

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Katedra informatiky a výpočetní techniky

KIV/VSP

Testování výkonosti webového REST API

Autor: Antonín NEUMANN, A14N0139P
E-mail: neumann@students.zcu.cz
Akademický rok: 2015/2016

Zadání

Programové prostředky

- Javascript (Coffeescript), Node.js
- PHP, Apache

Definice

- bude navrženo jedno REST API, které bude společné všem implementacím
- všechny endpointy (bude-li to možné) budou stejné (tj. včetně parametrů)
- autentizace pro testovací účely nebude žádná
- všechny implementace budou využívat pouze předem vytvořené odpovědi nebo data uložená přímo v paměti (tj. žádná databáze nebude potřeba)
- všechny objekty budou používat knihovny (preferovaný) nebo ruční převod z datové třídy do formátu JSON
- každá implementace serveru dostane k dispozici 512MB paměti RAM
 - **PHP:** `ini_set('memory_limit','512M');`
 - **NodeJS:** `--max-old-space-size=512`

Popis testování

- pro všechny implementace API bude vytvořena jedna aplikace (klient), která bude vykonávat dotazy (například jednoduchá JS aplikace) a simulovat tím činnost mnoha uživatelů
- tato aplikace bude spuštěna na minimálně jednom počítači, jiném než na kterém bude spuštěn server
- server bude představovat notebook: Lenovo ThinkPad E330, Core i5-3210M 2.5GHz, 8GB RAM, Windows 10 64bit, RealTek Semiconductor RTL8168/8111 PCI-E Gigabit Ethernet (více informací o HW konfiguraci v příloze v systémovém logu z programu HWINFO64)

Parametry SW prostředí

- **PHP:** Server Apache ve verzi 2.4.7 (32bit) a PHP verze 5.5.8
- **NodeJs:** Verze Nodejs 4.2.3 LTS, verze package managera NPM 2.14.0 a verze Coffeescriptu 1.9.3

Měření

- čas potřebný k implementaci v jednotlivých programovacích jazycích (speciální neopakovatelná veličina, velmi subjektivní)
- rychlost odpovědi na různé dotazy (od základních po složitější)
- rychlost odpovědí na základě stoupajícího počtu požadavků (simulace připojení více uživatelů)
- kolik dotazů je potřeba aby API přestalo reagovat (zahlcení, pád celého apliakce)

Návrh API

Celé API je popsané pomocí online nástroje Apiary.io s využitím syntaxe API Blueprint (www.apiblueprint.org), dokumentace celého API je dostupná online na: <http://docs.vspapibenchmark.apiary.io>

Zjednodušeně jde o URL cesty:

- pro testování requestů
GET, POST, PUT, DELETE /test
- pro uživatele
GET, POST, PUT /users
GET, DELETE /users/:userId
- pro finanční toky (příjmy a výdaje)
GET, POST, PUT /money
GET, DELETE /money/:moneyId

Výsledky měření

Měření času

Vzhledem k tomu, že jednotlivé requesty jsou posílány asynchronně, bylo měření času docela problematické. Pokud budeme měřit čas od odeslání požadavku do jeho přijetí získáme tím docela přesný výsledek, který ale příliš neodpovídá realitě, jelikož za dobu návratu jednoho requestu se reálně stihne provést a vrátit requestů několik.

Další možností je měřit celkovou dobu, kterou klient čekal na vyřízení všech svých requestů.

NodeJS

Jako první jsem začal s implementací API v Nodejs (www.nodejs.org) pomocí Coffeescriptu (www.coffeescript.org). Nodejs je implementace Javascriptového enginu V8, používaného v Google Chrome, která je uzpůsobená k běhu na serveru. Coffeescript je jazyk, který je přímo transpilován do Javascriptu.

Dále jsem si zvolil velmi rozšířený framework Express (www.expressjs.com), který slouží jako základ pro HTTP server.

Nejvíce času zabralo řešení tzv. CORS (Cross-origin resource sharing), který umožňuje Javascriptu vykonávat requesty i na jinou doménu. Řešení CORS zabralo v souhrnu asi 6 hodin, které nebudu do základního času implementace serveru započítávat.

