

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA EKONOMICKÁ

Diplomová práce

**Distribuované výpočty s využitím „open source“
infrastruktury**

**Distributed computing using open source
infrastructure**

Bc. Josef HRUŠKA

Plzeň 2012

Místo pro zadání

Čestné prohlášení

Prohlašuji, že jsem diplomovou práci na téma

Distribuované výpočty s využitím „open source“ infrastruktury

vypracoval samostatně pod odborným dohledem vedoucího diplomové práce za použití pramenů uvedených v příložené bibliografii.

V Plzni, dne

.....

podpis autora

Poděkování

Tímto bych rád poděkoval vedoucímu práce panu RNDr. Jaroslavu Potměšilovi, CSc. za odborné rady a četné konzultace. Dále bych rád poděkoval rodině a přítelkyni za maximální podporu nejen během psaní této práce.

Obsah

Úvod	8
1 Distribuované systémy.....	9
1.1 Definice	9
1.2 Vztahy mezi komponentami systému	10
1.3 Trendy v distribuovaných systémech.....	11
1.3.1 Moderní internet.....	11
1.3.2 Mobilní výpočetní technika	12
1.3.3 Distribuované multimediální systémy.....	12
1.3.4 Distribuovaný systém jako služba.....	13
1.4 Motivace pro využití distribuovaných systémů	13
1.5 Modely distribuovaných systémů	14
1.5.1 Klient – server.....	14
1.5.2 Peer-to-peer.....	14
1.5.3 Cloud computing.....	15
1.5.4 Grid computing	16
1.5.5 Dobrovolnické počítání	17
2 Úvod do systému BOINC	19
2.1 Historie distribuovaných výpočtů	19
2.2 BOINC.....	19
2.2.1 Vznik	19
2.2.2 Popis	20
2.2.3 Cíle	20
2.2.4 Architektura	21
3 Vytvoření BOINC projektu.....	25
3.1 Instalace BOINC	25
3.2 BOINC projekt.....	25
3.2.1 Skript „make_project“.....	25
3.2.2 Adresářová struktura projektu	26
3.2.3 Databáze projektu	27
3.2.4 Konfigurační soubor projektu.....	27
3.2.5 Webové stránky projektu.....	28

3.2.6	Projektová aplikace	30
3.3	Bezpečnost projektu	31
3.3.1	Falšování výsledků a pozměňování získaného kreditu	32
3.3.2	Distribuce škodlivého spustitelného souboru	32
3.3.3	Útok na datový server	32
3.3.4	Krádež informací z účastnických účtů útokem na server	33
3.3.5	Krádež informací z účastnických účtů síťovým útokem.....	33
3.3.6	Krádež souborů projektu	33
3.3.7	Úmyslné zneužití účastníků projektem	34
3.3.8	Náhodné zneužití účastníků projektem	34
3.4	Získání a udržení dobrovolníků	34
3.4.1	Projektové webové stránky.....	34
3.4.2	Propagace.....	35
3.4.3	Emailové mechanismy	35
4	Zapojení do projektu z pohledu klienta	36
4.1	Komunikace počítače s projektovým serverem	36
4.2	Pravidla využití BOINC	37
4.2.1	Využívejte BOINC na autorizovaných počítačích.....	37
4.2.2	Využití počítače programem BOINC.....	37
4.2.3	Ochrana osobních údajů	37
4.2.4	Odpovědnost	38
4.3	Týmy	38
4.4	Kredit.....	38
4.5	Alternativní zařízení provádějící výpočty	39
4.5.1	Výpočty pomocí GPU	39
4.5.2	Výpočty pomocí Play Station 3	39
4.5.3	Výpočty pomocí mobilních zařízení	39
4.6	Negativní vlivy zapojení do projektu	40
4.6.1	Zahřívání komponent	40
4.6.2	Snížení životnosti.....	40
4.6.3	Zatěžování internetového připojení.....	40
4.6.4	Zvýšení spotřeby elektrické energie.....	40
5	Modelový distribuovaný výpočet.....	41

5.1	Serverová část	41
5.1.1	BOINC virtuální server	41
5.1.2	Vytvoření projektu	43
5.1.3	Přidání aplikace	44
5.1.4	Spuštění projektu	50
5.1.5	Výsledky	52
5.1.6	Webové rozhraní pro administraci	52
5.2	Klientská část	53
5.2.1	Připojení k projektu	54
5.2.2	Nastavení účtu	55
5.2.3	BOINC Manager	56
5.3	Nastavení stránek projektu	59
6	Projekty distribuovaného počítání	62
6.1	Dosud získané výsledky distribuovaných výpočtů	62
6.2	BOINC projekty	62
6.2.1	PrimeGrid	63
6.2.2	SETI@home	64
6.2.3	MilkyWay@home	65
6.2.4	Collatz Conjecture	65
6.2.5	DNETC@home	65
6.2.6	World Community Grid	66
6.2.7	Einstein@home	67
6.2.8	GPU GRID	67
6.3	Ostatní projekty	68
6.3.1	Distributed.net	68
6.3.2	Folding@home	68
6.3.3	Wieferich@home	68
	Závěr	69
	Seznam obrázků	70
	Seznam použitých zkratk	71
	Seznam použité literatury	72
	Seznam příloh	74

Úvod

Distribuované výpočty se v posledních letech staly předmětem velkého zájmu v mnoha oblastech dnešního světa. S jejich pomocí může každý člověk vlastní počítač pomáhat především ve výzkumných projektech, například od výzkumu léků proti vážným chorobám až po hledání mimozemských civilizací. Stačí k tomu poskytnout výpočetní výkon vlastního počítače ve chvílích, kdy jej nevyužíváme naplno. Zapojení se do nějakého projektu využívajícího distribuované výpočty není nic složitého a opravdu to může výrazně pomoci při vědeckých výzkumech. Předložená práce se zabývá distribuovanými výpočty a popisuje několik projektů, do kterých je možno se v současné době zapojit. Je také umožněno vytvořit vlastní nový projekt.

V první kapitole této práce jsou popsány distribuované systémy, jejich definice, vztahy mezi jednotlivými komponentami systému a trendy a modely distribuovaných systémů. Druhá kapitola se zabývá systémem Berkeley Open Infrastructure for Network Computing (dále jen BOINC), který byl vybrán k realizaci modelového distribuovaného výpočtu. Obsahuje údaje o vzniku, cílích a architektuře tohoto systému. Třetí kapitola popisuje vytvoření BOINC projektu, bezpečnost a možnosti při získávání dobrovolníků pro projekt. Obsahem čtvrté kapitoly je zapojení se do projektu BOINC z pohledu klienta. V páté kapitole je popsáno provedení modelového distribuovaného výpočtu. Nejprve je ukázána instalace serveru, jeho nastavení a vytvoření projektu. Následuje připojení klienta na vytvořený projekt a úprava webových stránek vytvořeného projektu. Poslední šestá kapitola obsahuje informace o současně běžících projektech distribuovaných výpočtů a některé dosud získané výsledky projektů.

Hlavním cílem této práce je provedení modelového distribuovaného výpočtu. Prvním úkolem je vypracovat teoretický vstup k danému tématu. Dalším cílem je zvolit vhodný „open source“ softwarový nástroj a provést modelový distribuovaný výpočet. Posledním a neméně důležitým cílem je popsat současný stav distribuovaných výpočtů a jejich využití v různých oblastech dnešního světa.

1 Distribuované systémy

Oblast distribuovaných výpočtů zahrnuje jak samotné výpočty, tak i informační přístupy k mnohonásobnému zpracování částí spojených jakoukoliv komunikační sítí. Od příchodu internetu v roce 1970 zde byl trvalý růst nových aplikací, které vyžadují distribuované zpracování. To bylo způsobeno pokrokem v síťových a hardwarových technologiích, klesajícími náklady na hardware a větší informovaností koncových uživatelů. Tyto faktory přispěly k tomu, že distribuované výpočty se staly nákladově efektivní, vysoce výkonné a odolné proti chybám. (Kshemkalyani, Singhal, 2008)

Na přelomu tisíciletí došlo k prudkému růstu v rozšiřování a účinnosti internetu, který byl spojen s lepším přístupem k síťovým zdrojům pomocí World WideWeb (www) po celém světě. Spolu s dramatickým růstem v bezdrátových a mobilních síťových oblastech a poklesem cen za připojení k internetu a cen úložných zařízení dochází k rychlému růstu distribuovaných aplikací. Ten je doprovázen zájmem v oblasti distribuovaných výpočtů u vysokých škol, vládních organizací i soukromých institucí. (Kshemkalyani, Singhal, 2008)

1.1 Definice

Distribuovaný systém je kolekce nezávislých prvků, které spolupracují při řešení problému, který nemůže být řešen individuálně. U výpočetních systémů lze distribuovaný systém charakterizovat několika způsoby:

- Kolekce počítačů, které nesdílejí společnou paměť nebo společné fyzické hodiny, komunikují posíláním zpráv přes komunikační síť a kde každý počítač má svou vlastní paměť a svůj operační systém.
- Kolekce nezávislých počítačů, které se uživateli jeví jako jediný počítač.
- Distribuovaný systém popisuje širokou škálu počítačů - od slabě vázaných systémů, jako je Wide Area Network (WAN), k silně vázaným systémům, jako je Local Area Network (LAN), až k velmi silně vázaným systémům, jako jsou víceprocesorové systémy. (Kshemkalyani, Singhal, 2008)

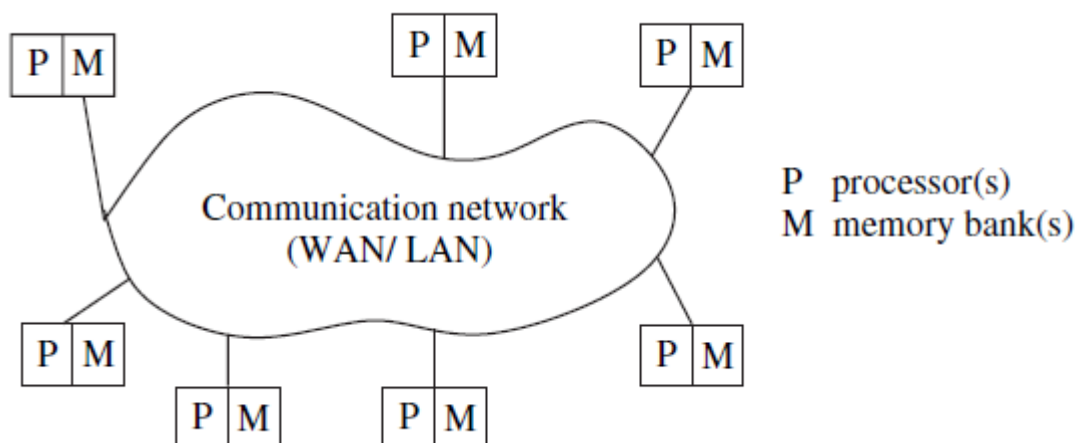
Distribuovaný systém lze tedy charakterizovat jako skupinu převážně autonomních procesorů, které komunikují přes komunikační síť a mají následující vlastnosti:

- **Nemají společné fyzické hodiny.** Toto je důležitý předpoklad, protože zavádí prvek distribuce v systému a vede k přirozené asynchronii mezi procesory.
- **Nesdílená paměť.** Nesdílená paměť je vlastností systému, která vyžaduje posílání zpráv pro komunikaci.
- **Geografické rozdělení.** Čím jsou procesory geograficky více odděleny, tím reprezentativnější distribuovaný systém je. Nicméně není nutné, aby byl procesor na WAN. V poslední době se za distribuovaný systém považuje i propojení procesorů pomocí sítě LAN.
- **Autonomie a heterogenita.** Jednotlivé procesory mají rozdílné rychlosti a každý může pracovat pod jiným operačním systémem. Obvykle nejsou součástí speciálního systému, ale spolupracují mezi sebou navzájem tak, že nabízejí služby nebo řešení problému společně. (Kshemkalyani, Singhal, 2008)

1.2 Vztahy mezi komponentami systému

Typický distribuovaný systém je znázorněn na obrázku číslo 1. Každý počítač je jednotkou s pamětí a procesorem a počítače jsou propojeny do komunikační sítě.

Obr. č. 1: Typický distribuovaný systém

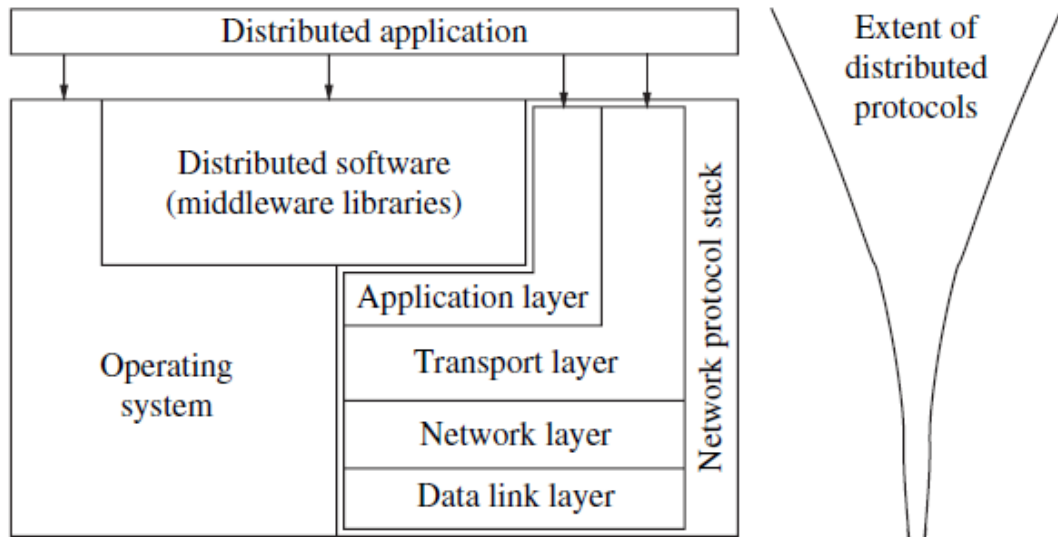


Zdroj: Kshemkalyani, Singhal, 2008

Obrázek číslo 2 ukazuje vztahy softwarových komponent běžících na každém počítači a využívajících operačního systému počítače a síťového protokolového zásobníku. Distribuovaný software je také označován jako middleware. Realizace v distribuovaném systému je provedení procesů, které při spolupráci vedou k dosažení společných cílů.

Realizace v distribuovaném systému se také nazývá výpočet nebo běh. (Kshemkalyani, Singhal, 2008)

Obr. č. 2: Interakce softwarových komponent na každém procesoru



Zdroj: Kshemkalyani, Singhal, 2008

Distribuovaný systém využívá vícevrstvou architekturu. Middleware je software, který řídí distribuovaný systém, když zajišťuje transparentnost různorodosti na úrovni platformy. Obrázek 2 schematicky znázorňuje interakci tohoto software se systémovými komponentami na každém procesoru. Zde se předpokládá, že middleware vrstva neobsahuje tradiční funkce aplikační vrstvy síťového protokolového zásobníku - http, mail, ftp a telnet. (Kshemkalyani, Singhal, 2008)

1.3 Trendy v distribuovaných systémech

Distribuované systémy procházejí řadou změn, které lze vysledovat v řadě významných trendů.

1.3.1 Moderní internet

Moderní internet je rozsáhlý soubor propojených počítačových sítí různých typů. Škála těchto typů sítí neustále roste a nyní zahrnuje celou řadu bezdrátových komunikačních technologií jako je WiFi, WiMAX, Bluetooth a 3G síť. Výsledkem je, že internet se stal všudypřítomným zdrojem a zařízení se mohou připojit kdykoliv a kdekoliv. Internet je velice rozsáhlým distribuovaným systémem. To umožňuje uživatelům, ať jsou kdekoliv, využívat služby, jako je World Wide Web, email či přenos souborů. Soubor služeb je

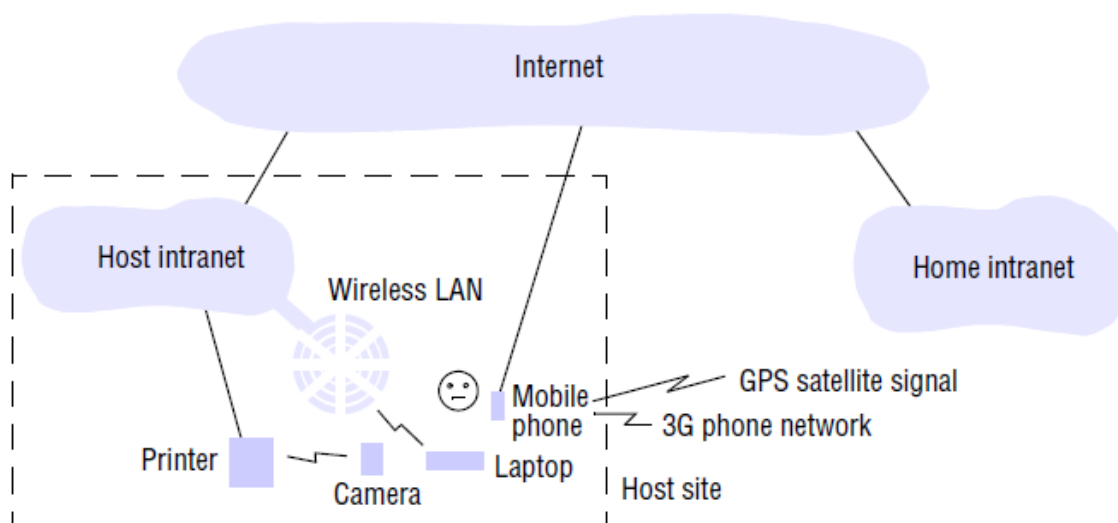
otevřený, může být rozšířen přidáním nových serverových počítačů a nových typů služeb. (Colouris, Dollimore, Kindberg, Blair, 2011)

1.3.2 Mobilní výpočetní technika

Technologický pokrok v miniaturizaci zařízení a bezdrátových sítích vedl k integraci malých a přenosných elektronických zařízení do distribuovaných systémů. Mezi tato zařízení patří například notebooky, kapesní zařízení jako mobilní telefony, smartphony a PDA zařízení. Přenositelnost spolu s možností připojení k síti na různých místech dělá tyto zařízení vhodné k využití v distribuovaných systémech. (Colouris, Dollimore, Kindberg, Blair, 2011)

Na obrázku číslo 3 lze vidět různá přenosná a kapesní zařízení, která jsou připojena k hostitelské síti. Dále je zde domácí síť. Jak hostitelská, tak domácí síť jsou připojeny k internetu. Mobilní telefon se k internetu může připojit buď pomocí hostitelské sítě nebo samostatně přes 3G síť.

Obr. č. 3: Přenosná a kapesní zařízení v distribuovaném systému



Zdroj: Colouris, Dollimore, Kindberg, Blair, 2011

1.3.3 Distribuované multimediální systémy

Dalším významným trendem v distribuovaných systémech je požadavek na podporu multimediálních služeb. Očekává se, že distribuovaný systém podporuje ukládání, přenos a prezentaci jednotlivých typů médií, jako jsou například textové zprávy, obrázky či video a audio soubory. (Colouris, Dollimore, Kindberg, Blair, 2011)

1.3.4 Distribuovaný systém jako služba

S rostoucí vypsělostí infrastruktury distribuovaných systémů roste počet firem zastávajících pohled na distribuovaný systém jako na komoditu či službu. V tomto modelu jsou zdroje poskytovány příslušnými dodavateli a efektivně pronajímány, než že by je koncový uživatel vlastnil. S tímto trendem úzce souvisí pojem Cloud computing, který je popsán v podkapitole 1.5.3. (Colouris, Dollimore, Kindberg, Blair, 2011)

1.4 Motivace pro využití distribuovaných systémů

Pro využití distribuovaného systému jsou většinou motivací některé nebo všechny tyto požadavky:

- **Distribuovaný běh ze své podstaty:** V mnoha případech je běh aplikací přirozeně rozdělen (např. převod peněz v bankovníctví).
- **Sdílení prostředků:** Prostředky, jako jsou periferie, údaje v databázích, stejně jako určitá data nemohou být replikovány na všech místech sítě, protože to často není ani praktické, ani nákladově efektivní. Dále nemohou být umístěny na jednom místě, protože přístup k tomuto místu může být problémový. Proto jsou tyto zdroje obvykle distribuovány v rámci celého systému.
- **Přístup ke geograficky vzdáleným datům a prostředkům:** Někdy není možné určitá data účastníci se distribuované realizace replikovat, protože tyto data pro tento účel mohou být příliš velká nebo příliš citlivá. Například mzdové údaje v rámci nadnárodní společnosti jsou příliš velké i příliš citlivé, aby mohly být replikovány na každé pobočce. Jsou tedy uloženy na centrálním serveru, který je dostupný z jednotlivých poboček.
- **Zvýšená spolehlivost:** Distribuovaný systém má přirozené schopnosti poskytovat zvýšenou spolehlivost díky možnosti replikace prostředků. Stejně tak je potřeba zvážit skutečnost, že u geograficky distribuovaných prostředků není pravděpodobné, že dojde za normálních okolností k současnému pádu či poruše. Spolehlivost zahrnuje několik aspektů:
 - Dostupnost – prostředky by měly být dostupné neustále
 - Integrita – stav či hodnoty prostředků by měly být správné i při souběžném víceprocesorovém přístupu
 - Odolnost proti chybám – schopnost systému zotavit se po selhání

- **Zvýšený poměr výkon/cena:** Sdílením prostředků a přístupem ke geograficky vzdáleným datům a prostředkům roste poměr výkon/cena.
- **Škálovatelnost:** Vzhledem k tomu, že jsou procesory obvykle propojeny pomocí WAN, přidání více procesorů nepředstavuje přímou překážku pro komunikaci sítě.
- **Modularita a rozšiřitelnost:** Různorodé procesory mohou být snadno přidány do systému bez ovlivnění výkonu, pokud tyto procesory běží pod stejnými middleware algoritmy. Podobně mohou být stávající procesory snadno nahrazeny jinými procesory.

