master/src/java.base/unix/native/libnet/Inet4AddressImpl.c>

[https://github.com/openjdk/jdk/blob/master/src/java.base/unix/native/libnet/net\\_util\\_md.c](https://github.com/openjdk/jdk/blob/master/src/java.base/unix/native/libnet/net_util_md.c)

<https://github.com/openjdk/jdk/blob/master/src/java.base/share/classes/java/net/InetAddress.java>

<https://github.com/openjdk/jdk/blob/master/src/java.base/share/classes/java/net/Inet4Address.java>

<https://github.com/openjdk/jdk/blob/master/src/java.base/share/classes/java/net/Inet6Address.java>

<https://github.com/openjdk/jdk/blob/master/src/java.base/share/classes/java/util/UUID.java>

`man capabilities(7)`

# Děkuji – Thank You 🎉

Pro pozdější dotazy a zpětnou vazbu:  
[adam.kalisz@notnullmakers.com](mailto:adam.kalisz@notnullmakers.com)