Čas pro implementaci

Výsledný čas pro implementaci, jsem rozdělil na získání základního přehledu pro tvorbu API (tj. zhruba přečtení jednoho až dvou tutoriálů a letmý pohled do dokumentace výše zmíněných nástrojů). S většinou z těchto nástrojů jsem již pracoval nebo dokonce pracuji, což byl jeden z důvodů proč jsem toho seznámení oddělil od vlastní implementace. Dále na řešení problémů ohledně CORS a jako poslední čas vlastní implementace API.

Úvodní seznámení s nástroji	42 minut
Řešení problémů s CORS	5 hodin 46 minut
Vlastní implementace API	4 hodiny 26 minut

Změřené výsledky

Request loops	Request loop steps	Celkový počet requestů	Typ requestu	Celkový čas	Suma času jednotlivých requestů
1000	1	4000	Test request	9,923 s	58,206 s
600	1	4000	Všechny	3,388 s	1406 s
1000	1	8000	Všechny	19,97 s	2089 s
1500	1	12000	Všechny	28,7 s	6045 s
2000	1	16000	Všechny	48,88 s	32113 s
5000	1	40000	Všechny	154 s	299258 s

Pro hodnoty nad 5000 request loop při zvolení všech testovacích metod, což je dohromady asi 40 000 requestů, začne prohlížeč vracet chybu ERR_INSUFFICIENT_RESOURCES, tedy nedostatečné zdroje - přesto nedojde k pádu serveru.

PHP

Při implementaci API v PHP jsem se rozhodl použít framework Slim (www.slimframework.com), jedná se o micro framework přímo určený pro tvorbu API. Na tento nástroj, jsem narazil přes zmínku o jiném frameworku z předmětu KIV/WEB.

V PHP byl největší problém zajistit persistentní uložení dat bez databáze, jelikož requesty generované javascriptovým klientem získávaly pokaždé jiné session_id, byl jsem nucen tuto vlastnost obejít nastavením jednotného session_id pro všechny requesty. Toto by samozřejmě v běžné webové aplikaci bylo velké bezpečnostní riziko, ale v tomto testovacím případě to nevádí. V reálném nasazení se navíc počítá s použitím databáze jakožto perzistentního úložiště dat.

Čas pro implementaci

Výsledný čas jsem opět rozdělil, stejně jako v předchozím případě, na tři části. V případě PHP jsem myslel, že problém s CORS bude vyřešený velmi rychle, ale ukázalo se, že klient (potažmo přímo webový prohlížeč) posílá před samotným requestem vždy tzv. "preflighted request" (https://developer.mozilla.org/en-US/docs/Web/HTTP/Access_control_CORS#Preflighted_requests), který nejdříve pošle request OPTIONS a až následně samotný request námi zvolenou metodou. Tedy i v tomto případě si řešení CORS vyžádalo poměrně dost času.

Úvodní seznámení s nástroji	1 hodina 16 minut
Řešení problémů s CORS	2 hodiny 52 minut
Vlastní implementace API	3 hodiny 38 minut

Změřené výsledky

Request loops	Request loop steps	Celkový počet requestů	Typ requestu	Celkový čas	Suma času jednotlivých requestů
1000	1	4000	Test request	4,271 s	0
100	1	800	Všechny	0,412 s	0
200	1	1600	Všechny	1,053 s	0
500	1	4000	Všechny	2,880 s	0
600	1	4800	Všechny	2,530 s	0

Chybu ERR_INSUFFICIENT_RESOURCES začne prohlížeč při použití PHP serveru vracen už někde u hodnoty 600 request loops. Hodnota vracená při překročení zdrojů - ERR_INSUFFICIENT_RESOURCES - by se dala považovat za hodnotu kdy server přestane vyřizovat jednotlivé požadavky, ale po přezkoumání to vypadá na chybu generovanou přímo webovým prohlížečem, a při opakování dotazů z jiného prohlížeče je možné znovu provádět requesty. Navíc pokud uvážíme že hodnota 600 request loops znamená 4800 requestů odeslaných prakticky naráz, jde o docela dobrou dostupnost i v případě PHP serveru.