(Kshemkalyani, Singhal, 2008)

1.5 Modely distribuovaných systémů

1.5.1 Klient – server

Model klient – server je zvláštním typem modelu distribuovaných systémů, který jasně rozlišuje vztah mezi dvěma počítači. Server poskytuje nějakou službu (např. zaslání aktuálních cen akcií) a klient využívá služby, které jsou na serveru k dispozici. (Neves, Mestre, Machado, Lopes, 2012)

Komunikace mezi klientem a serverem musí být spolehlivá. To znamená, že žádná data nesmí být ztracena a na klientskou stranu musí dorazit v pořadí, v jakém byly ze serveru poslány. K zajištění spolehlivé komunikace je využíván protokol TCP/IP. (Neves, Mestre, Machado, Lopes, 2012)

1.5.2 Peer-to-peer

Peer-to-peer se často označuje zkratkou P2P. Koncepce P2P komunikace je založena na myšlence, že každý uzel v síti je klient i server současně. Na rozdíl od modelu klient – server, P2P systémy spoléhají na hromadění aplikací a systémů, aby měli přístup k distribuovaným zdrojům decentralizovaným způsobem. V porovnání s architekturou klient-server je P2P levnější, protože obsah a systémy jsou udržovány individuálně. (Neves, Mestre, Machado, Lopes, 2012)

P2P systémy lze rozdělit do tří hlavních skupin: centralizované, decentralizované a hybridní. (Neves, Mestre, Machado, Lopes, 2012)

Koncept centralizovaných P2P systémů vychází z existence centrálního serveru, který vykonává jednoduché funkce pro systém, jako například plánování pracovní zátěže a ověřování výsledků. (Neves, Mestre, Machado, Lopes, 2012)

V decentralizovaných systémech uzly vykonávají všechny funkce bez zásahu centrálního serveru. V tomto případě musí být ověřování výsledků i komunikace zpracovány v každém uzlu a koordinovány s ostatními uzly. (Neves, Mestre, Machado, Lopes, 2012)

Hybridní systémy jsou mixem centralizovaných a decentralizovaných systémů. Objevuje se zde pojem super uzel, což je uzel vytvořený z normálního uzlu, který je odpovědný za centralizovanou práci, zatímco ostatní uzly pracují decentralizovaným způsobem. (Neves, Mestre, Machado, Lopes, 2012)

1.5.3 Cloud computing

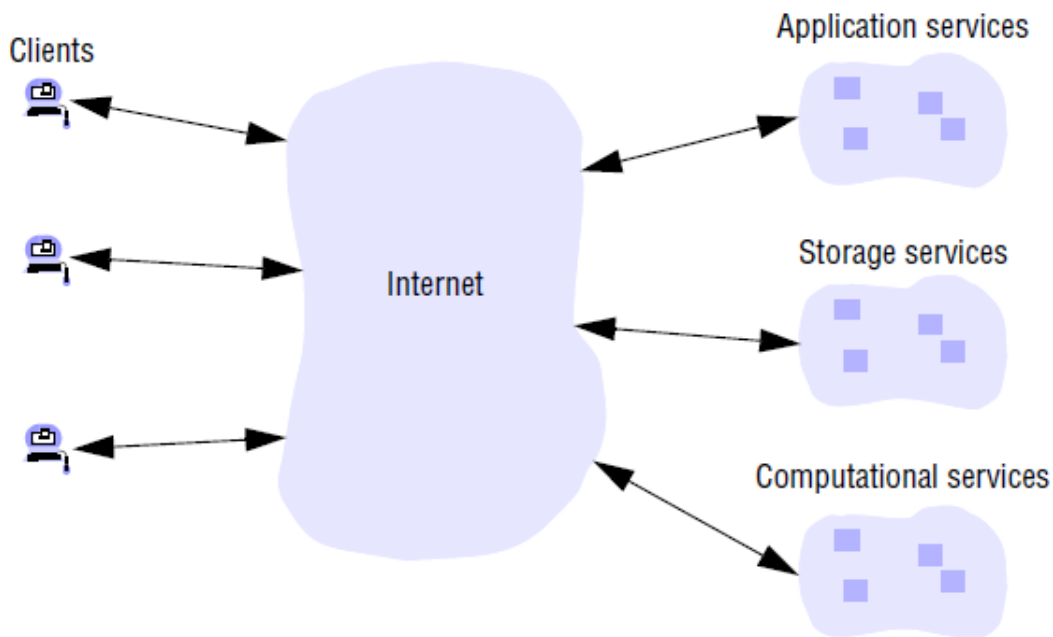
Cloud computing vychází z nižší potřeby uživatele mít aplikace nainstalovány na svých počítačích v důsledku zvýšené komunikační rychlosti a dostupnosti. (Neves, Mestre, Machado, Lopes, 2012)

Cloud je definován jako soubor internetové aplikace, úložiště a výpočetních služeb, který je dostatečný pro většinu uživatelů a tak jim umožňuje z větší části či úplně pracovat, aniž by ukládali data či aplikační software na vlastním zařízení. To lze vidět na obrázku číslo 4 na následující straně. Cloud computing snižuje požadavky na zařízení uživatelů, což umožňuje přístup k široké škále služeb a zdrojů i z velice jednoduchého zařízení. (Colouris, Dollimore, Kindberg, Blair, 2011)

Množství poskytovaných služeb a aplikací v cloudu se každý den zvyšuje a nejedná se jen o jednoduché nástroje. Firmy začínají využívat služby založené na cloudu pro všechny druhy manažersky a obchodně orientovaných úkolů. (Neves, Mestre, Machado, Lopes, 2012)

Nevýhodu u cloud computingu je obava o ochranu údajů a důvěrnost, především ochrana osobních údajů není brána jako dostatečně věrohodná. (Neves, Mestre, Machado, Lopes, 2012)

Obr. č. 4: Cloud computing



Zdroj: Colouris, Dollimore, Kindberg, Blair, 2011

1.5.4 Grid computing

Pojem grid byl nejprve použit jako metafora k elektrické rozvodné síti (electric power grid). Hlavní myšlenkou bylo, že přístup k výpočtu a datům měl být stejně jednoduchý jako připojení spotřebiče do zásuvky. Není jednoduché najít konsensuální definici, proto je jich zde uvedeno více:

- Výpočetní grid je hardwarová a softwarová infrastruktura, která poskytuje spolehlivý, konzistentní a levný přístup ke špičkovým výpočetním schopnostem.
- Výpočetní grid je technologie, která umožňuje virtualizaci zdrojů, jejich poskytnutí na vyžádání a sdílení služeb mezi organizacemi.
- Grid computing má schopnost pomocí sady otevřených standardů a protokolů získat přístup k aplikacím a datům, výpočetnímu výkonu, úložišti a k dalším zdrojům přes internet. Grid je druh paralelního a distribuovaného systému, který umožňuje sdílení, výběr a agregaci zdrojů distribuovaných přes různé domény, které jsou založeny na uživatelských požadavcích na dostupnost, kapacitu, výkon, náklady a kvalitu služeb. (Neves, Mestre, Machado, Lopes, 2012)

Do grid computingu je vzhledem k jeho nejednotné definici někdy řazeno i dobrovolnické počítání, zde je uvedeno jako samostatný model distribuovaného systému.

Oddělení grid computingu a dobrovolnického počítání lze nalézt také v oficiální BOINC wiki, která definuje grid computing následovně: Grid computing je forma distribuovaného počítání, při které organizace využívá své stávající počítače (desktopové či uzly clusteru) ke zpracování svého vlastního výpočetního úkolu.

1.5.5 Dobrovolnické počítání

Dobrovolnické počítání je uspořádání, ve kterém lidé (dobrovolníci) poskytují výpočetní prostředky na projekty, které tyto prostředky využívají k distribuovaným výpočtům. (BOINC, 2012, webová adresa v této citaci vzhledem k častému využití není úplná, je však jednoduché dostat se z uvedené stránky k zde užitým informacím)

Dobrovolníci jsou obvykle členy široké veřejnosti, kteří vlastní počítač s internetovým připojením. Organizace (školy i podniky) mohou také dobrovolně poskytnout své počítače. (BOINC, 2012)

Projekty jsou většinou akademické a provádí vědecký výzkum. I zde však existují výjimky, například GIMPS a distributed.net (dva velké projekty) nejsou akademické. (BOINC, 2012)

Několik aspektů vztahu projektu a dobrovolníků:

- Dobrovolníci jsou anonymní. Ačkoliv dobrovolník může být požádán o registraci, kde musí zadat emailovou adresu nebo jiné informace, není spojen s identitou v reálném životě.
- Vzhledem k anonymitě nejsou dobrovolníci projektům odpovědní. Pokud se dobrovolník nějakým způsobem chová špatně (např. vrací úmyslně nesprávné výsledky), projekt ho nemůže stíhat ani trestat.
- Dobrovolník musí projektu důvěřovat:
 - že poskytnuté aplikace nepoškodí jeho počítač nebo neproniknou do jeho soukromí
 - že projekt je pravdivý (o tom jakou práci dělají jeho aplikace a jak bude využito dosažených výsledků)

- že projekt dodržuje správné bezpečnostní postupy tak, aby jej nezneužili hackeři pro nebezpečnou činnost.

(BOINC, 2012)

Dobrovolnické počítání je důležité z několika důvodů:

- Vzhledem k obrovskému množství počítačů na světě může dobrovolnické počítání dodat více výpočetního výkonu než jakýkoliv jiný typ počítání. Tento výpočetní výkon umožňuje provádět vědecký výzkum, který by jinak nebyl možný.
- Dobrovolnický výpočetní výkon se nedá koupit, musí se zasloužit. Výzkumný projekt, který má omezené finanční prostředky, ale je o něj velký zájem veřejnosti, může dosáhnout obrovského výpočetního výkonu. Naproti tomu tradiční superpočítače jsou drahé, takže jsou dostupné jen pro aplikace, které si je mohou dovolit.
- Dobrovolnické počítání podporuje zájem veřejnosti o vědu a poskytuje veřejnosti hlas při určování směru vědeckého výzkumu.

(BOINC, 2012)

2 Úvod do systému BOINC

Pro realizaci distribuovaného výpočtu jsem si zvolil systém BOINC. Jedná se o „open source“ software, který je volně k dispozici a těší se velké oblibě u vědeckých pracovníků, kteří ho využívají pro své vědecké projekty.

2.1 Historie distribuovaných výpočtů

Prvním projektem, který stál na počátku zrodu odvětví distribuovaných výpočtů, byl projekt distributed.net, který již na přelomu let 1997 a 1998 vydal vlastního klienta pro distribuované výpočty. Jedná se o projekt v matematice, který se zabývá výpočty na základě Golomb Ruler (matematický termín, v češtině známý jako Golombovo pravítko) a lámáním šifer hrubou silou. (Vykouřil, 2008)

Druhým spuštěným projektem byl GIMPS (Great Internet Mersenne Prime Search), který se zabývá hledáním Mersennových prvočísel. Dalším projektem byl Folding@home, zabývající se výzkumy v oblasti biologie a hledající léky na různé lidské choroby. Tento projekt patří mezi nejdůležitější, nejznámější a také nejvýkonnější vědecké projekty distribuovaných výpočtů. (Vykouřil, 2008)

2.2 BOINC

2.2.1 Vznik

U zrodu BOINC byl projekt SETI (Search for Extraterrestrial Intelligence), který vyhledává mimozemské civilizace ve vesmíru dostupnou technikou. SETI výzkumy využívaly superpočítače, jejichž výkon byl obrovský, ale nedostačující, což vedlo ke zrodu myšlenky distribuovaných výpočtů. (Vykouřil, 2008)

V roce 1995 David Gedye s Craigem Kasnoffem navrhli projekt SETI@home. Jejich řešením bylo využití tisíců běžných počítačů, které by pracovaly jen krátký časový úsek, namísto jednoho drahého superpočítače, který by běžel dlouhou dobu. 17. 5. 1999 došlo ke spuštění projektu SETI@home a již po třech měsících měl projekt přes jeden milion uživatelů. (Vykouřil, 2008)

Díky popularitě a úspěchu projektu SETI@home vznikla myšlenka na vytvoření platformy, která by spojila více projektů založených na distribuovaných výpočtech. To vedlo ke vzniku BOINC, kdy 22. 6. 2004 byl projekt SETI@home integrován do systému BOINC a jako první projekt tohoto systému spuštěn. Postupně se přidávaly

další projekty, ať již zaběhlé v distribuovaných výpočtech tak i zcela nové. (Vykouřil, 2008)

2.2.2 Popis

BOINC je zkratka názvu Berkeley Open Infrastructure for Network Computing. Jedná se o open source softwarovou platformu pro dobrovolnické počítání či grid computing, která byla vyvinuta na Kalifornské univerzitě v Berkeley. (*Unofficial BOINC Wiki*, 2012, také tato webová adresa v citaci není úplná, užité informace jsou z ní však jednoduše dostupné)

Logo systému BOINC, které lze vidět na následujícím obrázku, se objevuje na všech internetových stránkách projektů, které využívají tento systém. Dokonce i mnou vytvořený projekt jej, jak se budete moci v kapitole 5 sami přesvědčit, obsahuje.

Obr. č. 5: Logo BOINC



Zdroj: BOINC, 2012

2.2.3 Cíle

Základním cílem je podpora tvorby projektů využívajících veřejné výpočetní zdroje a přesvědčení vlastníků počítačů k podílení se na jednom či více projektech. (Anderson, 2012)

Mezi specifické cíle patří:

- Redukování bariéry vstupu do dobrovolnického počítání – BOINC umožňuje vědeckým pracovníkům s průměrnými počítačovými dovednostmi vytvořit a spravovat rozsáhlé projekty s asi týdenní přípravnou prací a hodinou týdně na údržbu. Server se může skládat z jednoho stroje, který je vybaven běžným open source softwarem (např. Linux, Apache, PHP, MySQL, Python).
- Sdílení zdrojů mezi projekty – projekty založené na BOINC jsou autonomní, nejsou centrálně autorizovány nebo registrovány. Každý projekt provozuje vlastní servery a databáze. Přesto se mohou majitelé počítačů účastnit více projektů a mohou svůj výpočetní výkon rozdělit mezi tyto projekty. Pokud se většina uživatelů účastní více

projektů, zvětšuje se tím celkové využití zdrojů. Zatímco je jeden projekt pozastaven pro opravy, další projekty dočasně převezmou jeho výpočetní výkon. Na konkrétním počítači může procesor pracovat na jednom projektu, zatímco přes síť se přenášejí soubory k jinému projektu.

- Podpora různých aplikací – BOINC podporuje širokou škálu aplikací, poskytuje flexibilní a škálovatelné mechanismy pro distribuci dat. Stávající aplikace napsané v běžných jazycích, jako je C, C++, FORTRAN, mohou běžet jako BOINC aplikace s malými modifikacemi. Aplikace se mohou skládat z několika souborů. Nové verze aplikací lze nasadit, aniž by o tom museli účastníci projektu vědět.
- Odměna účastníků – dobrovolnické projekty musí motivovat účastníky, aby je přilákaly a také udržely. Primární motivací pro mnoho účastníků je kredit, což je číselná hodnota vyjadřující kolik výpočtů účastník provedl. BOINC poskytuje systém přidělování kreditu, který zohledňuje využití různých zdrojů (procesoru, síť, pevného disku), je společný pro více projektů a je vysoce odolný vůči podvodům. (Anderson, 2012)

2.2.4 Architektura

Na obrázku číslo 6 na následující straně je zachycena architektura systému BOINC. Všechny objekty mající modrou barvu jsou součástí softwarového balíku a červené objekty jsou součástí projektu, který je na systému BOINC provozován. (*Unofficial BOINC Wiki*, 2012)

Systém BOINC se skládá z komplexu hardware a software. Systém má dvě základní vrstvy - serverovou a klientskou. (*Unofficial BOINC Wiki*, 2012)

Serverová vrstva

Serverová vrstva obsahuje na straně softwaru systémový software včetně databáze a webových stránek a na straně hardwaru samotné servery. (*Unofficial BOINC Wiki*, 2012)

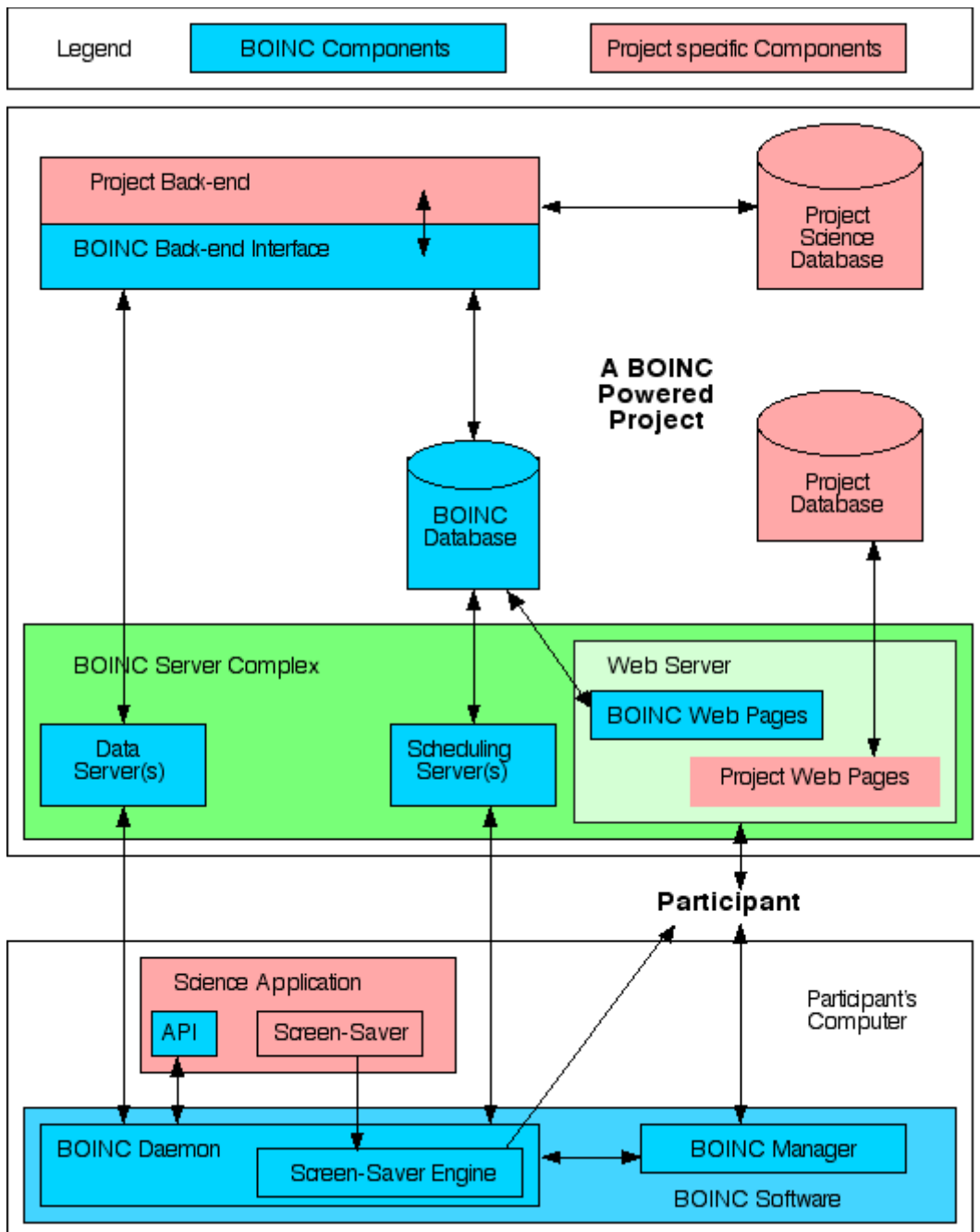
Databáze

BOINC využívá databázi nejméně ve dvou úrovních systému. Na jedné úrovni je BOINC databáze. Jedná se o online databázi, která uchovává informace o účastnících, jejich počítačích, týmech, výsledcích a dalších údajích. (*Unofficial BOINC Wiki*, 2012)

Druhá databáze uchovává informace o výsledcích vrácené od účastníků projektu, které prošly validací. Tato databáze je někdy nazývána vědeckou. (*Unofficial BOINC Wiki*, 2012)

BOINC využívá databázový systém MySQL.