Závěr

Přestože chyba ERR_INSUFFICIENT_RESOURCES není zcela věrohodný údaj vypovídající o rychlosti API, ukazuje určité vodítko využití paměti.

Z hlediska paměťové náročnosti a z hlediska rychlosti vývoje je z mého hlediska lepší použít NodeJS. Nadruhou stranu PHP zase vede v rychlosti, i když tady jsou rozdíly velmi malé.

Upřímně jsem většinu problémů, které nastali neočekával a vůbec jsem netušil že otestovat různé implementace stejného API může být tolik obtížné.

Problémy

Bohužel moje lokální bezdrátová síť a druhý testovací notebook nebyli schopni unést zátěž nad 5000 requestů najednou. Propustnost sítě byla v tu chvíli na svém maximu a procesor běžel po ustálení mezi 65-85%.

Protože nemám veřejnou IP adresu, nemohl jsem test provést ani vzdáleně s pomocí počítače některého z mých spolužáků nebo přátel.

Byl jsem tedy přinucen použít klientskou aplikaci na stejném stroji na jakém běží server, ukázalo se to jako použitelné řešení. Při čtení výsledků je ovšem nutné si uvědomit, že nedochází k žádnému spoždění při přenosu požadavků. Naštěstí procesor a paměť, které jsou v tomto případě zatíženy jak samotným serverem, tak rovněž i klientskou aplikací běželi vyrovnaně někde okolo 65% RAM a 42-58% CPU.

Původně měla být v testu zahrnuta ještě implementace serveru v programovacím jazyce Java, postavená na JAX-RS a nebo JPA, ale vzhledem k neočekávané náročnosti implementace jednotlivých serverů, zdržení kvůli řešení potíží ohledně CORS a také z důvodu, že samotná Java byla na seznam přidána pouze ze zvědavosti (fakticky jsem uvažoval o finální implementaci jen mezi NodeJS a PHP) jsem server v tomto jazyce neimplementoval.

U NodeJS serveru, který má k dispozici pouze jedno vlákno (záložka webového prohlížeče má také pouze jedno) došlo k jakémusi ustálenému stavu, kdy se jednotlivé procesy začali zřejmě pravidelně střídat. A rychlost vyřizování požadavků začalo stoupat dost pravidelně.

Přínos

I přesto pro mně byla příprava tohoto testu prospěšná a výsledky využiji ve své diplomové práci. Hlavním přínosem byl problém ohledně CORS, který při samotné implementaci zabral hodně času s pochopením této, pro mě do té doby neznámé, vlastnosti. A řešení problémů s CORS mě tak již nebude trápit při implementaci skutečného API.

Jelikož dělám diplomovou práci společně ještě s jedním mým kolegou, bylo nalezení a vyzkoušení dokumentace samotného API v nástroji Apiary.io velkým přínosem, který v DP určitě využijeme.

Dále bylo dobrým zjištěním, že i přes velkou hromadu požadavků, které v jedné chvíli na server proudili, se jej nepodařilo donutit schodit a zastavit jeho činnost.

Budoucí postup

Hlavně podle rychlosti a komfortnosti práce při implementaci, jsem se přiklonil k výběru NodeJS a Coffeescriptu, který by mohl v kombinaci například s nerelační databází poskytovat dobrou cestu k tvorbě API jaké zamýšlím.

Přesto jsem se rozhodl, že svůj test zopakuji začátek semestru, v nějaké univerzitní počítačové laboratoři, abych nakonec do své diplomové práce získal trochu lepší výsledky měření.

Příloha A: Javascript AJAX request client

KIV/VSP API testing AJAX Client v1.3

Request Send request

Request settings Query settings 1 Docs

Request base URL: Where's API server running.

Number of request loops: How much requests will be sent.

Number of request loop steps: How much requests will be sent at the same time.