Obr. č. 6: Architektura systému BOINC



Zdroj: Unofficial BOINC Wiki, 2012

Webové stránky projektu

System BOINC dodává každému projektu standardizovaný soubor webových stránek, které by měli tento projekt podpořit. Využívá k tomu softwarový webový server Apache. (*Unofficial BOINC Wiki*, 2012)

Webové stránky projektu jsou blíže popsány v podkapitole 3.2.5 a webové rozhraní pro administraci projektu v podkapitole 5.1.6.

Servery

Pro experimentování a ladění může být použit téměř jakýkoliv počítač jako BOINC server. Před zavedením projektu v širší míře by měl mít server dostatečnou výkonnost, dostupnost a bezpečnost.

Server by měl splňovat následující faktory:

- Internetové připojení by mělo mít odpovídající výkonnost a spolehlivost. Server musí mít statickou IP adresu.
- Server by měl disponovat dobrou rychlostí procesoru (Intel Dual Xeon, AMD Opteron), nejméně 2 GB RAM a 40 GB místa na disku.
- Měla by být zajištěna co největší spolehlivost (UPS napájení, RAID konfigurace disku, za provozu vyměnitelné náhradní díly, strojovna s kontrolovanou teplotou)
- Server by se měl nacházet za firewallem.
- Měly by být vypnuty všechny nepotřebné síťové služby, zejména ty, které používají nešifrovaná hesla (FTP, Telnet).

(BOINC, 2012)

Alternativou k užití vlastního serveru je zavedení BOINC serveru na Amazon Elastic Computing Cloud. Toto řešení je užitečné pro malé a střední projekty, kde může být cena za užití cloudu nižší, než pořízení vlastního hardware a platba za rychlé internetové připojení a elektrickou energii. (BOINC, 2012)

Klientská vrstva

Na klientské vrstvě jsou hardwarem jednotlivé počítače provozované účastníkem projektu a software obsahuje BOINC Manager a BOINC aplikace. (*Unofficial BOINC Wiki*, 2012)

BOINC Manager

Program umožňující připojení klienta k serveru. V podkapitole 5.2.3 je tento program popsán podrobněji. (*Unofficial BOINC Wiki*, 2012)

BOINC aplikace

Jedná se o aplikace, které provádí samotný výpočet. Každý projekt má své specifické aplikace, které BOINC Manager stáhne ze serveru do uživatelského počítače.

3 Vytvoření BOINC projektu

3.1 Instalace BOINC

BOINC aplikace mohou běžet pod operačním systémem Windows, Mac OS X i Unix, ale server musí být počítač se systémem Unix. (Creating and Configuring a BOINC Project, 2012)

Existují tři způsoby nastavení serveru, aby mohl hostit BOINC projekty:

- použití linuxových distribucí Debian či Ubuntu, pro které jsou připraveny BOINC server instalační balíčky
- stažení a nainstalování všech nutných softwarových komponent na libovolný Unix systém
- využití BOINC server virtuálního stroje (BOINC, 2012)

Pro modelový distribuovaný výpočet v páté kapitole je využit BOINC server virtuálního stroje.

3.2 BOINC projekt

BOINC projekt se skládá z databáze, adresářové struktury a konfiguračního souboru. Nejjednodušším způsobem vytvoření projektu je skript „make_project“, který vytvoří všechny tyto komponenty. Na jednom serveru může být více projektů, což je užitečné při testování a ladění. (BOINC, 2012)

3.2.1 Skript „make_project“

Skript „make_project“ vytváří serverové komponenty projektu. Před jeho použitím je potřeba mít správně nastaven BOINC server. (BOINC, 2012)

Konkrétně skript provádí následující instrukce:

- vytvoří projektový adresář a jeho podadresáře
- vytvoří projektové šifrovací klíče
- vytvoří a inicializuje MySQL databázi
- nakopíruje zdrojové a spustitelné soubory
- vygeneruje konfigurační soubor projektu

- vygeneruje šablony pro webový server
(BOINC, 2012)

Při spuštění skriptu je možné zadat několik parametrů instalace projektu, kterými jsou adresářové, URL, databázové a další různé možnosti. Jednou z možností je instalace testovací aplikace, kterou lze využít ke snadnému ověření správného nastavení serveru. Nevýhodou této aplikace je, že klient musí pracovat pod operačním systémem Linux. (BOINC, 2012)

3.2.2 Adresářová struktura projektu

Adresářová struktura typického BOINC projektu je zachycena na následujícím obrázku, kde „PROJECT“ je název projektu a „HOSTNAME“ je název serveru. (BOINC, 2012)

Obr. č. 7: Adresářová struktura projektu

```
PROJECT/
  apps/
  bin/
  cgi-bin/
  log_HOSTNAME/
  pid_HOSTNAME/
  download/
  html/
    inc/
    ops/
    project/
    stats/
    user/
    user_profile/
  keys/
  upload/
```

Zdroj: BOINC, 2012

Jednotlivé složky obsahují:

- apps: aplikace a klientem spustitelné soubory
- bin: serverové daemony a programy
- cgi-bin: CGI (Common Gateway Interface) programy
- log_HOSTNAME: log soubory jednotlivých částí serveru
- pid_hostname: uzamčené soubory a PID (process identifier) soubory
- download: úložiště dat pro stažení ze serveru
- html: soubory pro webové rozhraní

- keys: šifrovací klíče
- upload: úložiště dat nahraných na server
(BOINC, 2012)

Datový server pro velký projekt může do adresáře „download“ a „upload“ uložit stovky tisíc až miliony souborů. Pokud by tyto soubory byly uloženy pouze v těchto adresářích, datový server by jejich vyhledáváním strávil mnoho procesorového času. Proto BOINC využívá hierarchické „download“ a „upload“ adresáře. Každý má sadu 1024 podadresářů pojmenovaných 0 až 3ff. Soubory jsou do jednotlivých adresářů umístěny podle jejich názvu. K přesnému určení názvu adresáře, ve kterém se soubor nachází, jsou v BOINC vytvořeny různé funkce. (BOINC, 2012)

3.2.3 Databáze projektu

BOINC ukládá informace do MySQL databáze. Tato databáze je vytvořena skriptem „make_project“. Obvykle není nutné s touto databází přímo manipulovat, pokud to však administrátor projektu potřebuje, může využít MySQL příkazovou řádku nebo administrativní webové rozhraní (to je blíže popsáno v podkapitole 5.1.6). (BOINC, 2012)

Mezi hlavní tabulky patří:

- app – informace o aplikacích
- app_version – informace o verzích aplikací
- host – informace o počítačích účastníků
- platform – informace o platformách klientů a aplikací
- result – informace o stavu výsledků – stav validace, procesorový čas, stav ukončení
- user – informace o účastnících – email, uživatelské jméno, heslo
- workunit – informace o pracovních jednotkách
(*Unofficial BOINC Wiki*, 2012)

3.2.4 Konfigurační soubor projektu

Konfigurace projektu je popsána v souboru „*config.xml*“, který se nachází v adresáři projektu. Tento soubor je s výchozími hodnotami vytvořen skriptem „make_project“. Formát souboru je zachycen na následujícím obrázku. (BOINC, 2012)

Obr. č. 8: Konfigurační soubor projektu

```
<boinc>
  <config>
    [ configuration options ]
  </config>
  <daemons>
    [ list of daemons ]
  </daemons>
  <tasks>
    [ list of periodic tasks ]
  </tasks>
</boinc>
```

Zdroj: BOINC, 2012

Konfigurační možnosti

V konfiguračních možnostech lze nastavit plánovač, který kontroluje, jak jsou úlohy odeslány ke klientům. Plánovač také vybírá verze aplikací, které odešle klientovi, například určuje, zda bude aplikace 32 nebo 64 bitová. Kromě plánovače jsou v konfiguračních možnostech projektu informace o databázi, o přidělování kreditu, o politice mazání souborů na serveru nebo o funkcích webových stránek. (BOINC, 2012)

Daemon

Daemon je serverový program, který běží nepřetržitě. V souboru „*config.xml*“ je pro tento program zadán příkaz, který ho spouští, hostitel, na kterém běží, a jméno vstupního a výstupního souboru. (BOINC, 2012)

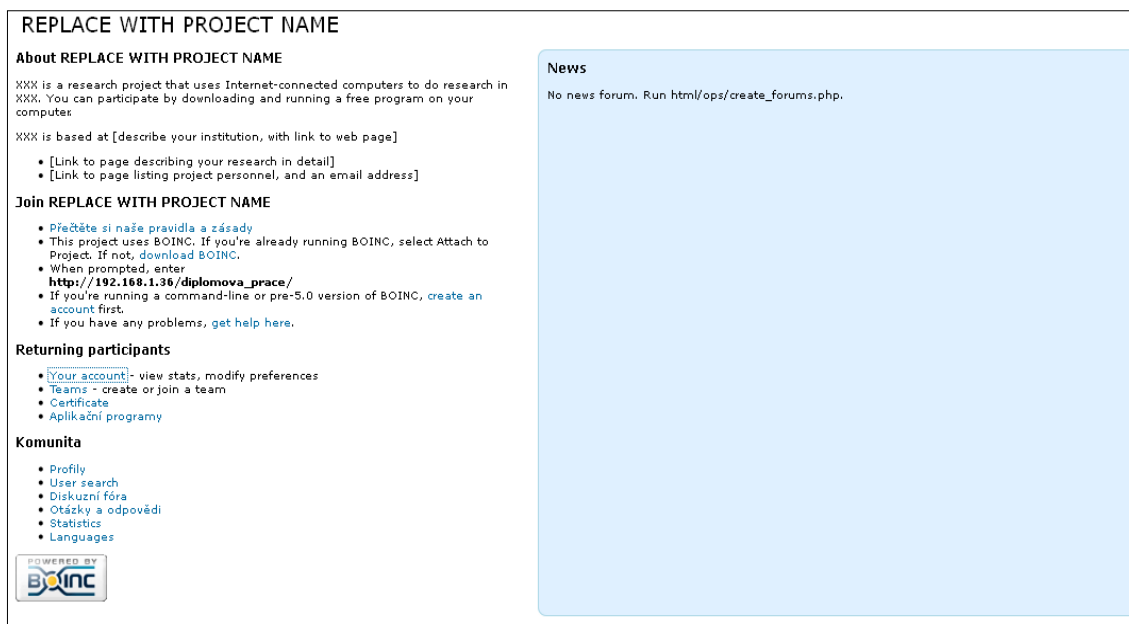
Úkoly

Jedná se o opakované úlohy, které systém provádí v nějakých intervalech. Mezi tyto úkoly patří například zápis statistických dat do XML souborů, generování HTML souborů s odkazy na uživatelské profily nebo aktualizace průměrného kreditu týmů, počítačů a uživatelů. (BOINC, 2012)

3.2.5 Webové stránky projektu

Při vytváření projektu pomocí skriptu „*make_project*“ dojde k vytvoření webových stránek projektu. Ty je potřeba upravit po obsahové i vzhledové stránce. Na následujícím obrázku se nachází hlavní webová stránka projektu, jak byla vygenerována skriptem „*make_project*“.

Obr. č. 9: Výchozí webové stránky projektu



Zdroj: vlastní zpracování, 2012

Vytvořené webové stránky se skládají z hlavní stránky, která obsahuje odkazy na další stránky projektu, mezi které patří:

- Pravidla a zásady – tato stránka obsahuje informace o ochraně soukromí, bezpečnosti zapojení do projektu, definuje zodpovědnost projektu vůči uživateli a informuje o využití účastnickova počítače a o možnosti zapojení se do ostatních projektů BOINC.
- Vytvoření účtu – na této stránce se nachází formulář pro založení účtu v projektu (pokud uživatel využívá BOINC Manager verze 5.2 a vyšší, registruje se prostřednictvím tohoto programu – tak to probíhá i v podkapitole 5.2.1 při přihlášení do vytvořeného projektu „Diplomová práce“).
- Přístup k účtu – stránky účtu uživatele jsou podrobně popsány v podkapitole 5.2.2
- Stránky týmů – lze vyhledávat týmy či si vytvořit svůj vlastní.
- Certifikát – na této stránce se uživateli vygeneruje certifikát, který obsahuje jeho jméno, datum zapojení do projektu a statistiku o vypočítané práci na projektu.
- Aplikační programy – stránka obsahující soupis aplikací projektu a platform, na kterých tyto aplikace běží.
- Diskusní fórum, vyhledávání mezi účastníky projektu, statistiky a žebříčky nejlepších účastníků, počítačů a týmů.

Předtím, než je projekt spuštěn pro veřejnost, je potřeba tyto webové stránky doplnit o obsah a design specifický pro daný projekt. Tomu by měla být přiložena vysoká důležitost, neboť webové stránky mají velký vliv na získání a udržení účastníků projektu. (BOINC, 2012)

Některých přizpůsobení lze dosáhnout změnou hlavní stránky projektu (to je ukázáno v podkapitole 5.3) a změnou šablony vzhledu. Další aspekty se mění pomocí konfiguračního souboru, který mimo konstant (například jméno projektu, URL projektu) obsahuje také různé funkce. Změna názvu projektu v konfiguračním souboru je také popsána v podkapitole 5.3. (BOINC, 2012)

BOINC je vybaven softwarem pro tvorbu diskusního fóra, aby mohli účastníci projektu mezi sebou jednoduše komunikovat. Pro spuštění fóra je potřeba spustit PHP skript, který vytvoří jednotlivé kategorie a pod nimi i samotná fóra. Také je potřeba vytvořit hlavní stránku diskusního fóra. (Creating and Configuring a BOINC Project, 2012)

Na webové stránky projektu je možno přidat i wiki. Slouží k tomu software MediaWiki, který používá například i Wikipedie. Projektoví administrátoři mohou snadno kontrolovat, kdo smí wiki číst nebo upravovat, na základě speciálního statutu uživatele (např. administrátor, moderátor) nebo na základě dosaženého kreditu. (BOINC, 2012)

V dnešní době je velice důležitou součástí webových stránek ochrana před automatickým vytvářením profilů a před spamem. Pro ochranu před vytvořením účtu roboty lze využít systém reCaptcha, který umožňuje modifikaci profilu pouze po opsání dvou slov z obrázku. Jinou možností ochrany je vyžadování minimálního kreditu. Pro ochranu před spamem lze využít systém Akismet, který je považován za nejlepší antispamový program na světě. (BOINC, 2012)

3.2.6 Projektová aplikace

Do vytvořeného projektu je potřeba přidat aplikaci, která provádí samotný výpočet. Projekt může mít jednu i více aplikací.

Vytvoření aplikace

BOINC umožňuje využít skoro všechny aplikace napsané v jakémkoliv jazyce. Nejjednodušším způsobem spuštění existujících aplikací pod BOINC je využití programu BOINC wrapper. (BOINC, 2012)

K vytvoření aplikace je možné využít BOINC API, což je sada funkcí v jazyce C++. Dnes velmi oblíbený jazyk JAVA není BOINC API podporován. Tento nedostatek se podařilo překonat pomocí DC-API, které poskytuje API pro nativní JAVA aplikace pro BOINC. (Marosi, Gombás, Balaton, Kacsuk, 2008)

Přidání a aktualizace BOINC aplikace

Přidání aplikace do projektu vloží informace o aplikaci do databáze. Může také být vytvořena složka, ze které budou do projektu přidávány nové verze aplikací. Po přidání aplikace do projektu je potřeba uvolnit verze pro všechny platformy, které mají být podporovány. (Adding and Updating BOINC applications, 2010)

K dispozici jsou dva způsoby přidání aplikace do projektu. První spočívá v editaci souboru „*project.xml*“ a poté spuštění skriptu „*xadd*“. Druhým způsobem je přidání přes administrační webové rozhraní. (Adding and Updating BOINC applications, 2010)

Pokud přidaná aplikace funguje v samostatném módu, je možné vytvořit verzi pro projekt. K tomu slouží skript „*update_versions*“. Tento program přidá první verzi aplikace a slouží i k přidání nové verze aplikace. Přidaná aplikace musí mít úplné jméno, které obsahuje kromě názvu aplikace také jméno platformy, pro kterou je určena. (Adding and Updating BOINC applications, 2010)

Platforma je většinou kombinací procesorové architektury a operačního systému. Každá verze aplikace musí být spojena s konkrétní platformou. Čím víc platformám projekt podporuje, tím víc počítačů se do projektu může zapojit. (BOINC, 2012)

3.3 Bezpečnost projektu

Při vytváření projektu BOINC je potřeba řešit otázky bezpečnosti. BOINC stanovuje mechanismy, které řeší hlavní problémy pro vývojáře i pro dobrovolníka. Pokud by tyto mechanismy nebyly používány správně, byl by projekt citlivý na řadu útoků. V nejhrošším případě by projekt mohl být využit k šíření škodlivého software na vysoký počet počítačů. (BOINC, 2012)

Projekty by tedy měly provádět pouze organizace, které mají dostatečné odborné znalosti a zdroje pro zajištění serverů. Úspěšně provedený útok by totiž mohl zdiskreditovat všechny BOINC projekty a dobrovolnické počítání obecně. (BOINC, 2012)

3.3.1 Falšování výsledků a pozměňování získaného kreditu

Falšování výsledků probíhá tak, že útočníci vracejí nesprávné výsledky. U pozměňování získaného kreditu pak přihlašují více procesorového času, než opravdu využili. (BOINC, 2012)

Tyto útoky mohou být sníženy pomocí replikace nebo jiné metody ověřování výsledků. BOINC nabízí těchto metod několik, nicméně není jedno doporučené nastavení, volba závisí na požadavcích a povaze aplikací projektu. Každá aplikace by měla mít na straně serveru dva programy, „validátor“ a „asimilátor“. Validátor rozhoduje, zda jsou výsledky správné. Asimilátor se stará o samotné ověření výsledků. (BOINC, 2012)

BOINC podporuje replikaci, což představuje odeslání každé práce na několik klientů a výsledek je považován za platný, pokud ho vrátí naprostá většina klientů. Problémem u replikace může být, že existují rozdíly ve způsobu, jakým různé počítače zpracovávají pohyblivou řádovou čárku. (BOINC, 2012)

Další možností je nepoužívání replikace. Každá práce se udělá jednou a validátor tento výsledek přezkoumá. Tento přístup je užitečný, pokud jsou špatné výsledky nějakým způsobem odhalovány s vysokou pravděpodobností. (BOINC, 2012)

3.3.2 Distribuce škodlivého spustitelného souboru

Útočníci proniknou do BOINC serveru a úpravou databáze a souborů se pokusí o šíření vlastního spustitelného souboru (např. zavirovaný program), který bude maskován za BOINC aplikaci. (BOINC, 2012)

BOINC využívá jako prevenci digitální podepisování kódu. Přestože útočníci proniknou do BOINC serveru, nemohou způsobit, že by klient přijal soubor s falešným kódem. U veřejně dostupných projektů je velice důležité používat správný postup při podepisování kódu. Pokud by totiž útočníci využili BOINC projekt k šíření vlastních programů, mohlo by to vést k zastavení projektu a mělo by to negativní dopad na všechny BOINC projekty. (BOINC, 2012)

3.3.3 Útok na datový server

Při tomto útoku útočníci opakovaně posílají velké soubory na datový server BOINC, který tak zaplní všechny svoje disky a stane se nepoužitelným. (BOINC, 2012)

Pro odepření útoků na datový server poskytuje BOINC volitelný mechanismus nahrávání certifikátů. Každý výstupní soubor má stanovenou maximální velikost. Projekt disponuje dvojicí ověřovacích klíčů pro nahrávání. Veřejný klíč je uložen na projektovém datovém serveru. Popis výsledného souboru je poslán klientům s digitálním podpisem, který je při nahrávání souboru do datového serveru předáván s tímto souborem. Datový server pak ověří popis souboru a zjistí, zda soubor nepřesahuje maximální povolenou velikost. (BOINC, 2012)

3.3.4 Krádež informací z účastnických účtů útokem na server

Při tomto útoku proniknou útočníci do BOINC serveru a z něj ukradnou emailové adresy a další informace z účtů účastníků projektu. (BOINC, 2012)

Projekt by měl řešit tento druh útoků pomocí běžných bezpečnostních postupů. Serverové stroje by měly být chráněny firewallem a nepoužívané síťové služby by měly být vypnuty. Přístup na server by měl být umožněn pouze s šifrovanými protokoly. Doporučuje se také provádět pravidelné bezpečnostní audity všech serverů. (BOINC, 2012)

3.3.5 Krádež informací z účastnických účtů síťovým útokem

K ukradení informací z účtu účastníků je při tomto druhu útoku využíváno BOINC síťových protokolů. (BOINC, 2012)

Útočníci odposlouchávají síťový provoz za účelem získání hesla k uživatelskému účtu. Pokud se jim to podaří, získají přístup k informacím z účtu uživatele a mohou změnit jeho projektové nastavení. BOINC proti těmto útokům nijak nezasahuje. (BOINC, 2012)

3.3.6 Krádež souborů projektu

Útočníci mohou ukrást vstupní i výstupní data projektu. Aplikace BOINC tato data nijak nešifrují. Aplikace by to mohly dělat, ale mělo by to jen malý vliv, protože data se v paměti, kde je k nim snadný přístup pomocí různých programů, nacházejí v podobě textu. (BOINC, 2012)

3.3.7 Úmyslné zneužití účastníků projektem

Projekt úmyslně uvolní aplikaci, která zneužívá účastníka tím, že mu například krade citlivé údaje z osobních souborů. (BOINC, 2012)

BOINC proto využívá účet založený na takzvaném sandboxu. Aplikace běží v rámci speciálně vytvořeného účtu. Pokud jsou k tomuto účtu správně nastavena oprávnění pro soubory a složky, aplikace nemají přístup mimo adresář BOINC. (BOINC, 2012)

3.3.8 Náhodné zneužití účastníků projektem

Projekt uvolní aplikaci, která neúmyslně zneužije účastníkův počítač, například smaže nějaké soubory či způsobí pád systému. (BOINC, 2012)

BOINC některým těmto problémům brání. Když například zjistí, že aplikace používá příliš mnoho paměti, procesoru nebo místa na disku, ukončí ji. Projekt může minimalizovat výskyt těchto problémů dostatečným testováním. Aplikace by měla být otestována na všech platformách a se všemi vstupními daty. (BOINC, 2012)

3.4 Získání a udržení dobrovolníků

Při vytváření projektu dostupného pro dobrovolníky je dobré řídit se následujícími pravidly, která jsou založena na výzkumu mezi účastníky BOINC projektů. (BOINC, 2012)

Angličtina je nejrozšířenějším jazykem mezi účastníky BOINC projektů, proto se doporučuje využívat ji jako hlavní jazyk pro webové materiály a emaily. BOINC poskytuje mechanismy pro překlad webových stránek, který lze svěřit dobrovolníkům. (BOINC, 2012)

3.4.1 Projektové webové stránky

Webové stránky projektu hrají velkou roli při získávání účastníků. Měly by obsahovat:

- představení projektových odborníků, jejich vzdělání a bilanci jejich výzkumu
- popis projektu – vědecké cíle, metody, podrobné údaje o výpočtech prováděných dobrovolníky, jak výzkum ovlivní životy obyčejných lidí
- vlastníka duševního vlastnictví plynoucího z dobrovolných výpočtů, zda bude uvolněno pro veřejnost, kdy a za jakých podmínek
- dosud dosažené výsledky a publikace plynoucí z výzkumu

- informace o členech projektového týmu – osobní informace jako jména, zázemí, zájmy i fotografie (tyto údaje zlidšťují projekt v očích potenciálních účastníků)
 - rychlé reakce na dotazy potenciálních účastníků
 - jasnou navigaci – důležité informace by měly být dostupné z hlavní stránky
 - grafickou identitu projektu (např. logo, barevné schéma), měla by být vidět profesionalita projektu
- (BOINC, 2012)

3.4.2 Propagace

Je potřeba tvrdě pracovat, aby se rozšířilo povědomí o projektu. K tomu nejlépe slouží média jako noviny, časopisy, rozhlas a televize. Lze využít i stávající organizační vztahy. Při práci na univerzitě lze pro práci na projektu využít počítače v laboratořích a počítače studentů a pracovníků. Důležité je zahrnout projekt do seznamu všech projektů distribuovaných výpočtů, které jsou zveřejňovány na různých webových stránkách. (BOINC, 2012)

3.4.3 Emailové mechanismy

BOINC poskytuje nástroje pro zasílání tří typů emailů účastníkům projektu:

- zpravodaj – pravidelný email zasílaný všem účastníkům projektu, obvykle obsahuje dosažené výsledky, plány do budoucna a různá oznámení
 - upomínka – ta je zasílána účastníkům, kteří přestali počítat pro projekt, obvykle jde o krátkou zprávu na podporu zúčastnění výpočtů
 - přítel příteli – jsou zasílány účastníky projektu jejich přátelům, aby jim řekly o projektu a nalákaly je k připojení se do projektu
- (BOINC, 2012)

Efektivní využívání všech těchto typů emailů je rozhodující pro udržení a rozvoj účastnické základny projektu. Při absenci emailů účast na projektu obvykle poklesne o několik procent každý měsíc. BOINC dodává rámce, ale skutečný email je potřeba napsat. (BOINC, 2012)

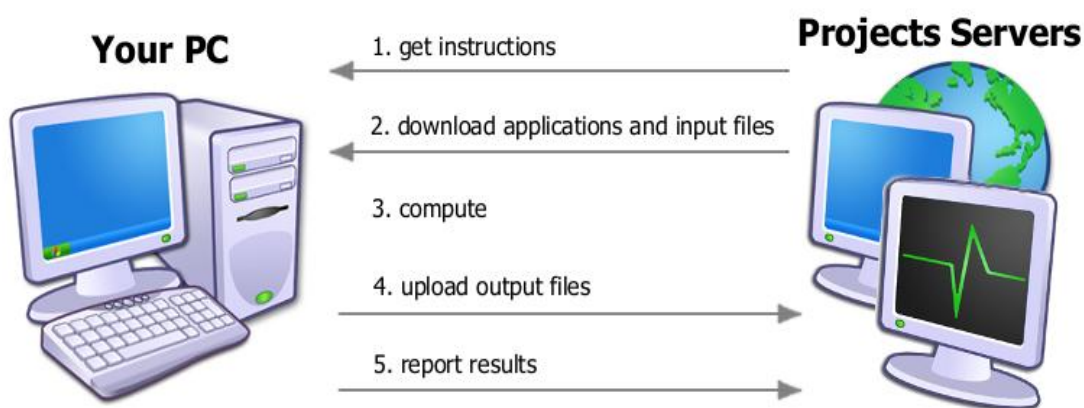
4 Zapojení do projektu z pohledu klienta

Zapojení klienta do projektu je velice jednoduché, stačí stáhnout a nainstalovat BOINC klientský software, vybrat projekt a zadat svoje přihlašovací údaje. (*User manual – BOINC*, 2012, i tato webová adresa v citaci není úplná, užité informace jsou z ní však jednoduše dostupné)

4.1 Komunikace počítače s projektovým serverem

Schéma průběhu komunikace mezi projektovým serverem a klientským počítačem je znázorněno na následujícím obrázku.

Obr. č. 10: Komunikace počítače s projektovým serverem



Zdroj: User manual – BOINC, 2012

Účast počítače v projektu lze rozdělit na pět dílčích úkonů:

1. Počítač obdrží sadu úkolů z projektového plánovače. Úkoly jsou závislé na konfiguraci počítače, neobdrží například úkol, který vyžaduje více operační paměti, než sám má. Projekty mohou podporovat více aplikací a server posílá pouze některou z nich.
2. Počítač stáhne spustitelné a vstupní soubory z projektového datového serveru. Pokud projekt uvolní novou verzi aplikace, spustitelné soubory se stáhnou automaticky.
3. Na počítači běží projektové aplikace produkující výstupní soubory.
4. Počítač nahraje výstupní soubory na datový server projektu.

5. Počítač ohlásí projektovému plánovači splněný úkol a obdrží nové úkoly. Ohlášení plánovači může proběhnout až o několik dní později než nahrání výstupních souborů, závisí to na nastavení preferencí uživatele. (User manual – BOINC, 2012)

Tento cyklus se opakuje neustále, BOINC vše dělá automaticky a účastník projektu do toho nezasahuje. (User manual – BOINC, 2012)

4.2 Pravidla využití BOINC

V této části textu jsou popsány pravidla a strategie platící pro BOINC. Jednotlivé projekty mohou mít ještě další pravidla. (User manual – BOINC, 2012)

4.2.1 Využívejte BOINC na autorizovaných počítačích

Spouštějte BOINC pouze na počítačích, které vlastníte nebo ke kterým jste získali oprávnění od majitele. Některé firmy a školy zakazují využívání BOINC na svých počítačích. (User manual – BOINC, 2012)

4.2.2 Využití počítače programem BOINC

Pokud je na počítači spuštěn BOINC, využívá část výkonu procesoru, místo na disku a internetové připojení. Lze nastavit, jak velkou část těchto zdrojů může BOINC používat a kdy. (User manual – BOINC, 2012)

4.2.3 Ochrana osobních údajů

Aby se zabránilo využití počítače, když uživatel pracuje, BOINC monitoruje aktivitu myši a klávesnice (nezaznamenává stisknuté klávesy), což může spustit nějaký antivirový software. (User manual – BOINC, 2012)

Účet uživatele v projektu je označen názvem, který si sám zvolí. Toto jméno může být uvedeno na webových stránkách projektu spolu s přehledem práce na projektu a ostatních BOINC projektech. Pokud chce uživatel zůstat v anonymitě, zvolí jméno, které neodhaluje jeho identitu. (User manual – BOINC, 2012)

Při účasti na BOINC projektu jsou zaznamenávány informace o počítači (např. typ procesoru, velikost paměti), které slouží k rozhodnutí, jaká práce bude danému počítači přidělena. Tyto informace jsou také uvedeny na webových stránkách projektu, není zde ale uvedeno nic, co by odhalilo umístění počítače. (User manual – BOINC, 2012)

K účasti na projektu je nutné uvést emailovou adresu, na kterou je zaslán email. Tato adresa není zobrazena na webových stránkách projektu a není ani sdílena s nějakou organizací. Projekt může na emailovou adresu zasílat i pravidelné zpravodaje, uživatel si ale může zvolit, zda mu mají být posílány. (User manual – BOINC, 2012)

4.2.4 Odpovědnost

BOINC projekty a Kalifornská univerzita nenesou žádnou odpovědnost za škody na počítači, ztrátu dat či jakékoliv jiné události, které mohou nastat v důsledku účasti v BOINC projektech. (User manual – BOINC, 2012)

4.3 Týmy

Účastníci projektu mohou tvořit týmy. Každý účastník může patřit pouze do jednoho týmu. Tým je skupina stejně smýšlejících lidí, která provádí vědecké výpočty jako kolektiv. Týmy mají často svoji webovou stránku a pomáhají svým členům při nastavení systému, provozují diskusní fórum atd. Jednou z klíčových vlastností je, že skupina má větší šanci hrát nějakou roli ve statistikách než jednotlivec. Kromě toho je větší zábava být součástí skupiny a stýkat se s přáteli. (*Unofficial BOINC Wiki*, 2012)

Pro každý tým je vypočten celkový a průměrný kredit, který odráží veškerou práci vykonanou na projektu účastníky týmu. (*Unofficial BOINC Wiki*, 2012)

Tým má následující atributy: jméno, HTML jméno, textový popis, volitelné URL (např. týmové webové stránky), typ (firma, škola, klub, apod.), jméno zakladatele a seznam členů. Zakladatel týmu může upravovat týmové atributy, odebírat členy z týmu, změnit zakladatele a rozpustit tým. (*Unofficial BOINC Wiki*, 2012)

4.4 Kredit

V předchozím textu již bylo slovo kredit několikrát zmíněno, nyní bude vysvětleno, jaký má toto slovo ve spojení s BOINC význam.

Kredit znamená množství úsilí, kterým uživatel přispěl do BOINC projektu. Je to měřítko, které umožňuje uživatelům monitorovat jejich vlastní příspěvky. (*Unofficial BOINC Wiki*, 2012)

Databáze projektu eviduje, kolik práce počítač udělal. Aby bylo zajištěno správné přidělování kreditu, BOINC projekty postupují následovně: Každá pracovní jednotka může být zaslána na více počítačů. Když počítač vrací výsledek, uplatňuje nárok na

určité množství kreditu podle času využití procesoru. V momentě, kdy byl vrácen dostatek pracovních jednotek, validátor je začne porovnávat. Pokud se výsledky shodují, uživatel obdrží kredit, jehož výše je vypočítána na základě projektových pravidel. (*Unofficial BOINC Wiki*, 2012)

Mezi okamžikem odeslání výsledku a připsání kreditu může být prodleva několik dní. Na stránkách svého účtu proto uživatel vidí i množství kreditu, které nárokuje, ale zatím mu nebylo přiděleno. (*Unofficial BOINC Wiki*, 2012)

4.5 Alternativní zařízení provádějící výpočty

4.5.1 Výpočty pomocí GPU

Většina počítačů je dnes vybavena grafickým procesorem (GPU), který zpracovává jejich grafický výstup. Výpočetní výkon GPU rychle roste a často jsou dnes grafické procesory rychlejší než počítačové procesory (CPU). (User manual – BOINC, 2012)

Některé BOINC projekty nabízejí aplikace běžící na GPU. Tyto aplikace běží až desetkrát rychleji než při použití CPU. U výběru projektu je potřeba nejprve ověřit, zda je podporován uživatelův grafický čip. Některé projekty podporují grafické čipy jen od společnosti ATI, některé od společnosti NVIDIA a některé oboje. (User manual – BOINC, 2012)

4.5.2 Výpočty pomocí Play Station 3

Herní konzole Play Station 3 je i díky velice výkonnému sedmijádrovému procesoru také využívána v BOINC projektech. (*Vykouřil*, 2008)

4.5.3 Výpočty pomocí mobilních zařízení

Vzhledem k rostoucí výpočetní síle mobilních zařízení je zde zájem využít je pro vědecké výpočty. Vývojáři BOINC se zaměřili na operační systém Android, který je založený na Linuxu a je „open source“. (BOINC, 2012)

4.6 Negativní vlivy zapojení do projektu

Zapojení počítače do projektu přináší také negativní vlivy, které je potřeba zmínit.

4.6.1 Zahřívání komponent

Při práci počítače na projektu dochází k většímu zahřívání procesoru i grafické karty, což může způsobovat vyšší hluk i při běžné nenáročné práci uživatele. Toto zahřívání lze regulovat nastavením projektu tak, aby to uživatele nerušilo. Je možné například povolit výpočty pouze v době, kdy počítač není využíván, nebo omezit výkon procesoru při počítání. Tento problém je velmi individuální, někdy nedojde ke zvýšení hlučnosti při vyšším výkonu, někdo zvýšení hluku nevnímá. (VYKOUŘIL, 2009)

4.6.2 Snížení životnosti

Zapojení do projektu vede ke zvýšenému výkonu komponent počítače, hlavně tedy procesoru nebo grafické karty, což snižuje jejich životnost. Vzhledem k dlouhé životnosti těchto komponent by tento negativní vliv neměl mít velkou váhu při rozhodování, zda se zapojit do nějakého projektu. (VYKOUŘIL, 2009)

4.6.3 Zatěžování internetového připojení

Zapojení se do výpočtů zahrnuje i komunikaci s projektovými servery, která probíhá přes internet. Při stahování a odesílání dat tak dochází k zatěžování internetového připojení. Celkově se množství dat za den pohybuje v řádech jednotek MB, takže by to neměl být problém ani v případě připojení s měsíčním limitem na stahovaná data. V programu BOINC Manager je navíc možné omezit rychlost stahování a odesílání dat, čímž lze zatěžování internetového připojení zcela minimalizovat. (VYKOUŘIL, 2009)

4.6.4 Zvýšení spotřeby elektrické energie

Práce na projektu vyžaduje větší zatížení procesoru nebo grafické karty, což vede ke zvýšené spotřebě elektrické energie. Regulace zvýšení spotřeby je možná omezením výpočtů, uživatel se musí rozhodnout, kolik je ochoten do projektu investovat. (VYKOUŘIL, 2009)

5 Modelový distribuovaný výpočet

Pro modelový distribuovaný výpočet jsem využil stolní počítač a notebook. Na notebooku jsem pomocí virtuálního stroje nejprve zprovoznil virtuální server a pomocí domácí sítě jsem se na tento server připojil z domácího počítače. Na notebooku mám nainstalován operační systém Windows XP Service Pack 3 a na domácím počítači linuxovou distribuci Ubuntu 11.10. V následujícím textu je podrobně popsán celý postup od stažení virtuálního serveru až k distribuovanému výpočtu.

K realizování modelového distribuovaného výpočtu jsem využil virtuální server BOINC, jenž lze stáhnout na adrese <http://boinc.berkeley.edu/trac/wiki/VmServer>. Ten jsem spustil v programu VMware Player 4.0.2. od společnosti VMware, který je volně ke stažení na webových stránkách této společnosti.

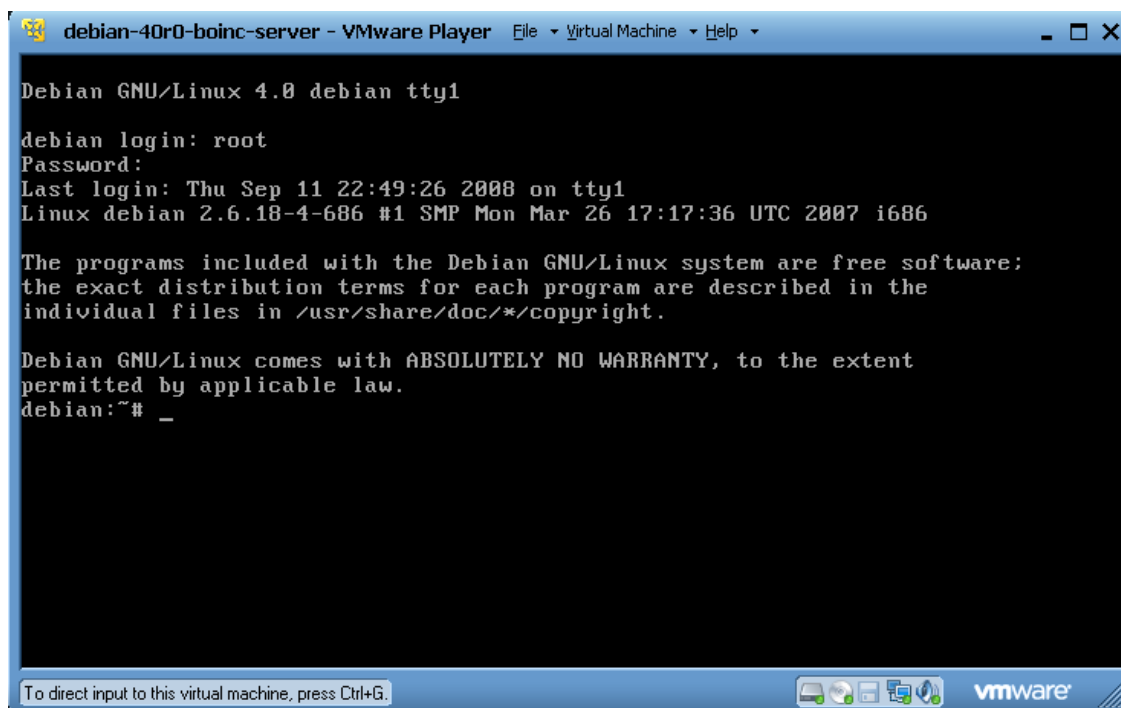
5.1 Serverová část

5.1.1 BOINC virtuální server

Nejprve je nutné stažený soubor s BOINC serverem rozbalit do nějaké složky, ze které ho následně pomocí programu VMware Player načteme. Po načtení je potřeba zadat login a heslo. Použijeme následující údaje: login - „root“, heslo - „rootpw“. Zde stojí za zmínku informace, že při psaní hesla se na obrazovce nezobrazují žádné znaky, jako je to běžné ve většině systémů, kde se místo znaků zobrazují hvězdičky. Ihned po prvním přihlášení je vhodné heslo změnit, zvláště pak, pokud bude možno na server přistupovat z internetu.

Na následujícím obrázku je vidět prostředí virtuálního stroje i s programem VMware Player, ve kterém tento stroj běží. Na dalších obrázcích už bude pouze prostředí virtuálního stroje.