Interval between requests:
 milliseconds

Send request

Response Clear all App restart

HTTP status count: 200: 1 201: 0 400: 0 401: 0 404: 0 500+: 0
Total requests time: 37.36000000000058 milliseconds

#	\$	Status
1	1	200 OK

Time: 37.36000000000058 milliseconds

Response URL:
http://localhost:8080/api/users

```
[{"id_user":1,"name":"Antonín","lastname":"Neumann","birthday":"1989-
```

Příloha B: HW info

Zde je přiložen výstup programu HWiNFO64 (5.12-2740), který popisuje kompletní hardwarové vybavení stroje použitého jako server.

Central Processor(s) -----

[CPU Unit Count]

Number Of Processor Packages (Physical): 1
Number Of Processors Cores: 2
Number Of Logical Processors: 4

Intel Core i5-3210M -----

[General Information]

Processor Name: Intel Core i5-3210M
Original Processor Frequency: 2500.0 MHz
Original Processor Frequency [MHz]: 2500
CPU ID: 000306A9
CPU Brand Name: Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz
CPU Vendor: GenuineIntel
CPU Stepping: L1
CPU Code Name: Ivy Bridge-MB PGA SV
CPU S-Spec: SR0MZ
CPU Thermal Design Power (TDP): 35.0 W
CPU IA Cores Thermal Design Current (TDC): 112.0 A
CPU GT Cores Thermal Design Current (TDC): 46.0 A
CPU Power Limits (Max): Power = Unlimited, Time = 64.00 sec
CPU Power Limit 1 (Long Duration): Power = 35.00 W, Time = 28.00 sec [Locked]
CPU Power Limit 2 (Short Duration): Power = 43.75 W, Time = Unlimited [Locked]
CPU Type: Production Unit
CPU Platform: rPGA988B

Microcode Update Revision: 1B

Number of CPU Cores: 2

Number of Logical CPUs: 4

[Operating Points]

CPU LFM (Minimum): 1200.0 MHz = 12 x 100.0 MHz

CPU HFM (Maximum): 2500.0 MHz = 25 x 100.0 MHz

CPU Turbo: 3100.0 MHz = 31 x 100.0 MHz [Locked]

CPU Current: 2893.7 MHz = 29 x 99.8 MHz @ 0.9457 V

CPU Bus Type: Intel Direct Media Interface (DMI) v2.0

Maximum DMI Link Speed: 5.0 GT/s

Current DMI Link Speed: 5.0 GT/s

[Cache and TLB]

L1 Cache: Instruction: 2 x 32 KBytes, Data: 2 x 32 KBytes

L2 Cache: Integrated: 2 x 256 KBytes

L3 Cache: 3 MBytes

Instruction TLB: 2MB/4MB Pages, Fully associative, 8 entries

Data TLB: 4 KB Pages, 4-way set associative, 64 entries

[Standard Feature Flags]

[Extended Feature Flags]

[Enhanced Features]

[CPU IVB Features]

[Memory Ranges]

[MTRRs]

Motherboard -----

[Computer]

Computer Brand Name: LENOVO ThinkPad Edge E330

[Motherboard]

Motherboard Model: LENOVO 33542KG

Motherboard Chipset: Intel HM77 (Panther Point)

Motherboard Slots: 4xPCI Express x1

PCI Express Version Supported: v2.0

USB Version Supported: v3.0

[PCH Features]

USB Port Count: 14 ports

RAID Capability: RAID0/1/5/10 + Smart Response Technology

SATA Ports 2 and 3: Supported

SATA Port 1 6 Gb/s: Supported

SATA Port 0 6 Gb/s: Supported

PCI Interface: Not Supported

PCI Express Ports 5 and 6: Supported

USB Redirect (USBr): Not Supported

Intel Anti-Theft Technology: Not Supported

PCI Express Ports 7 and 8: Supported

PCH Integrated Graphics Support: Supported

Data Center Manageability Interface (DCMI): Supported

Node Manager: Supported

[BIOS]

BIOS Manufacturer: Phoenix Technologies

BIOS Date: 07/13/2012

BIOS Version: H3ET31WW(1.14)