Obr. č. 11: Virtuální stroj po přihlášení



Zdroj: vlastní zpracování, 2012

Jak je možno vidět na obrázku, virtuální stroj pracuje v systému Debian GNU/Linux 4.0. Proto je nutné při nastavování serveru a aplikací využívat příkazy právě pro tento systém. Jednotlivé příkazy budu pro zvýšení přehlednosti textu uvádět do rámečku vždy na novou řádku.

Nastavení síťového připojení

Před vytvořením projektu je potřeba ve virtuálním stroji nastavit síťové připojení. Nastavení musí proběhnout jak ve VMware Playeru, tak v samotném systému Debian. V dalším textu je popsáno nastavení síťového připojení pro lokální síť.

Připojení ve VMware Playeru se nastaví pomocí nastavení síťového adaptéru, kde se zaškrtně „Bridged“.

K nastavení sítě v Debianu je potřeba použít několik příkazů. Nejprve je nutno vymazat soubor „z25_persistent-net.rules“ z adresáře „rules.d“. To provedeme pomocí následujícího příkazu:

```
rm /etc/udev/rules.d/z25_persistent-net.rules
```

Po provedení příkazu se na obrazovce neobjeví otázka, zda chceme soubor opravdu smazat, jako je tomu u Windows, a neobjeví se ani žádné potvrzení o smazání.

Dále je potřeba nastavit IP adresu a další náležitosti, což se dělá v souboru „*interfaces*“ v adresáři `network`.

```
vi /etc/network/interfaces
```

V tomto souboru upravíme IP adresu, masku sítě a můžeme zadat další parametry pro síťové připojení. Na obrázku číslo 11 lze vidět strukturu souboru „*interfaces*“ a konkrétní hodnoty nastavení sítě. Po provedení změn stiskneme klávesu Esc a pro uložení změn zadáme:

```
:wq
```

Obr. č. 11: Nastavení síťového připojení

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
#allow-hotplug eth0
#iface eth0 inet dhcp
auto eth0
iface eth0 inet static
address 192.168.1.36
netmask 255.255.255.0
dns-search ssl.berkeley.edu
dns-nameservers 192.168.1.36_
```

Zdroj: vlastní zpracování, 2012

Aby se projevíly změny, je potřeba restartovat síťové připojení zadáním příkazu:

```
/etc/init.d/networking restart
```

Nyní je nastaveno síťové připojení a lze přejít k vytvoření projektu.

5.1.2 Vytvoření projektu

Pro vytvoření projektu je nutné přihlásit se jako jiný uživatel. Zadáme tedy příkaz:

```
login
```

Jako `login` zadáme „*boincadm*“ a heslo „*boincadmpw*“. Opět je vhodné heslo po přihlášení změnit.

K vytvoření projektu slouží v BOINC skript *make_project*, který již byl popsán v podkapitole 3.2.1. V tomto případě, pro vytvoření projektu s názvem „*Diplomová*“

práce“, se nejprve dostaneme do adresáře *tools* , ve kterém se nachází skript *make_project* a poté jej spolu se zadáním parametrů vytvářeného projektu spustíme.

```
cd ~/boinc/tools  
./make_project --url_base http://192.168.1.36 diplomova_prace
```

Během instalace se na obrazovku vypisují základní informace o vytvářeném projektu, jejichž správnost potvrdíme zadáním „Y“, následně je potřeba ještě jednou zadat „Y“ a poté se již vytváření projektu automaticky dokončí. Pokud vše proběhlo správně, na obrazovce se objeví informace: „Done creating project“. Celý proces vytváření projektu lze vidět na následujícím obrázku.

Obr. č. 12: Skript *make_project*

```
boincadm@debian:~/boinc/tools$ ./make_project --url_base http://192.168.1.36 diplomova_prace  
Creating project 'diplomova_prace' (short name 'diplomova_prace'):  
  PROJECT_ROOT = /home/boincadm/projects/diplomova_prace/  
  URL_BASE = http://192.168.1.36/  
  HTML_USER_URL = http://192.168.1.36/diplomova_prace/  
  HTML_OPS_URL = http://192.168.1.36/diplomova_prace_ops/  
  CGI_URL = http://192.168.1.36/diplomova_prace_cgi/  
  KEY_DIR = /home/boincadm/projects/diplomova_prace/keys/  
  DB_NAME = diplomova_prace  
  DB_HOST =  
  
Continue? [Y/n] Y  
Setting up server: creating directories  
  
Keys don't exist in /home/boincadm/projects/diplomova_prace/keys/: generate them?  
 [Y/n] Y  
Setting up server files: generating keys  
Setting up server files: copying files  
Setting up database  
Setting up server files: writing config files  
Setting up server files: linking cgi programs  
update_translations finished  
Done installing default daemons.  
  
Done creating project. Please view  
/home/boincadm/projects/diplomova_prace/diplomova_prace.readme  
for important additional instructions.
```

Zdroj: vlastní zpracování, 2012

5.1.3 Přidání aplikace

Nyní již zbývá do projektu přidat aplikaci. V jednom projektu může být více aplikací, ale v této práci je použita pouze jediná. Zde je potřeba upozornit na fakt, že pro každý typ operačního systému musí být jiná aplikace. V tomto případě to znamená, že projekt bude podporovat pouze jeden druh operačního systému. Z důvodu toho, že klientský počítač v mé domácí síti běží pod operačním systémem Linux, bude přidána aplikace pro Linux.

Popis aplikace

K vytvoření aplikace jsem využil program „Hello, World“ z webových stránek <http://www.spy-hill.net/help/boinc/hello.html>. Jedná se o program, který otevře soubor „*out.txt*“, něco do něj zapíše a odešle ho na server.

Vše potřebné pro vytvoření aplikace „Hello World“ je ke stažení na webových stránkách uvedených výše. Je zde několik verzí aplikace. Já jsem použil dosud nejnovější verzi programu s označením 6.12, která je zkomprimována v souboru pojmenovaném „*hello-6.12.tgz*“. Pro vytvoření aplikace je z tohoto souboru využít zdrojový kód, který je uložen v souboru „*hello.C*“ a soubor „*Makefile*“.

Zdrojový kód

Zdrojový kód v souboru „*hello.C*“ jsem upravil tak, že nyní do souboru „*out.txt*“ zapíše dvě náhodná čísla oddělená čárkou a mezerou a ty odešle na server. Celý upravený kód se nachází v příloze A.

Zde bych chtěl ukázat a okomentovat pouze několik zajímavých částí kódu:

- **Pracovní smyčka:**

```
int j, num, N;
N = 123456789;
for ( j=0 ; j<N ; j++ ){
    num=rand()+rand();
}
```

Tento cyklus 123 456 789 krát sečte dvě náhodná čísla, která ani nikam nezapisuje. Účelem této smyčky je zatížit na nějaký čas procesor klientského počítače, aby byly za průběh aplikace přičteny klientskému počítači kredity.

- **Zapsání dvou náhodných čísel:**

```
srand(time(NULL));
cislo1=rand();
cislo2=rand();
fprintf(f, "%i, %i \n", cislo1, cislo2);
```

Tato část kódu zapíše do souboru *out.txt* dvě náhodná čísla. Důležitý je příkaz `srand(time(NULL))`, který pokud by chyběl, pokaždé bychom dostali stejná čísla.

Makefile

V souboru „*Makefile*“ je potřeba provést několik úprav. Pokud zavádíme aplikaci do projektu poprvé, není třeba měnit číslo `MINOR_VERSION`, které udává verzi aplikace. Pokud však již aplikaci v projektu máme a přidáváme novější, či jen upravenou verzi, je potřeba toto číslo zvýšit. V originálním „*Makefile*“ je u `MINOR_VERSION` číslo 12, v upraveném „*Makefile*“ jsem toto číslo zvýšil na 13, protože jsem v aplikaci udělal několik změn.

Vzhledem k tomu, že „*Makefile*“ je připraven i pro vytvoření aplikace pro operační systém Mac, jsou zde v komentářích příkazy, které se použijí pro vytvoření aplikace právě pro tento operační systém.

Dále je potřeba zadat cestu do složky, kde se nachází BOINC. K tomu slouží příkazy `BOINC_BUILD` a `BOINC_PREFIX`. Pro server ve virtuálním stroji má cesta do složky s BOINC následující podobu: „`/home/boincadm/boinc`“.

V další části souboru je na výběr z varianty A a B, kde je potřeba jednu variantu odkomentovat. Varianta A umožňuje vyzkoušení aplikace ještě před nasazením na server, proto jsem využil právě ji.

Z důvodu problémů s průběhem při překladu zdrojového kódu jsem z „*Makefile*“ vymazal všechny komentáře. Ty jsou dostupné v originálním souboru, takže pokud by někdo komentáře vyžadoval, lze je snadno dohledat. Proto se v příloze B nachází originální a v příloze C upravená verze souboru „*Makefile*“.

Samotné přidání aplikace do projektu

Pro přidání aplikace do projektu je nejprve nutné dostat soubory „*Makefile*“ a „*hello.C*“ do virtuální mašiny. To je možné přes CD či pomocí ISO souboru. Já jsem zvolil druhou možnost.

V nastavení VMware Playeru v záložce hardware u nastavení „`CD/DVD`“ zvolíme možnost „*Use ISO image file*“ a vybereme ISO soubor, ve kterém jsou potřebná data.

Dále se již postupuje ve virtuální mašině. Nejprve je nutné se odhlásit jako uživatel „`boincadm`“, což lze jednoduše pomocí příkazu:

logout

Nyní je aktivní znovu uživatel „root“. Tento krok by nebyl nutný, pokud bychom přidělili uživateli „boincadm“ práva pro práci se soubory a pro editaci databáze.

K procházení souborů na CD/DVD či jako v tomto případě v ISO souboru je v Debianu potřeba nejprve připojit mechaniku. K tomu slouží následující příkaz:

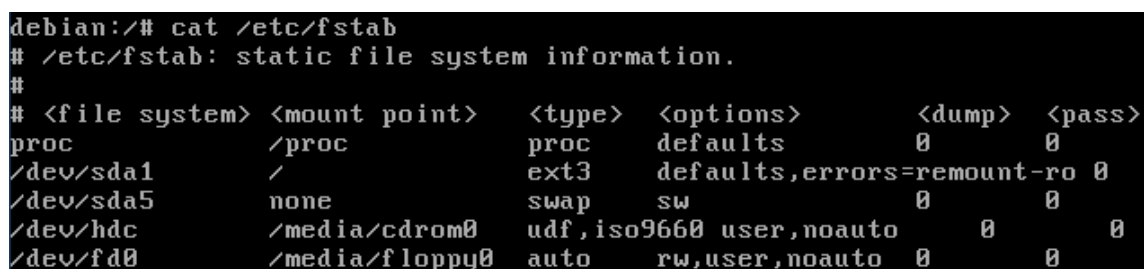
```
mount /dev/hdc /media/cdrom0
```

Pokud by tento příkaz nefungoval, je nutné zjistit přesné označení a cestu k mechanice, k čemuž slouží následující příkaz, který vypíše všechny mechaniky a disky, které systém detekoval:

```
cat /etc/fstab
```

Ukázka výpisu ve virtuální mašině je zachycena na následujícím obrázku. Můžeme si z obrázku ověřit, že cesta k mechanice je v předešlém příkazu správná.

Obr. č. 14: Výpis disků a mechanik ve virtuální mašině



```
debian:~# cat /etc/fstab
# /etc/fstab: static file system information.
#
# <file system> <mount point> <type> <options> <dump> <pass>
proc /proc proc defaults 0 0
/dev/sda1 / ext3 defaults,errors=remount-ro 0
/dev/sda5 none swap sw 0 0
/dev/hdc /media/cdrom0 udf,iso9660 user,noauto 0 0
/dev/fd0 /media/floppy0 auto rw,user,noauto 0 0
```

Zdroj: vlastní zpracování, 2012

Nyní se tedy dostaneme do ISO souboru a překopírujeme soubory „Makefile“ a „hello.C“ do složky „apps“ v adresáři „BOINC“. Vzhledem k tomu, že se oba soubory nalézají ve složce se jménem „Hello“, překopírujeme celou tuto složku:

```
cd /media/cdrom0
cp -r Hello /home/boincadm/boinc/apps
```

Následně je potřeba přeložit zdrojový kód „hello.C“ do souboru. To lze provést přímo v právě nakopírované složce „Hello“ pomocí příkazu:

```
make
```

Po úspěšném vykonání tohoto příkazu se ve složce „Hello“ objeví soubor „hello_6.13_i686-pc-linux-gnu“, což je aplikace pro přidání do projektu. Správnost průběhu aplikace lze už nyní vyzkoušet zadáním příkazu:

```
./hello
```

Je třeba počkat na dokončení průběhu aplikace a poté již můžeme v souboru „*out.txt*“ zkontrolovat, zda aplikace opravdu do tohoto souboru uložila dvě náhodná čísla oddělená čárkou a mezerou. K přečtení souboru „*out.txt*“ lze využít například následující příkaz:

```
cat out.txt
```

V BOINC je nutné dodržovat přesně daná pojmenování aplikací i souborů, ve kterých se tyto aplikace nacházejí. Pro přidání aplikace „Hello, World“ do projektu je nejprve nutné ve složce „apps“ v adresáři projektu vytvořit novou složku s názvem aplikace. V této složce pak vytvořit další složku, jejíž název se shoduje s úplným názvem aplikace, tedy i s označením verze a operačního systému:

```
mkdir hello
```

```
mkdir hello_6.13_i686-pc-linux-gnu
```

Do tohoto nově vytvořeného adresáře nyní nakopírujeme soubor „*hello_6.13_i686-pc-linux-gnu*“:

```
cp hello_6.13_i686-pc-linux-gnu /home/boincadm/projects/diplomova_prace/apps/hello/hello_6.13_i686-pc-linux-gnu
```

Dalším krokem při přidání aplikace do projektu je úprava souboru „*project.xml*“, který se nachází ve složce projektu. Do tohoto souboru je potřeba přidat následující údaje:

```
<app>
  <name>hello</name>
  <user_friendly_name>Hello world</user_friendly_name>
</app>
```

Za tagem „name“ musí být jméno aplikace shodné s názvem složky, ve které je aplikace uložena ve složce „apps“ v adresáři projektu, kdežto za tagem „user_friendly_name“ může být prakticky cokoliv. Přidání je umožněno po zadání příkazu:

```
vi project.xml
```


Následující obrázek ukazuje část souboru „*project.xml*“ s nově přidanými údaji. Pro ukončení modifikace stiskneme klávesu Esc a pro uložení změn zadáme:

```
:wq
```

Obr. č. 15: Úprava „*project.xml*“

```
<platform>
  <name>sparc-sun-solaris</name>
  <user_friendly_name>Solaris 2.8 or later running on a SPARC-compatible C
PU</user_friendly_name>
</platform>
<platform>
  <name>sparc64-sun-solaris</name>
  <user_friendly_name>Solaris 2.8 or later running on a SPARC 64-bit CPU</
user_friendly_name>
</platform>
<platform>
  <name>powerpc64-ps3-linux-gnu</name>
  <user_friendly_name>Sony Playstation 3 running Linux</user_friendly_name
>
</platform>
<platform>
  <name>anonymous</name>
  <user_friendly_name>anonymous</user_friendly_name>
</platform>
<app>
  <name>uppercase</name>
  <user_friendly_name>upperCASE</user_friendly_name>
</app>
<app>
  <name>hello</name>
  <user_friendly_name>Hello world</user_friendly_name>
</app>
</boinc>
```

Zdroj: vlastní zpracování, 2012

Následující příkazy dokončí přidání nové aplikace do projektu:

```
bin/xadd
```

```
bin/update_versions
```

Po zadání druhého příkazu je potřeba dvakrát potvrdit pokračování zadáním „y“. Pokud proces proběhne úspěšně, na obrazovce se objeví nápis „*Done*“. Průběh skriptu „*update_versions*“ ukazuje následující obrázek.

Obr. č. 16: Průběh skriptu „update_versions“

```
debian:/home/boincadm/projects/diplomova_prace# bin/update_versions
  Found <App#2 hello> version 613 for <Platform#3 i686-pc-linux-gnu>: hello_6.13_i686-pc-linux-gnu

SECURITY WARNING:
=====

You have not provided a signature file for /home/boincadm/projects/diplomova_prace/apps/hello/hello_6.13_i686-pc-linux-gnu/hello_6.13_i686-pc-linux-gnu.

I can generate one now, but this is highly unrecommended. Generating code signatures on network-connected computers is a security vulnerability, and should not be done for publicly-accessible projects.

Continue with automatically generating a code signature? [y/N] y
Signing /home/boincadm/projects/diplomova_prace/apps/hello/hello_6.13_i686-pc-linux-gnu/hello_6.13_i686-pc-linux-gnu
Copying hello_6.13_i686-pc-linux-gnu to /home/boincadm/projects/diplomova_prace/download/hello_6.13_i686-pc-linux-gnu
Ready to commit 1 items:
  <AppVersion#None hello 613 i686-pc-linux-gnu>
Continue [Y/n] Y
Committed:
  <AppVersion#2 hello 613 i686-pc-linux-gnu>
Touched trigger file to make feeder re-read app_version table from database
Done
```

Zdroj: vlastní zpracování, 2012

Pro aplikaci je ještě potřeba přidat vstupní a výstupní šablonu. Šablony jsou ve formátu XML, vstupní má název „world_wu.xml“ a výstupní „hello_re.xml“. Pro tuto aplikaci poslouží neupravené verze šablon, které jsou přiloženy v souboru „hello-6.12.tgz“. Jejich přenos do virtuální mašiny proběhl stejně jako u zdrojového kódu a „Makefile“, tedy přes ISO soubor. Obě šablony je potřeba nakopírovat do složky „templates“ v adresáři projektu:

```
cp world_wu.xml /home/boincadm/projects/diplomova_prace/templates
cp hello_re.xml /home/boincadm/projects/diplomova_prace/templates
```

Po přenesení šablon je aplikace do projektu plně přidána a může dojít ke spuštění projektu.

5.1.4 Spuštění projektu

Ještě před samotným spuštěním je nutné projekt přidat na webový server, což zprovozní webové stránky projektu a administrátorské webové rozhraní. Úpravě stránek se věnuje podkapitola 5.3, prozatím zůstanou stránky v podobě, ve které byly vygenerovány skriptem „make_project“. Přidání na webový server provedeme pomocí následujících příkazů, které zadáme v adresáři projektu:

```
su -c 'cat diplomova_prace.httpd.conf >> /etc/apache2/httpd.conf'
su -c 'apache2ctl -k restart'
```

Aby bylo možné využít webové rozhraní pro administraci projektu, je potřeba přidat uživatele a jeho heslo. Pro naši aplikaci postačí přidat jednoho uživatele s možností administrace projektu, u větších projektů těchto uživatelů bude jistě několik. Pro přidání je třeba vstoupit do složky „ops“, která se nachází v adresáři „html“, který je umístěn v adresáři projektu. Přidání uživatele boincadm proběhne pomocí příkazu:

```
htpasswd -c .htpasswd boincadm
```

Po zadání příkazu se systém dotáže na nové heslo, které je pro kontrolu nutné zadat dvakrát.

Nyní již můžeme spustit projekt. Pro spuštění stačí v adresáři projektu zadat příkaz:

```
bin/start
```

Tímto krokem jsme spustili projekt a lze se k němu z klienta přihlásit a registrovat. Webové stránky projektu jsou již také funkční na webové adrese http://192.168.1.36/diplomova_prace. K přístupu do webového rozhraní pro administraci projektu slouží adresa http://192.168.1.36/diplomova_prace_ops, která je ale dostupná až po zadání přístupových údajů.

V projektu zatím není vytvořena žádná pracovní jednotka, takže klient po připojení do projektu nepracuje. Vytvoření pracovní jednotky pro aplikaci „Hello, World“ provedeme ručně, u větších projektů se o to stará většinou nějaký program.

Vytvoření jedné jednotky práce proběhne po zadání následujícího příkazu v adresáři projektu:

```
./bin/create_work --appname hello --wu_name prvni --wu_template
templates/world_wu.xml --result_template templates/hello_re.xml
```

Tento příkaz vytvoří pracovní jednotku s názvem „prvni“. Při vytváření dalších jednotek, je vždy nutné tento název změnit, vytvoření více pracovních jednotek se stejným názvem není v BOINC možné.

5.1.5 Výsledky

Asi nejdůležitější částí projektu jsou výsledky, bez nich by práce na projektu neměla smysl. Na serveru se uchovávají pouze úspěšně ověřené výsledky. Ty se nachází v podsložce projektového adresáře s názvem „upload“. O pojmenování a hierarchické struktuře adresáře „upload“ pojednává podkapitola 3.2.2. Ve složce „upload“ se nachází tolik složek, kolik je zde výsledků. V každé této složce je uložen soubor, který obsahuje dvě čísla, tedy výsledek aplikace „hello“. Na následujícím obrázku jsou pro názornost tři tyto výsledky zobrazeny.

Obr. č. 17: Výsledky