UEFI BIOS: Capable

Super-IO/LPC Chip: Unknown

Intel ME -----

[ME Host Status]

ME Current Working State: Normal

Manufacturing Mode: Not Active

ME Current Operation Mode: Normal

[Intel Manageability Engine Features]

Intel ME Version: 8.0, Build 1441, Hot Fix 4

Intel ME Recovery Image Version: 8.0, Build 1441, Hot Fix 4
Intel ME FITC Version: 8.0, Build 1441, Hot Fix 4
Intel AMT Version: 8.0.4, Build 1441
Intel AMT Applications Version: 8.0.4
Flash Version: 8.0.4
Netstack Version: 8.0.4
Recovery Version: 8.0.4, Build 1441
BIOS Version: H3ET31WW(1.14)

[ME Firmware Capabilities]

Full Network Manageability: Not Capable
Standard Network Manageability: Not Capable
Manageability (AMT): Capable
Small Business Advantage: Capable
Manageability Upgrade: Not Capable
Intel Anti-Theft: Capable
Capability Licensing Service: Capable
Virtualization Engine: Not Capable
Power Sharing Technology (MPC): Capable
ICC Over Clocking: Capable
Protected Audio Video Path (PAVP): Capable
Identity Protection Technology (IPT): Not Capable
Remote PC Assist (RPAT): Not Capable
IPV6: Not Capable
KVM Remote Control: Not Capable
Outbreak Containment Heuristic (OCH): Not Capable
Virtual LAN (VLAN): Capable
Cipher Transport Layer (TLS): Capable
Wireless LAN (WLAN): Not Capable
Platform Trust Technology (PTT): Not Capable
Near Field Communication (NFC): Not Capable

[ME Firmware Feature State]

Full Network Manageability: Disabled

Standard Network Manageability:	Disabled
Manageability (AMT):	Enabled
Small Business Advantage:	Capable
Manageability Upgrade:	Not Capable
Intel Anti-Theft:	Enabled
Capability Licensing Service:	Enabled
Virtualization Engine:	Disabled
Power Sharing Technology (MPC):	Enabled
ICC Over Clocking:	Enabled
Protected Audio Video Path (PAVP):	Enabled
Identity Protection Technology (IPT):	Not Capable
Remote PC Assist (RPAT):	Disabled
IPV6:	Disabled
KVM Remote Control:	Disabled
Outbreak Containment Heuristic (OCH):	Disabled
Virtual LAN (VLAN):	Capable
Cipher Transport Layer (TLS):	Enabled
Wireless LAN (WLAN):	Disabled
Platform Trust Technology (PTT):	Disabled
Near Field Communication (NFC):	Disabled
[ME Firmware Platform Type]	
Platform Target Usage Type:	Mobile
Platform Target Market Type:	Consumer
SKU:	Regular SKU
ME Firmware Image Type:	5.0MB Firmware
Platform Brand:	Intel Small Business Advantage (SBA)
Host ME Region Flash Protection Override (HMRFPPO) Status: Locked	

Memory -----

[General information]

Total Memory Size:	8 GBytes
--------------------	----------

Total Memory Size [MB]: 8192

[Current Performance Settings]

Maximum Supported Memory Clock: 1300.0 MHz

Current Memory Clock: 798.3 MHz (8 : 1 ratio)

Current Timing (tCAS-tRCD-tRP-tRAS): 11.0-11-11-30

Memory Runs At: Dual-Channel

Command Rate: 1T

Read to Read Delay (tRD_RD) Different Rank: 1T

Read to Read Delay (tRD_RD) Different DIMM: 3T

Write to Write Delay (tWR_WR) Different Rank: 3T

Write to Write Delay (tWR_WR) Different DIMM: 5T

Read to Write Delay (tRD_WR) Same Rank: 3T

Read to Write Delay (tRD_WR) Different Rank: 5T

Read to Write Delay (tRD_WR) Different DIMM: 5T

Write to Read Delay (tWR_RD) Same Rank (tWTR): 6T

Write to Read Delay (tWR_RD) Different Rank: 1T

Write to Read Delay (tWR_RD) Different DIMM: 1T

Read to Precharge Delay (tRTP): 6T

Write to Precharge Delay (tWTP): 23T

Write Recovery Time (tWR): 12T

RAS# to RAS# Delay (tRRD): 5T

Refresh Cycle Time (tRFC): 208T

Four Activate Window (tFAW): 24T

Row: 0 - 4096 MB PC3-12800 DDR3 SDRAM Corsair CMSO4GX3M1A1600C11 -----

[General Module Information]