```
debian:/home/boincadm/projects/diplomova_prace/upload/18d# cat treti_2_0
628219686, 29883995
debian:/home/boincadm/projects/diplomova_prace/upload/1e2# cat druhy_2_0
177683837, 1834843621
debian:/home/boincadm/projects/diplomova_prace/upload/389# cat ctvrty_3_0
2872274924, 349891774
```

Zdroj: vlastní zpracování, 2012

5.1.6 Webové rozhraní pro administraci

BOINC nabízí pro administrátory projektu webové rozhraní. V předchozí podkapitole byly nastaveny přístupové údaje pro uživatele „boincadm“ a nyní se lze pomocí těchto údajů do webového rozhraní přihlásit pomocí webového prohlížeče zadáním adresy http://192.168.1.36/diplomova_prace_ops.

Webové stránky pro administraci obsahují místo názvu projektu nápis „REPLACE WITH PROJECT NAME“. Vzhledem k využití pro administrátora a s ohledem na jediný existující projekt není nutné tyto stránky nijak upravovat. V podkapitole 5.3 bude nicméně ukázáno, jak se dá tento text nahradit názvem projektu. Na následujícím obrázku je zachycena hlavní stránka webového rozhraní pro administraci projektu před upravením názvu projektu.

Webové rozhraní pro administraci umožňuje procházet databázi projektu. Jsou dostupné informace o podporovaných platformách, o aplikacích a jejich verzích, o uživatelích, týmech, o zařízeních podílejících se na výpočtu, o pracovních jednotkách, výsledcích a o tabulkách v databázi a jejím využití místa na disku. V databázi se dá přes webové rozhraní velice dobře vyhledávat, na výběr je mnoho parametrů prohledávání.

Obr. č. 18: Webové rozhraní pro administraci projektu

REPLACE WITH PROJECT NAME: Project Management

- Currently used SVN revision: 20737 ;
Warning: fopen() [function.fopen]: php_network_getaddresses: getaddrinfo failed: Temporary failure in name resolution in /home/boincad/index.php on line 61
- Warning:** fopen(http://boinc.berkeley.edu/svn/) [function.fopen]: failed to open stream: Connection refused in /home/boincad/ Can't get latest SVN revision
- There are 0 remaining candidates for User of the Day.

Browse database:	Regular Operations:	Special Operations:
<ul style="list-style-type: none">• Platforms• Applications• Application versions• Users (recently registered)• Teams• Hosts• Workunits• Results• DB row counts and disk usage	<ul style="list-style-type: none">• Screen user profiles• Manage special users	<ul style="list-style-type: none">• Manage applications• Manage application versions• Send mass email to a selected set of users• Email user with misconfigured host• FLOP count statistics• Cancel workunits• Manage user ID: <input type="text"/>• Clear Host: <input type="text"/> <input type="button" value="Clear RPC"/>

Result summary for uppercase:

- Past 24 hours: [summary](#) | [pass percentage by platform](#) | [failure by host](#) | [failure by platform](#)
- Past 7 days: [summary](#) | [pass percentage by platform](#) | [failure by host](#) | [failure by platform](#)

Result summary for hello:

- Past 24 hours: [summary](#) | [pass percentage by platform](#) | [failure by host](#) | [failure by platform](#)
- Past 7 days: [summary](#) | [pass percentage by platform](#) | [failure by host](#) | [failure by platform](#)

[Show deprecated applications](#)

Zdroj: vlastní zpracování, 2012

Dále je přes webové rozhraní pro administraci projektu umožněno spravovat uživatelské profily a „speciální“ uživatele (moderátor na fóru, administrátor projektu, vývojář projektu, tester projektu, dobrovolný vývojář, dobrovolný tester a vědecký pracovník projektu). Lze rozeslat hromadný email všem účastníkům projektu či jen vybraným skupinám. Je také umožněno uživateli pozastavit přístup k projektu na určitou dobu dlouhou od jedné hodiny do dvou týdnů. K pozastavení lze připojit důvod, proč byl uživateli zamezen přístup k projektu.

Dalšími možnostmi webového rozhraní jsou správa aplikací a jejich verzí v projektu, rušení pracovních jednotek, zobrazení nejnovějších výsledků a analýza neúspěchů.

Webové rozhraní ve výchozím nastavení sice není nijak graficky lákavé, je ale přehledné a svůj účel rozhodně plní. Hodně jsem ocenil, že se na každé stránce vyskytuje odkaz pro návrat na hlavní stránku.

5.2 Klientská část

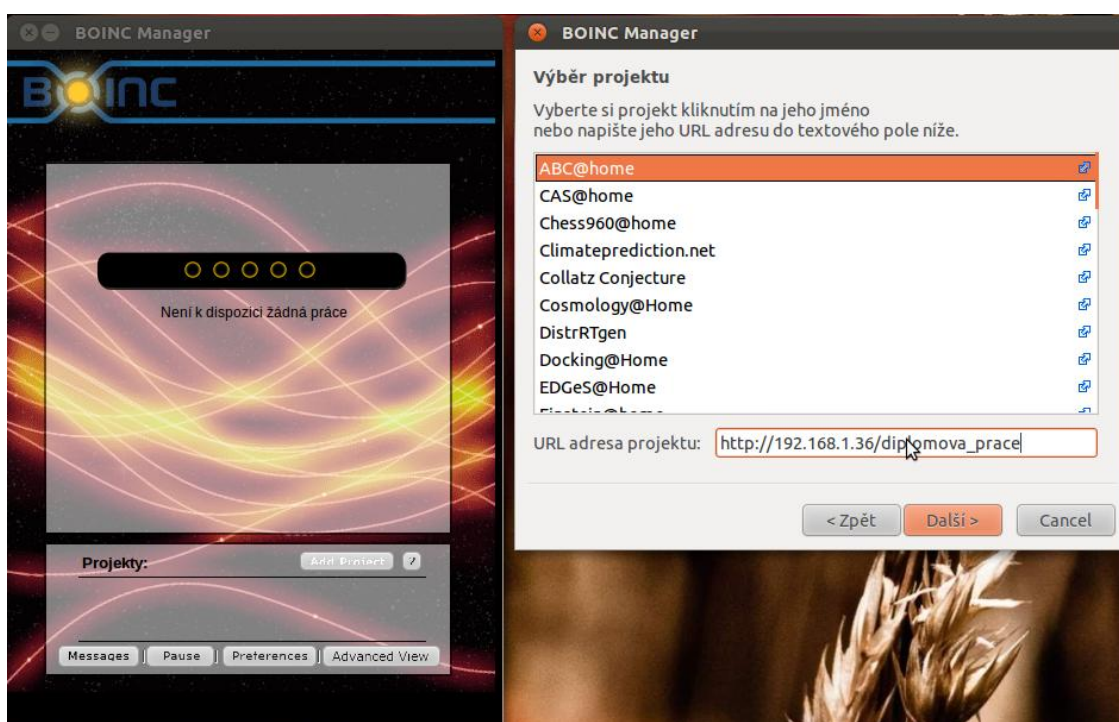
Klientskou část programu jsem spouštěl na osobním počítači s operačním systémem Ubuntu 11.10. Na tomto počítači bylo nejprve nutné nainstalovat klientský program, který se stará o komunikaci se serverem. Instalaci jsem provedl v programu Ubuntu

Software Center. Využil jsem připravený balík obsahující BOINC klienta a manažera. Veškeré nastavení a připojení jsem tudíž prováděl v programu BOINC Manager, který je navíc v češtině.

5.2.1 Připojení k projektu

Po spuštění programu BOINC Manger se lze připojit k projektu kliknutím na políčko „Add Project“. To vyvolá nabídku, kde se uživatel dozví, že je k dispozici více než 30 BOINC projektů a projekt lze přidat buď přímo, nebo s využitím Správce účtu. Pro připojení k vytvořenému projektu s názvem „Diplomová práce“ je nutné zvolit přidání přímo. Tímto se uživatel dostane k samotnému výběru z projektů, kde lze vybírat z nabídky, či pomocí zadání URL adresy projektu, jak ukazuje následující obrázek. K připojení k projektu „Diplomová práce“ je potřeba zadat URL adresu projektu: http://192.168.1.36/diplomova_prace.

Obr. č. 19: Výběr projektu



Zdroj: vlastní zpracování, 2012

Po zadání URL adresy se klient přihlásí k projektu. Nyní je nutné vytvořit účet zadáním emailové adresy a hesla. Pro dokončení nastavení účtu je uživatel přesměrován na stránky projektu, kde zadá své jméno (lze použít přezdívku) a pokud chce i zemi a poštovní směrovací číslo. Dále je mu umožněno najít a připojit se k nějakému týmu.

Když uživatel úspěšně dokončí toto nastavení, je přesměrován na webovou stránku svého účtu, kde ho může spravovat a kde se zobrazuje jím dosažený kredit na projektu.

5.2.2 Nastavení účtu

Kromě změny osobních údajů a hesla může uživatel na webové stránce svého účtu upravit výpočetní nastavení BOINC, nastavení daného projektu či komunitní nastavení.

Výpočetní nastavení BOINC je rozděleno do tří částí, kterými jsou využití procesoru, využití internetu a využití disku a paměti. Toto nastavení je shodné s nastavením v programu BOINC Manager, které je popsáno v následující podkapitole.

Komunitní nastavení obsahuje následující možnosti:

- Informace o soukromých zprávách, o přátelstvích a o příspěvcích ve zvoleném fóru lze obdržet třemi způsoby – na stránce svého účtu, okamžitě emailem či v jednom denním emailu.
- Lze nahrát svého avatara (obrázek, který představuje uživatele), využít služeb Gravatar.com a nebo avatara vůbec nepoužívat.
- Je možné zadat podpis, který se zobrazí pod každou vaší zprávou na fóru. Podpis může mít maximálně 250 znaků a lze využít služeb, které generují dynamický obrázek například s informacemi o získaném kreditu, projektových novinkách atd.
- Při zobrazování zpráv lze skrýt avatary, skrýt podpisy, zvýraznit speciální uživatele, použít obrázky jako odkazy na uživatele a otevírat odkazy v novém okně nebo záložce.
- Vlákna lze zobrazovat podle čtyř možností řazení, jako první může být – nejnovější odpověď, nejsledovanější, s nejvíce odpověďmi nebo nejnovější.
- Příspěvky v každém vláknu mohou být řazeny od nejstaršího, od nejnovějšího nebo od nejlépe hodnoceného.
- Zajímavou možností představuje filtrování zpráv, kdy po zadání ID uživatele se nezobrazují jeho příspěvky.

Nastavení projektu je možné změnit v následujících oblastech:

- Rozdělení zdrojů – určuje poměr prostředků přidělených tomuto projektu (důležité, pokud je uživatel připojen k více projektům).

- Posílání emailů – umožňuje uživateli zakázat, aby mu byly posílány emaily z projektu či z jeho týmu (pokud nějaký má).
- Webové stránky projektu – uživatel rozhoduje, zda může být jeho počítač zobrazen na stránkách projektu.
- Umístění počítače – nepovinné, na výběr umístění doma, v práci či ve škole.
- Využití CPU pro grafiku – procentní nastavení využití procesoru pro grafické úkoly v projektu.

Po provedení nastavení projektu se uživatel nemusí o nic starat, vše probíhá automaticky.

5.2.3 BOINC Manager

V této části práce se zaměřím na popis a ukázkou možností programu BOINC Manager v operačním systému Ubuntu, který je připojen k projektu „Diplomová práce“. Připojení k projektu je popsáno v podkapitole 5.2.1, proto už se mu zde nebudu věnovat a ukážu jen možnosti nastavení a informace o projektu, ke kterému je program připojený.

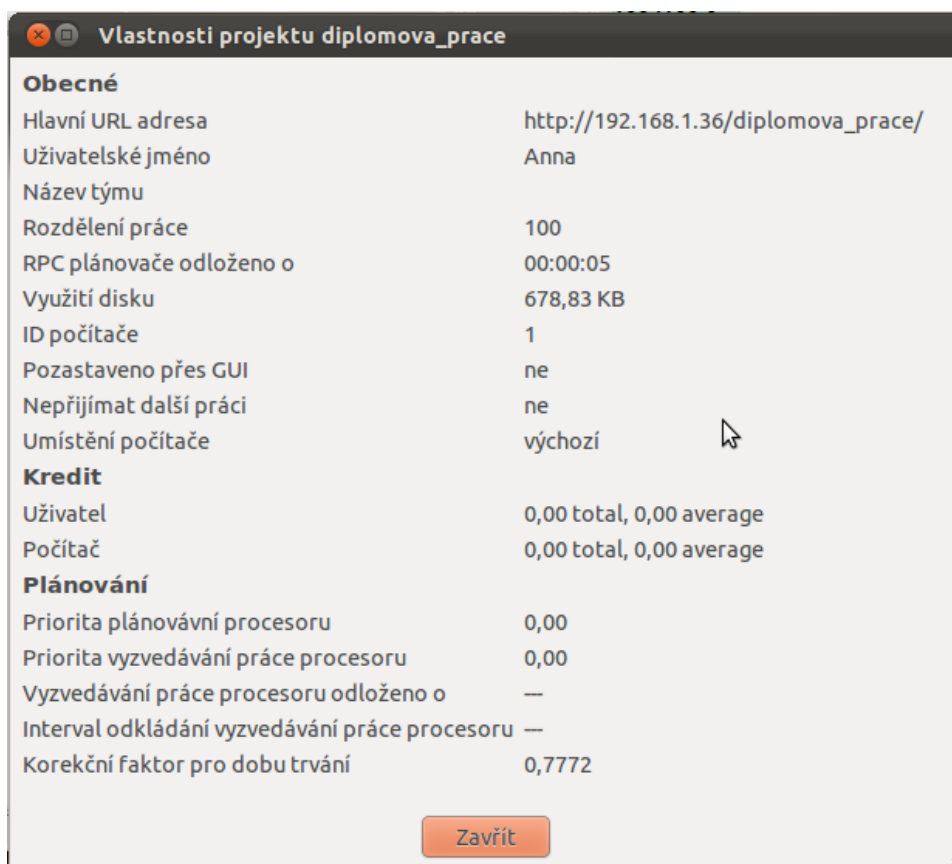
BOINC Manager nabízí dvě možnosti zobrazení – jednoduché a rozšířené. Jednoduché zobrazení je zjednodušením toho rozšířeného a nabízí nižší počet možností nastavení a méně informací o probíhajících úkolech. Vzhled jednoduchého zobrazení je možno vidět na obrázku číslo 19 při výběru projektu.

Rozšířené zobrazení zde popíši podrobněji. Okno BOINC Manageru v rozšířeném zobrazení obsahuje šest záložek – zprávy, projekty, úkoly, přenos, statistiky a disk.

Záložka „Zprávy“ obsahuje obdržené zprávy buďto z projektu či od jiného uživatele. V případě připojení k projektu „Diplomová práce“ nejsou dostupné žádné zprávy.

V záložce „Projekty“ lze daný projekt aktualizovat, pozastavit, restartovat nebo odebrat. Dále je zde umožněno vypnout přijímání další práce z projektu a zobrazit vlastnosti projektu. Vlastnosti projektu „Diplomová práce“ jsou na následujícím obrázku.

Obr. č. 20: Vlastnosti projektu v programu BOINC Manager



Zdroj: vlastní zpracování, 2012

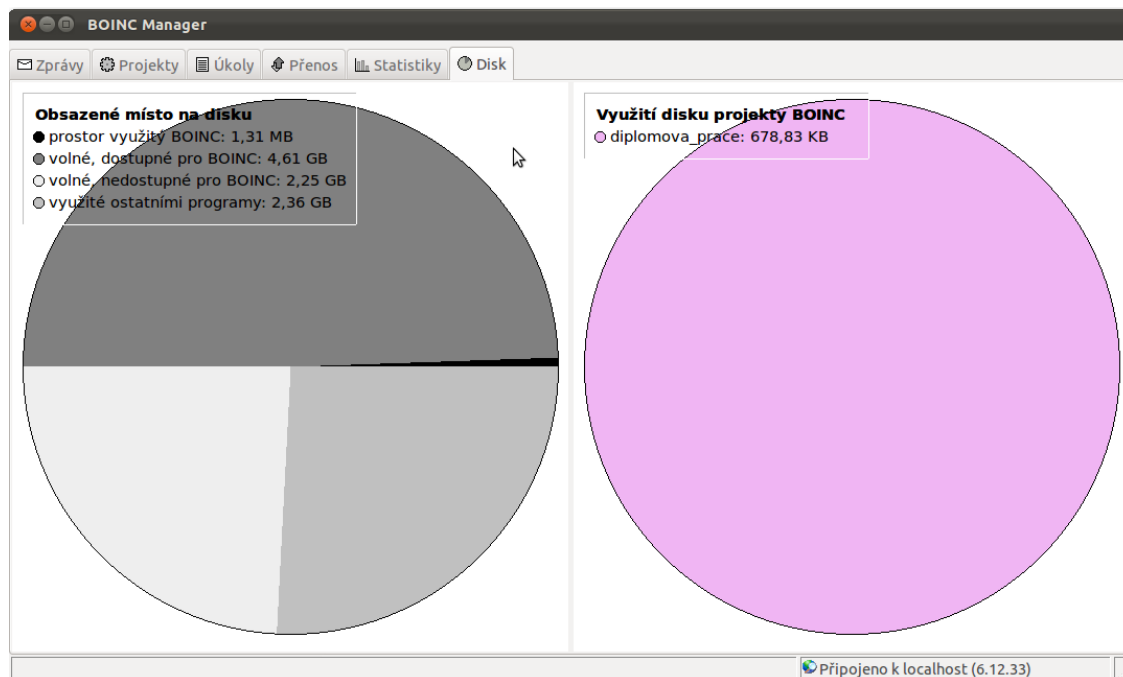
Jednotlivé stažené a rozpracované pracovní jednotky jsou zobrazeny v záložce „Úkoly“. O každé pracovní jednotce se zde nacházejí informace o jménu, aplikaci, projektu, průběhu, stavu, zbývajícím čase a informace, do kdy musí být pracovní jednotka odeslána na server.

Záložka „Přenos“ obsahuje přehled všech souborů, které čekají na stažení nebo odeslání, informaci o jejich velikosti a o průběhu procesů s těmito soubory.

V záložce „Statistiky“ je graficky vyobrazen stav kreditu uživatele a kreditu počítače. Jeden uživatel totiž může k výpočtům využívat více počítačů a statistiky v BOINC Manageru zobrazují jak stav kreditu uživatele, tak i stav kreditu jednotlivých počítačů.

Grafické vyobrazení je možné najít také v záložce „Disk“. Zde se nachází dva výšečové grafy, z nichž jeden znázorňuje obsazené místo na disku a druhý využití disku projekty BOINC. Vzhled těchto grafů v programu BOINC Manager připojeném pouze k projektu „Diplomová práce“ je na následujícím obrázku.

Obr. č. 21: Záložka „Disk“ v programu BOINC Manager



Zdroj: vlastní zpracování, 2012

Nastavení samotného BOINC Manageru probíhá v menu „předvolby“ a to ve třech záložkách – využití procesoru, využití Internetu a využití disku a paměti. Toto nastavení platí pro aktuální počítač, na kterém je manager spuštěn.

Záložka „Využití procesoru“ nabízí následující možnosti:

- povolit počítání
 - při provozu na baterie, je-li počítač používán
 - povolit využití GPU pro výpočty, i když je počítač právě používán
 - pouze pokud je počítač nečinný po určitou dobu
 - při vytížení procesoru na méně než zadaný počet procent (pokud je zadána nula, počítá se bez omezení)
 - každý den v určitých hodinách (pokud jsou časy shodné, počítá se bez omezení)
 - pouze pro vybrané dny v týdnu
- přepínat mezi aplikacemi po určité době
- na víceprocesorových systémech použít nejvýše určité procento všech procesorů
- použít nejvýše zadané procento procesorového času

Ze všech těchto možností je zřejmé, že při správném nastavení počítání projektů BOINC nemusí uživatele nijak omezovat při práci.

V záložce „Využití Internetu“ je možné předvolit:

- maximální rychlost stahování i odesílání
- omezení množství dat stažených za den
- periodicitu připojení se na server projektu
- množství zásoby pracovních jednotek
- zobrazení dotazu před připojením k Internetu
- odpojení po ukončení komunikace
- povolení přístupu k Internetu pouze v určitý čas a pouze v určité dny

Nastavení množství místa na disku a využití paměti se nachází v záložce „Využití disku a paměti“. Pro disk lze nadefinovat:

- maximální množství dat uložených na disku a minimální velikost místa, které musí na disku zůstat volné
- maximální procentuální využití celkového místa na disku
- čas, za který se může kontrolní bod ukládat na disk
- procentuální využití stránkovacího souboru

U využití paměti je možné zadat množství využívané paměti v případě, kdy je nebo není počítač používán. Lze také zvolit možnost, kdy při pozastavení výpočtů zůstane aplikace ponechána v paměti.

Výčet možností nadefinování BOINC Managera je dlouhý, ale pokud si uživatel není s něčím jist, může ponechat výchozí hodnoty nastavení.

5.3 Nastavení stránek projektu

Webové stránky projektu jsou velice důležitou součástí správy projektu. Vzhledem k tomu, že projekt „Diplomová práce“ není určen pro dobrovolnické počítání a počítače zapojené do projektu mají přesné informace, není potřeba o stránky nějak pečlivě dbát. Přesto si myslím, že alespoň hlavní stránka projektu by mohla poskytovat základní údaje o projektu.

Obsahem a možnostmi webových stránek projektu se zabývá podkapitola 3.2.5, zde se zaměříme pouze na nastavení jména projektu a upravení hlavní stránky.

Nastavení jména projektu provedeme v adresáři projektu, ve složce „html“ a její podsložce „project“. Následujícím příkazem se dostaneme k úpravám souboru „*project.inc*“:

```
vi project.inc
```

V tomto souboru ve třetím řádku zaměníme za „REPLACE WITH PROJECT NAME“ jméno projektu, tedy „Diplomová práce“ (ve virtuální mašině je to nutné zadat bez diakritiky). Pro ukončení editace souboru a uložení změn stiskneme tlačítko Esc a zadáme:

```
:wq
```

Tímto jednoduchým krokem se nastaví správné jméno projektu napříč celým webovým rozhraním projektu.

Nastavení hlavní stránky projektu se nachází v souboru „*index.php*“, který je uložen v adresáři projektu ve složce „html“ a podsložce „user“. Pro úpravu tohoto souboru zadáme příkaz:

```
vi index.php
```

Po provedení požadovaných úprav je pro jejich uložení nutné stisknout tlačítko Esc a zadat:

```
:wq
```

Úprav v tomto souboru jsem provedl několik, i když ne nějak zásadních. Přeložil jsem text hlavní stránky do češtiny a zadal krátkou informaci o projektu - Diplomová práce je projekt vytvořený pro provedení modelového distribuovaného výpočtu.

Na následujícím obrázku je zachycena upravená hlavní stránka projektu.

Obr. č. 22: Upravená hlavní stránka projektu

Diplomova prace

O Diplomova prace

Diplomova prace je projekt vytvoreny pro provedeni modeloveho distribuovaneho vypoctu.

Diplomova prace

- [Přečtěte si naše pravidla a zásady](#)
- Tento projekt využívá BOINC. Pokud používáte BOINC, zvolte Připojit k projektu. Když ne, [stáhněte si BOINC](#).
- Po vyzvání, zadejte http://192.168.1.36/diplomova_prace/
- Pokud využíváte příkazovou řádku nebo pre-5.0 verzi BOINC, [vytvořte si účet](#).
- V případě problému, [hledejte pomoc zde](#).

Ucastnici

- [Vas ucet](#) - statistiky, zmena preferenci
- [Tymy](#) - vytvor nebo se pripoj k tymu
- [Certifikat](#)
- [Aplicačni programy](#)

Komunita

- [Profily](#)
- [Vyhledavani uzivatelu](#)
- [Diskuzní fóra](#)
- [Otázky a odpovědi](#)
- [Statistiky](#)
- [Jazyky](#)



Novinky

No news forum. Run `html/ops/create_forums.php`.

Zdroj: vlastní zpracování, 2012

Pro porovnání se můžete vrátit do podkapitoly 3.2.5 na obrázek číslo 9, kde má tato hlavní stránka ještě výchozím vzhled.

6 Projekty distribuovaného počítání

V této části práce jsou popsány dosažené výsledky distribuovaných výpočtů, některé dobrovolnické projekty v systému BOINC a také projekty běžící pod jinými programy.

6.1 Dosud získané výsledky distribuovaných výpočtů

Když se někdo rozhoduje, zda se dobrovolně zapojí do distribuovaných výpočtů, zajímají ho dosud získané výsledky.

Projekt LHC@home již přinesl přímé výsledky v podobě spuštění největšího urychlovače částic na světě ve Švýcarském středisku CERN. Pomocí výpočtů tohoto projektu docházelo k postupnému usazování a kalibrování každého z 1300 gigantických magnetů. Na stavbě tohoto urychlovače se tak podílelo celkem 53 703 lidí na světě, z toho 1658 Čechů. Projekt LHC@home je stále aktivní, práci ovšem uvolňuje jen nahodile pro propočty nových nastavení, jako reakci na momentální potřeby urychlovače. (Vykouřil, 2012)

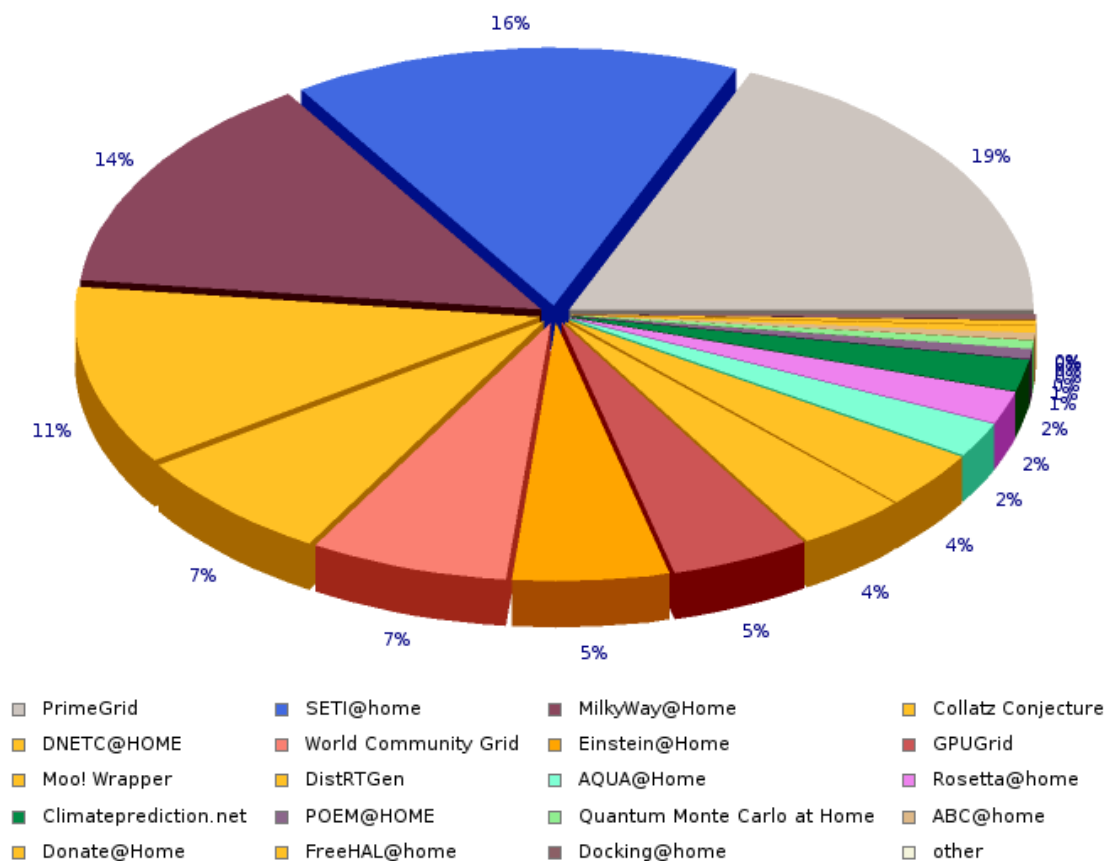
Další projekty mohou na výsledky čekat několik let. Některé přináší dílčí výsledky, například projekty na výzkum klimatu, které zpracovávají předpovědi vývoje na několik desítek let dopředu. Také mnoho biologických projektů již pomohlo vědcům převážně hledáním prostorového uspořádání molekul bílkovin. U vesmírných projektů dochází k průběžnému doplňování znalostí o vesmíru. Tyto projekty mapují oběžné dráhy objektů, které by se mohly srazit se Zemí, zabývají se výzkumem Mléčné dráhy či kosmickými mikrovlnami. (Vykouřil, 2009)

6.2 BOINC projekty

V předešlé podkapitole již byla uvedena jména některých projektů. Zde je několik BOINC projektů více popsáno, ke každému projektu je uvedena jeho webová stránka, podporované platformy a popis činnosti projektu.

Na následujícím obrázku je graf zobrazující rozdělení kreditu podle projektů k 23. 4. 2012. Prvních osm projektů s nejvíce kredity je popsáno v následujícím textu.

Obr. č. 23: BOINC projekty podle rozdělení kreditu



Zdroj: Boinc all Project Stats, 2012

6.2.1 PrimeGrid

Stránky projektu: <http://www.primegrid.com>

Primárním cílem projektu PrimeGrid je poskytnout i běžnému uživateli možnost hledat prvočísla. Dalším cílem projektu je poskytnout široké veřejnosti vzdělávací materiály o prvočíslech a díky nalezeným prvočíslym pomoci v různých oblastech vědy. Prvočísla například hrají velkou roli v kryptografických systémech, které jsou používány k zabezpečení počítačů. Prostřednictvím studie prvočísel lze zjistit, jak moc výpočetního výkonu je potřeba k prolomení šifrovaného kódu a zda jsou různé bezpečnostní systémy dostatečně bezpečné. (Czech National Team, 2012)

Po přihlášení do projektu si může uživatel vybrat, jaké prvočíslo bude hledat. Na výběr má patnáct podprojektů, z nichž se každý zabývá hledáním jiného prvočísla. (Czech National Team, 2012)

Tento projekt již zaznamenal řadu dílčích úspěchů, kdy například 9. 4. 2012 bylo nalezeno dosud největší prvočíslo Sophie Germainové, které je 200 701 číslic dlouhé. (PrimeGrid, 2012)

Kromě operačních systémů Windows, Linux, MacOS a Solaris jsou projektem podporovány výpočty s využitím grafických karet NVIDIA a ATI a také herní konzole Play Station 3. (Choosing BOINC projects, 2012)

6.2.2 SETI@home

Stránky projektu: <http://setiathome.berkeley.edu>

SETI@home uvedl mezi distribuované výpočty David Anderson, který v roce 2003 stál u zrodu celého systému BOINC. Projekt v současné době nabízí dva podprojekty – SETI@home Enhanced a Astropulse. (Czech National Team, 2012)

Cílem SETI@home Enhanced je objevit signály umělého původu, které by potvrdily přítomnost mimozemské civilizace. K zachycení signálu je využíván největší radioteleskop na světě, který se nachází v Portoriku a má průměr 306 metrů. Signály zachycené tímto teleskopem se skládají převážně z radiového šumu, který je tvořen signály z pulzarů a kvazarů, televizním a satelitním vysíláním, případně radarovou navigací. Analýza signálu proto vyžaduje obrovský výpočetní výkon, který se s příchodem distribuovaných výpočtů stal dostupný. (Czech National Team, 2012)

Astropulse hledá také mimozemský signál, ovšem na rozdíl od SETI@home Enhanced hledá silné pulzy, které by pokryly široké pásmo. Takovýto pulz by při cestování mezihvězdným prostorem kolidoval s hmotou, což by vedlo k relativnímu zpomalení nižších frekvencí oproti vyšším v efektu zvaném disperze. Pokud by bylo známo, kolik disperzí signál obdržel, bylo by možné vrátit ho do původní podoby. Protože to ale není známo, musí se brát v úvahu všechny možnosti, což vyžaduje obrovský výpočetní výkon. (Czech National Team, 2012)

Projekt objevil za dobu své existence několik desítek potenciálních signálů, které by potvrdily existenci mimozemské civilizace. Ten správný důkaz ale prozatím nalezen nebyl. (Czech National Team, 2012)

Výpočty na tomto projektu je možné provádět pomocí grafických karet NVIDIA a kromě Windows, Linux a MacOS je podporován i operační systém Solaris. (Choosing BOINC projects, 2012)

6.2.3 MilkyWay@home

Stránky projektu: <http://milkyway.cs.rpi.edu/milkyway>

Projekt MilkyWay@home vytváří pomocí distribuovaných výpočtů vysoce přesný trojrozměrný model Mléčné dráhy. (MilkyWay@home, 2012)

K projektu se lze připojit z operačních systémů Windows, Linux a MacOS. Jsou podporovány také výpočty s využitím grafických karet NVIDIA a ATI. (Czech National Team, 2012)

6.2.4 Collatz Conjecture

Stránky projektu: <http://boinc.thesonntags.com/collatz>

Jedná se o matematický projekt, který se zabývá metodou $3x+1$, nazývanou také Collatz conjecture. Metoda se zabývá rozkladem čísel. Pokud je číslo dělitelné dvěma s celočíselným výsledkem, je vyděleno, jinak je vynásobeno třemi a je k němu přičtena jednička. Tento postup se opakuje, dokud není výsledkem číslo jedna. Projekt se snaží ke každému číslu najít počet operací a zároveň potvrdit, že jde takto rozložit jakékoliv číslo. (Czech National Team, 2012)

Výpočty na tomto projektu je možné provádět pomocí grafických karet ATI i NVIDIA a na běžných operačních systémech, kterými jsou Windows, Linux a MacOS. (Czech National Team, 2012)

6.2.5 DNETC@home

Tento projekt zajišťoval zapojení se do počítání na projektu distributed.net (je popsán v podkapitole 6.3.1). Nyní již neběží a byl nahrazen projektem Moo! Wrapper. (Czech National Team, 2012)

6.2.6 World Community Grid

Stránky projektu: <http://www.worldcommunitygrid.org>

Tento projekt zpracovává několik podprojektů a uživatel si může sám určit, do kterého se chce zapojit. K 23. 4. 2012 bylo aktivních těchto dvanáct podprojektů:

- Computing for Sustainable Water má za cíl studovat vliv lidské činnosti na rozlehlém rozvodí řeky a získat tím hlubší poznatky, která opatření mohou vést k obnovení a udržitelnosti tohoto významného vodního zdroje.
- Say No to Schistosoma hledá potenciální kandidáty léků, které by mohly pomoci při léčbě schistosomózy.
- GO Fight Against Malaria objevuje nadějně kandidáty léčiv, ze kterých by mohly být vyvinuty nové léky, které by léčily dosud proti lékům odolné druhy malárie.
- Drug Search for Leishmaniasis identifikuje potenciální kandidáty molekul, ze kterých by mohl být vyroben lék na leishmaniózu. Hledají se sloučeniny chemických látek a bílkovin, které by vedly k výrobě efektivního léku proti této nemoci.
- Computing for Clean Water má za cíl poskytnout hlubší pohled na molekulární úroveň vzniku efektivního průtoku vody přes nové třídy filtračních materiálů.
- The Clean Energy Project - Phase 2 má za poslání najít nové materiály pro další generaci solárních článků a později pro ukládání energie.
- Discovering Dengue Drugs – Together – Phase 2 se zabývá výzkumem horečky Dengue a pomáhá při hledání nových léků. Tyto nové léky by zároveň mohly být využity i při léčbě hepatitidy C, viru Západního Nilu a žluté horečky, které patří do stejné skupiny virů.
- Help Cure Muscular Dystrophy - Phase 2 svůj výzkum zaměřil na více než 2200 bílkovin, o nichž se vědci domnívají, že hrají významnou roli při neuromuskulárních onemocněních. Tento výzkum by měl vést k lepšímu způsobu léčby svalové dystrofie a jiných neuromuskulárních onemocnění.
- Help Fight Childhood Cancer hledá léky, které by omezily nebo úplně vyřadily bílkoviny spojené s rakovinou neuroblastoma, což je jeden z nejčastěji se vyskytujících nádorů u dětí.
- Help Conquer Cancer má za cíl získat dokonalejší výsledky z rentgenové krystalografie bílkovin. Tento projekt by měl pomoci i k rozšíření znalostí o vzniku rakoviny a zlepšit postupy pro léčení.

- Human Proteome Folding - Phase 2 má za cíl získat struktury některých lidských a patogenních proteinů s vyšším rozlišením a rozvinout hranice předpovědi proteinové struktury.
- FightAIDS@Home je výzkum v oblasti léků proti nemoci AIDS.
(World Community Grid, 2012)

Kromě Windows, Linux a MacOS je podporován také operační systém FreeBSD.
(Choosing BOINC projects, 2012)

6.2.7 Einstein@home

Stránky projektu: <http://einstein.phys.uwm.edu/>

Tento projekt se snaží zaznamenat a v další fázi zkoumat gravitační vlny ve vesmíru. Využívá k tomu několik interferometrů obrovských rozměrů. Existenci gravitačních vln předpověděl Albert Einstein a na konci dvacátého století se ji podařilo nepřímo ověřit, ale stále chybí přímý důkaz. Díky zkoumání těchto vln by bylo možné se podívat do historie vesmíru. (Czech National Team, 2012)

Projekt umožňuje využít k výpočtům pouze grafické karty NVIDIA . Podporovány jsou operační systémy Windows, Linux i MacOS. (Czech National Team, 2012)

6.2.8 GPU GRID

Stránky projektu: <http://www.gpugrid.net/>

Projekt zaměřující se na výpočty v oblasti biologie a biomedicíny. Výpočty probíhají výhradně s aplikacemi pro grafické karty. Pro počítání na tomto projektu je potřeba vlastnit grafickou kartu značky NVIDIA, pro grafické karty ATI zatím běží pouze testovací aplikace. (Czech National Team, 2012)

6.3 Ostatní projekty

V oblasti distribuovaných výpočtů existují i projekty, které nevyužívají systém BOINC. Některé z nich jsou popsány v dalším textu.

6.3.1 *Distributed.net*

Stránky projektu: http://www.distributed.net/Main_Page

Tento projekt se zabývá matematickými výpočty a v současné době nabízí dva podprojekty RC5-72 a Optimal 27-Mark Golomb Rulers. (Czech National Team, 2012)

RC5-72 navazuje na úspěšně dokončené projekty RC5-56 a RC5-64 a pokouší se prolomit 72 bitovou variantu šifrovacího algoritmu RC5 hrubou silou. (Czech National Team, 2012)

Optimal 27-Mark Golomb Rulers hledá optimální Golombovo pravítko, tedy pravítko, na kterém jsou rysky rozloženy tak, že žádná vzdálenost dvou rysek není na pravítku vícekrát. Optimální Golombovo pravítko má nejkratší možnou vzdálenost vzhledem k počtu rysek. (Czech National Team, 2012)

6.3.2 *Folding@home*

Stránky projektu: <http://folding.stanford.edu/>

Projekt Folding@Home patří mezi nejznámější a také nejvýkonnější vědecké projekty distribuovaných výpočtů. Výzkum tohoto projektu je zaměřen na tvorbu a rozpad bílkovin a měl by sloužit k nalezení léků na Alzheimerovu a Parkinsonovu chorobu, některé druhy rakoviny a lidskou formu BSE. (Czech National Team, 2012)

Projekt jako první nabídl možnost výpočtů pomocí grafických karet a později umožnil zapojení do výpočtů i pro PlayStation 3. (Czech National Team, 2012)

6.3.3 *Wieferich@home*

Stránky projektu: <http://www.elmath.org/>

Tento projekt vznikl v roce 2008 a jedná se o první český projekt využívající dobrovolnické distribuované výpočty. Jeho cílem je najít třetí Wieferichovo prvočíslo. (Czech National Team, 2012)

Závěr

Hlavním cílem této diplomové práce bylo provést modelový distribuovaný výpočet. Prvním úkolem při řešení byl výběr „open source“ softwarového nástroje. Po analýze dostupných možností jsem zvolil systém BOINC. Celý zdrojový kód tohoto systému je k dispozici a navíc se tento program těší velké oblibě u účastníků na projektech distribuovaných výpočtů. BOINC je tedy volně dostupný, navíc pokud by došlo k rozšíření projektu a využití výpočtů od dobrovolníků, lze oslovit velkou základnu aktivních účastníků používajících tento systém.

Dalším cílem práce bylo provedení samotného distribuovaného výpočtu. K tomu jsem využil již hotovou aplikaci, kterou jsem upravil, aby vyhovovala potřebám této práce. Celý proces nastavení serveru, vytvoření projektu, přidání aplikace, připojení k projektu a provedení výpočtu je v práci podrobně popsán, což tvoří velký přínos této práce. Samotná aplikace je sice jednoduchá, ale nejlepší způsob, jak se naučit psát vlastní aplikace, je právě využití již hotových příkladů.

Přínos práce spatřuji také v uvedení všech aspektů, které se zapojením do projektů pod systémem BOINC souvisí. V práci jsou popsány bezpečnostní opatření systému, způsoby ochrany osobních údajů i negativní vlivy, které zapojení do projektu přináší. V závěru práce je popsáno několik aktuálně běžících projektů distribuovaného počítání, které působí v mnoha oblastech vědecké činnosti. Každý, kdo by chtěl poskytnout svůj výpočetní výkon, si tak může i díky této práci vybrat, který projekt mu je nejbližší.

Seznam obrázků

- Obr. č. 1: Typický distribuovaný systém – str. 10
- Obr. č. 2: Interakce softwarových komponent na každém procesoru – str. 11
- Obr. č. 3: Přenosná a kapesní zařízení v distribuovaném systému – str. 12
- Obr. č. 4: Cloud computing – str. 16
- Obr. č. 5: Logo BOINC – str. 20
- Obr. č. 6: Architektura systému BOINC – str. 22
- Obr. č. 7: Adresářová struktura projektu – str. 26
- Obr. č. 8: Konfigurační soubor projektu – str. 28
- Obr. č. 9: Výchozí webové stránky projektu – str. 29
- Obr. č. 10: Komunikace počítače s projektovým serverem – str. 36
- Obr. č. 11: Virtuální stroj po přihlášení – str. 42
- Obr. č. 13: Nastavení síťového připojení – str. 43
- Obr. č. 14: Skript `make_project` – str. 44
- Obr. č. 14: Výpis disků a mechanik ve virtuální mašině – str. 47
- Obr. č. 15: Úprava „`project.xml`“ – str. 49
- Obr. č. 16: Průběh skriptu „`update_versions`“ – str. 50
- Obr. č. 17: Výsledky – str. 52
- Obr. č. 18: Webové rozhraní pro administraci projektu – str. 53
- Obr. č. 19: Výběr projektu – str. 54
- Obr. č. 20: Vlastnosti projektu v programu BOINC Manager – str. 57
- Obr. č. 21: Záložka „Disk“ v programu BOINC Manager – str. 58
- Obr. č. 22: Upravená hlavní stránka projektu – str. 61
- Obr. č. 23: BOINC projekty podle rozdělení kreditu – str. 63

Seznam použitých zkratk

BOINC - Berkeley Open Infrastructure for Network Computing

WAN - Wide Area Network

LAN - Local Area Network

www - World Wide Web

P2P - peer to peer

GIMPS - Great Internet Mersenne Prime Search

SETI - Search for Extraterrestrial Intelligence

CGI - Common Gateway Interface

PID - process identifier

GPU - graphic processing unit

CPU - central processing unit

Seznam použité literatury

Adding and Updating BOINC applications. [online]. [cit. 16. 4. 2012]. Datum poslední aktualizace 15. 7. 2010. Dostupné na [www: http://www.spy-hill.com/~myers/help/boinc/app_add.html](http://www.spy-hill.com/~myers/help/boinc/app_add.html)

ANDERSON, P. David. *BOINC: A System for Public-Resource Computing and Storage.* [online]. [cit. 1. 4. 2012]. Dostupné na [www: http://boinc.berkeley.edu/grid_paper_04.pdf](http://boinc.berkeley.edu/grid_paper_04.pdf)

BOINC. [online]. [cit. 3. 4. 2012]. Dostupné na [www: http://boinc.berkeley.edu/trac/wiki](http://boinc.berkeley.edu/trac/wiki)

Boinc all Project Stats. [online]. [cit. 23. 4. 2012]. Dostupné na [www: http://us.allprojectstats.com/po.php?projekt=0](http://us.allprojectstats.com/po.php?projekt=0)

COLOURIS, George, DOLLIMORE, Jean, KINDBERG, Tim, BLAIR, Gordon *Distributed Systems: Concepts and Design.* 5. vyd., Addison-Wesley, 2011, 1008 s., ISBN 978-0-13-214301-1

Creating and Configuring a BOINC Project. [online]. [cit. 17. 4. 2012]. Datum poslední aktualizace 1. 1. 2012. Dostupné na [www: http://www.spy-hill.com/~myers/help/boinc/Create_Project.html#custom](http://www.spy-hill.com/~myers/help/boinc/Create_Project.html#custom)

Czech National Team. [online]. [cit. 20. 4. 2012]. Dostupné na [www: http://projekty.czechnationalteam.cz/](http://projekty.czechnationalteam.cz/)

Hello, BOINC World!. [online]. [cit. 10. 4. 2012]. Datum poslední aktualizace 30. 9. 2011. Dostupné na [www: http://www.spy-hill.net/help/boinc/hello.html](http://www.spy-hill.net/help/boinc/hello.html)

Choosing BOINC projects. [online]. [cit. 23. 4. 2012]. Datum poslední aktualizace: 30. 1. 2012. Dostupné na [www: http://boinc.berkeley.edu/projects.php](http://boinc.berkeley.edu/projects.php)

Index of /pub/meyers/src/boinc. [online]. [cit. 10. 4. 2012]. Dostupné na [www: http://www.spy-hill.net/pub/myers/src/boinc/](http://www.spy-hill.net/pub/myers/src/boinc/)

KSHEMKALYANI, D. Ajay, SINGHAL, Mukesh *Distributed Computing: Principles, Algorithms, and Systems.* Cambridge: Cambridge University Press, 2008, 736 s., ISBN 978-0-521-87634-6

MAROSI, C., Attila, GOMBÁS, Gábor, BALATON, Zoltán, KACSUK Péter. *Enabling Java applications for BOINC with DC-API*. In: KACSUK, Péter, LOVAS, Róbert, NÉMETH, Zsolt. *Distributed and Parallel Systems: In Focus: Desktop Grid Computing*. Springer, 2008, 220 s., ISBN 978-0-387-79447-1

MilkyWay@home. [online]. [cit. 21. 4. 2012]. Dostupné na [www: http://milkyway.cs.rpi.edu/milkyway/](http://milkyway.cs.rpi.edu/milkyway/)

NEVES, Rodrigo, MESTRE, Nuno, MACHADO, Francisco, LOPES Joao. *Parallel and Distributed Computing BOINC Grid Implementation*. [online]. [cit:1. 4. 2012]. Dostupné na [www: http://www.deei.fct.ualg.pt/~a27981/spd/boinc-grid.pdf](http://www.deei.fct.ualg.pt/~a27981/spd/boinc-grid.pdf)

PrimeGrid. [online]. [cit. 21. 4. 2012]. Dostupné na [www: http://www.primegrid.com/](http://www.primegrid.com/)

Unofficial BOINC Wiki. [online]. [cit. 2. 4. 2012]. Dostupné na [www: http://www.boinc-wiki.info/Main_Page](http://www.boinc-wiki.info/Main_Page)

User manual – BOINC. [online]. [cit. 10. 4. 2012]. Dostupné na [www: http://boinc.berkeley.edu/wiki/User_manual](http://boinc.berkeley.edu/wiki/User_manual)

VYKOUŘIL, Dušan. *BOINC - počítače všech zemí, spojte se 1/2* [online]. [cit. 3. 4. 2012]. Datum poslední aktualizace 25. 8. 2008. Dostupné na [www: http://pctuning.tyden.cz/software/jak-zkrotit-internet/11449-boinc-pocitace_vsech_zemi_spojte_se_12?start=1](http://pctuning.tyden.cz/software/jak-zkrotit-internet/11449-boinc-pocitace_vsech_zemi_spojte_se_12?start=1)

VYKOUŘIL, Dušan. *LHC@home – Jak Češi pomáhali budovat z tepla svého domova částicový urychlovač LHC*. [online]. [cit. 20. 4. 2012]. Dostupné na [www: http://www.czechnationalteam.cz/view.php?navezclanku=jak-cesi-pomahali-budovat-z-tepla-sveho-domova-casticovy-urychlovac-lhc&cislocclanku=2008070001](http://www.czechnationalteam.cz/view.php?navezclanku=jak-cesi-pomahali-budovat-z-tepla-sveho-domova-casticovy-urychlovac-lhc&cislocclanku=2008070001)

World Community Grid. [online]. [cit. 23. 4. 2012]. Dostupné na [www: http://www.worldcommunitygrid.org](http://www.worldcommunitygrid.org)

Seznam příloh

Příloha A: Zdrojový kód hello.C

Příloha B: Originální Makefile

Příloha C: Upravený Makefile

Příloha A: Zdrojový kód hello.C