Module Number: 0

Module Size: 4096 MBytes

Memory Type: DDR3 SDRAM

Module Type: SO-DIMM

Memory Speed: 800.0 MHz (DDR3-1600 / PC3-12800)

Module Manufacturer: Corsair

Module Part Number: CMSO4GX3M1A1600C11
Module Revision: 0
Module Serial Number: 0
Module Manufacturing Date: Year: 2000, Week: 0
Module Manufacturing Location: 1
SDRAM Manufacturer: Unknown
Error Check/Correction: None

[Module characteristics]

Row Address Bits: 16
Column Address Bits: 10
Number Of Banks: 8
Module Density: 4096 Mb
Number Of Ranks: 1
Device Width: 8 bits
Bus Width: 64 bits
Module Nominal Voltage (VDD): 1.5 V

[Module timing]

Minimum SDRAM Cycle Time (tCKmin): 1.250 ns
CAS# Latencies Supported: 6, 7, 9, 11
Minimum CAS# Latency Time (tAAmin): 13.500 ns
Minimum RAS# to CAS# Delay (tRCDmin): 13.500 ns
Minimum Row Precharge Time (tRPmin): 13.500 ns
Minimum Active to Precharge Time (tRASmin): 36.500 ns
Supported Module Timing at 800.0 MHz: 11-11-11-30
Supported Module Timing at 666.7 MHz: 9-9-9-25
Supported Module Timing at 600.0 MHz: 9-9-9-22
Supported Module Timing at 466.7 MHz: 7-7-7-18
Supported Module Timing at 400.0 MHz: 6-6-6-15
Minimum Write Recovery Time (tWRmin): 15.000 ns
Minimum Row Active to Row Active Delay (tRRDmin): 6.000 ns
Minimum Active to Active/Refresh Time (tRCmin): 48.125 ns
Minimum Refresh Recovery Time Delay (tRFCmin): 260.000 ns

Minimum Internal Write to Read Command Delay (tWTRmin): 7.500 ns
Minimum Internal Read to Precharge Command Delay (tRTPmin): 7.500 ns
Minimum Four Activate Window Delay Time (tFAWmin): 30.000 ns

[Features]

Partial Array Self Refresh (PASR): Supported
On-die Thermal Sensor (ODTS) Readout: Not Supported
Auto Self Refresh (ASR): Not Supported
Extended Temperature 1X Refresh Rate: Not Supported
Extended Temperature Range: Supported
Module Temperature Sensor: Not Supported
Pseudo Target Row Refresh (pTRR): Not Supported
Module Nominal Height: 29 - 30 mm
Module Maximum Thickness (Front): 2 - 3 mm
Module Maximum Thickness (Back): 2 - 3 mm

Row: 2 - 4096 MB PC3-12800 DDR3 SDRAM Corsair CMSO4GX3M1A1600C11 -----

[General Module Information]

Module Number: 2
Module Size: 4096 MBytes
Memory Type: DDR3 SDRAM
Module Type: SO-DIMM
Memory Speed: 800.0 MHz (DDR3-1600 / PC3-12800)
Module Manufacturer: Corsair
Module Part Number: CMSO4GX3M1A1600C11
Module Revision: 0
Module Serial Number: 0
Module Manufacturing Date: Year: 2000, Week: 0
Module Manufacturing Location: 1
SDRAM Manufacturer: Unknown
Error Check/Correction: None

[Module characteristics]

Row Address Bits: 16

Column Address Bits: 10
Number Of Banks: 8
Module Density: 4096 Mb
Number Of Ranks: 1
Device Width: 8 bits
Bus Width: 64 bits
Module Nominal Voltage (VDD): 1.5 V

[Module timing]

Minimum SDRAM Cycle Time (tCKmin): 1.250 ns
CAS# Latencies Supported: 6, 7, 9, 11
Minimum CAS# Latency Time (tAAmin): 13.500 ns
Minimum RAS# to CAS# Delay (tRCDmin): 13.500 ns
Minimum Row Precharge Time (tRPmin): 13.500 ns
Minimum Active to Precharge Time (tRASmin): 36.500 ns
Supported Module Timing at 800.0 MHz: 11-11-11-30
Supported Module Timing at 666.7 MHz: 9-9-9-25
Supported Module Timing at 600.0 MHz: 9-9-9-22
Supported Module Timing at 466.7 MHz: 7-7-7-18
Supported Module Timing at 400.0 MHz: 6-6-6-15
Minimum Write Recovery Time (tWRmin): 15.000 ns
Minimum Row Active to Row Active Delay (tRRDmin): 6.000 ns
Minimum Active to Active/Refresh Time (tRCmin): 48.125 ns
Minimum Refresh Recovery Time Delay (tRFCmin): 260.000 ns
Minimum Internal Write to Read Command Delay (tWTRmin): 7.500 ns
Minimum Internal Read to Precharge Command Delay (tRTPmin): 7.500 ns
Minimum Four Activate Window Delay Time (tFAWmin): 30.000 ns

[Features]

Partial Array Self Refresh (PASR): Supported
On-die Thermal Sensor (ODTS) Readout: Not Supported
Auto Self Refresh (ASR): Not Supported
Extended Temperature 1X Refresh Rate: Not Supported
Extended Temperature Range: Supported

Module Temperature Sensor: Not Supported
Pseudo Target Row Refresh (pTRR): Not Supported
Module Nominal Height: 29 - 30 mm
Module Maximum Thickness (Front): 2 - 3 mm
Module Maximum Thickness (Back): 2 - 3 mm

Network -----

Intel Centrino Wireless-N 2230 BGN 2x2 HMC WiFi/Bluetooth Adapter -----

[General information]

Network Card: Intel Centrino Wireless-N 2230 BGN 2x2 HMC
WiFi/Bluetooth Adapter

Vendor Description: Microsoft

MAC Address: 84-A6-C8-92-8D-67

[Capabilities]

Maximum Link Speed: 144 Mbps

Transmit Buffer Size: 6201344 Bytes

Receive Buffer Size: 6201344 Bytes

Hardware ID: PCI\VEN_8086&DEV_0888&SUBSYS_42628086&REV_C4

[Driver Information]

Driver Manufacturer: Intel Corporation

Driver Description: Intel(R) Centrino(R) Wireless-N 2230

Driver Provider: Intel

Driver Version: 15.16.0.2

Driver Date: 06-Jan-2015

DeviceInstanceId
PCI\VEN_8086&DEV_0888&SUBSYS_42628086&REV_C4\4&35CC4061&0&00E1

RealTek Semiconductor RTL8168/8111 PCI-E Gigabit Ethernet NIC -----

[General information]

Network Card: RealTek Semiconductor RTL8168/8111 PCI-E
Gigabit Ethernet NIC

Vendor Description: Realtek PCIe GBE Family Controller

MAC Address: F0-DE-F1-A1-67-B8

[Capabilities]

Maximum Link Speed: 1000 Mbps

Transmit Buffer Size: 193792 Bytes

Receive Buffer Size: 775168 Bytes

Hardware ID: PCI\VEN_10EC&DEV_8168&SUBSYS_500617AA&REV_07

[Driver Information]

Driver Manufacturer: Realtek

Driver Description: Řadič Realtek PCIe GBE Family Controller

Driver Provider: Realtek

Driver Version: 9.1.401.2015

Driver Date: 01-Apr-2015

DeviceInstanceId
PCI\VEN_10EC&DEV_8168&SUBSYS_500617AA&REV_07\4&24FD4A66&0&00E3