```
/*
 * Hello, BOINC World!                                     Version: 6.12
 *
 * This is the Hello World program for BOINC. It is the simplest application
 * one can write which uses the BOINC API and writes some output to a
 * file (called "out.txt"). See the sample workunit and result templates
 * to see how this file is mapped to a real file and how it is uploaded.
 *
 * Note: if you want to run this program "standalone" then the file "out.txt"
 * must already exist!
 *
 * For more information see the release notes at
 * http://www.spy-hill.com/help/boinc/hello.html
 *
 * Eric Myers <myers@spy-hill.net> - 16 June 2004 (Unix)/6 July 2004 (Windows)
 * @(#) $Revision: 1.23 $ - $Date: 2011/09/30 20:05:37 $
 */
#ifdef WIN32
#include "boinc_win.h"
#include "util.h"
#endif

/* BOINC API */

#include "boinc_api.h"
#include "diagnostics.h" // boinc_init_diagnostics()
#include "filesystem.h" // boinc_fopen(), etc...
#include "str_util.h" // for parse_command_line()

// version.h is only used to get BOINC version numbers
#include "version.h"

/* Begin: */

int main(int argc, char **argv) {
    int rc; // return code from various functions
    char resolved_name[512]; // physical file name for out.txt
    FILE* f; // file pointer for out.txt

    /*
     * Before initializing BOINC itself, initialize diagnostics, so as
     * to get stderr output to the file stderr.txt, and thence back home.
     */

    boinc_init_diagnostics(BOINC_DIAG_REDIRECTSTDERR|
                           BOINC_DIAG_MEMORYLEAKCHECKENABLED|
                           BOINC_DIAG_DUMP_CALLSTACKENABLED|
                           BOINC_DIAG_TRACETOSTDERR);

    /* Output written to stderr will be returned with the Result (task) */

    fprintf(stderr, "Hello, stderr!\n");

    /* BOINC apps that do not use graphics just call boinc_init() */
    rc = boinc_init();
    if (rc) {
        fprintf(stderr, "APP: boinc_init() failed. rc=%d\n", rc);
        fflush(0);
        exit(rc);
    }

    /*
     * Input and output files need to be "resolved" from their logical name
     * for the application to the actual path on the client's disk
     */
    rc = boinc_resolve_filename("out.txt", resolved_name, sizeof(resolved_name));
    if (rc) {
        fprintf(stderr, "APP: cannot resolve output file name. RC=%d\n", rc);
    }
}

```

```

    boinc_finish(rc);    /* back to BOINC core */
}
/*
 * Open files with boinc_fopen() not just fopen()
 * (Output files should usually be opened in "append" mode, in case
 * this is actually a restart (which will not be the case here)).
 */
f = boinc_fopen(resolved_name, "a");

/* Now run up a wee bit of credit.  This is the "worker" loop */
{ int j, num, N, cislo1, cislo2;
  N = 123456789;
  for ( j=0 ; j<N ; j++ ){
    num=rand()+rand();    // just do something to spin the wheels
  }
  srand(time(NULL));
  cislo1=rand();
  cislo2=rand();
  fprintf(f, "%i, %i \n", cislo1, cislo2);
}
/* All BOINC applications must exit via boinc_finish(rc), not merely exit() */

fclose(f);
fprintf(stderr, "goodbye!\n");
boinc_finish(0);    /* does not return */
}
/*
 * Dummy graphics API entry points.  This app does not do graphics,
 * but it still must provide these empty callbacks (for BOINC 5).
 */
#if BOINC_MAJOR_VERSION < 6
void app_graphics_init() {}
void app_graphics_resize(int width, int height){}
void app_graphics_render(int xs, int ys, double time_of_day) {}
void app_graphics_reread_prefs() {}
void boinc_app_mouse_move(int x, int y, int left, int middle, int right){}
void boinc_app_mouse_button(int x, int y, int which, int is_down){}
void boinc_app_key_press(int, int){}
void boinc_app_key_release(int, int){}
#endif
#ifdef _WIN32

/*****
 * Windows: Unix applications begin with main() while Windows applications
 * begin with WinMain, so this just makes WinMain() process the command line
 * and then invoke main()
 */
int WINAPI WinMain(HINSTANCE hInst, HINSTANCE hPrevInst,
                  LPSTR Args, int WinMode)
{
  LPSTR command_line;
  char* argv[100];
  int argc;

  command_line = GetCommandLine();
  argc = parse_command_line( command_line, argv );
  return main(argc, argv);
}
#endif
//EOF//

```

Zdroj: Index of /pub/meyers/src/boinc, 2012, upravil Hruška J.

Příloha B: Originální Makefile

```
# Makefile for the BOINC hello.C program - "Hello, BOINC World!"
#
# Unpack this and just say 'make' to build the simplest BOINC application
#
# Can be built using either
#   A) the directory in which BOINC was BUILT, (called BOINC_BUILD), or
#   B) the location where BOINC was installed (called BOINC_PREFIX)
# Uncomment accordingly below.
#
# To build on Mac (Darwin) uncomment the #MAC# alternatives as appropriate,
# or see use `make -f Makefile.Mac` using that file in the same directory.
#
# Eric Myers <myers@spy-hill.net> - 16 June 2004
# Spy Hill Research, Poughkeepsie, New York
# Department of Physics and Astronomy, Vassar College, Poughkeepsie, NY
# @(#) $Revision: 1.37 $ - $Date: 2011/09/30 20:05:36 $ - $Name: $
#####

default: hello

# Version of app we are building (change as appropriate):
# (Major version of an app must^H^H^Hshould match major version
# of the BOINC core client!)

MAJOR_VERSION=6
MINOR_VERSION=12
#MAC#BUILD_ARCH=powerpc-apple-darwin
BUILD_ARCH=i686-pc-linux-gnu

BIN_FILENAME=hello_$(MAJOR_VERSION)_$(MINOR_VERSION)_$(BUILD_ARCH)

# A) Use BOINC_BUILD to point to where the BOINC distribution was built,
# so that you can test your app before you "install" on the system.
#BOINC_BUILD=/usr/local/src/boinc/boinc
BOINC_BUILD=/usr/local/src/boinc/server_stable

# B) Use BOINC_PREFIX to point to where you have installed BOINC.
# There should be lib and include/BOINC directories under it.
BOINC_PREFIX=/usr/local/boinc

# Compiler custom flags (if you need these)
#DEFS= -D_REENTRANT -D__NO_CTYPE

##- Header files:

#A) INCLUDES= -I$(BOINC_BUILD) -I$(BOINC_BUILD)/api -I$(BOINC_BUILD)/lib -include
$(BOINC_BUILD)/config.h
#B)
INCLUDES= -I$(BOINC_PREFIX)/include/boinc

##- Libraries:
#A) LIBDIRS= -L. -L$(BOINC_BUILD)/lib -L$(BOINC_BUILD)/api
#B)
LIBDIRS= -L$(BOINC_PREFIX)/lib

LIBS= $(LIBDIRS) -lboinc_api -lboinc

# How to invoke the C++ compiler:
CXX = g++
CXXCPP = $(CXX) -E
CXXFLAGS = -g
#LDFLAGS= -pthread
LDFLAGS= -static -pthread
# No -pthread on Darwin, and don't need -static on Mac anyway
#MAC#LDFLAGS=

CXXCOMPILE = $(CXX) $(DEFS) $(INCLUDES) $(CPPFLAGS) $(CXXFLAGS)
CXXLINK = $(CXX) $(LDFLAGS)
```

```

##- What we gotta make, and how to do it
OBJECTS= hello.o

hello: $(OBJECTS)
    $(CXXLINK) $(OBJECTS) $(LIBS) -o $@
    strip $@
    cp $@ $(BIN_FILENAME)

hello.o: hello.C

.C.o:
    $(CXXCOMPILE) -c $< -o $@

##- For testing dynamic loading:
ldd: clean hello
    ldd hello

# You can run the app "standalone" if you create an output file
# called "out.txt" first:
out:
    rm -f out.txt
    touch out.txt

# Testing it just means running it
test: hello out
    $(BIN_FILENAME)
    grep "Computation completed." out.txt || echo "Failed! "

##- Clean it up
clean:
    rm -f hello hello.o
    rm -f $(BIN_FILENAME) $(BIN_FILENAME).gz $(BIN_FILENAME).sig
    rm -f boinc_finish_called boinc_lockfile init_data.xml
    rm -f stderr.txt foo
    rm -f hello_?._?_$(BUILD_ARCH)
    cat /dev/null > out.txt

veryclean: clean
    rm -f out.txt out
    rm -f hello*_i686-pc-linux-gnu
    rm -f hello*_windows_intelx86.exe
    rm -f hello*_windows_intelx86.pdb
    rm -f hello*_powerpc-apple-darwin
    rm -f hello*_x86_64-pc-linux-gnu

distclean: veryclean
##

```

Zdroj: Index of /pub/meyers/src/boinc, 2012

Příloha C: Upravený Makefile

```
default: hello

MAJOR_VERSION=6
MINOR_VERSION=13
BUILD_ARCH=i686-pc-linux-gnu

BIN_FILENAME=hello_$(MAJOR_VERSION)_$(MINOR_VERSION)_$(BUILD_ARCH)

BOINC_BUILD=/home/boincadm/boinc
BOINC_PREFIX=/home/boincadm/boinc

DEFS= -D_REENTRANT -D__NO_CTYPE

INCLUDES= -I$(BOINC_BUILD) -I$(BOINC_BUILD)/api -I$(BOINC_BUILD)/lib
-include $(BOINC_BUILD)/config.h

LIBDIRS= -L. -L$(BOINC_BUILD)/lib -L$(BOINC_BUILD)/api

LIBS= $(LIBDIRS) -lboinc_api -lboinc

CXX      = g++
CXXCPP   = $(CXX) -E
CXXFLAGS = -g
LDFLAGS  = -static -pthread

CXXCOMPILE = $(CXX) $(DEFS) $(INCLUDES) $(CPPFLAGS) $(CXXFLAGS)
CXXLINK    = $(CXX) $(LDFLAGS)

OBJECTS= hello.o

hello: $(OBJECTS)
    $(CXXLINK) $(OBJECTS) $(LIBS) -o $@
    strip $@
    cp $@ $(BIN_FILENAME)

hello.o: hello.C

.C.o:
    $(CXXCOMPILE) -c $< -o $@

ldd: clean hello
    ldd hello

out:
    rm -f out.txt
    touch out.txt

test: hello out
    $(BIN_FILENAME)
    grep "Computation completed." out.txt || echo "Failed! "
```

Zdroj: Index of /pub/meyers/src/boinc, 2012, upravil Hruška J.

Abstrakt

HRUŠKA, Josef. *Distribuované výpočty s využitím „open source“ infrastruktury*. Diplomová práce. Plzeň: Fakulta ekonomická ZČU, 81 s., 2012

Klíčová slova: distribuované výpočty, BOINC, dobrovolnické počítání

Tato práce pojednává o distribuovaných výpočtech. V teoretické části je nejprve uvedena teorie k distribuovaným systémům, dále je zde popsán systém BOINC, průběh vytváření projektu v tomto systému a nakonec připojení se k existujícím výpočtům z pohledu klienta. Teoretické poznatky jsou aplikovány ve druhé části práce, kde je proveden modelový distribuovaný výpočet v systému BOINC. Konkrétně se jedná o jednoduchý výpočet na domácí síti, kde server poskytne jednoduchou aplikaci a poté pro ni distribuuje pracovní jednotky. Klientský počítač si tuto aplikaci stáhne, použije ji pro výpočet a vrací výsledky, které jsou na serveru uchovávány. Dále jsou zde také uvedeny možnosti nastavení serveru, klientské aplikace a webových stránek projektu. V poslední části práce jsou popsány současné vědecké projekty distribuovaných výpočtů nejen v systému BOINC.

Abstract

HRUŠKA, Josef. *Distributed computing using open source infrastructure*. Diploma Paper. Pilsen: Faculty of Economics, WBU in Pilsen, 81 p., 2012

Key Words: distributed computing, BOINC, volunteer computing

This essay deals with distributed computing. In the theoretical part of this essay is described the theory of distributed systems. The next issue is describing the BOINC system, a project creation in this system and joining existing computations from the perspective of a client. The theoretical knowledge is applied to the practical part of this essay, where is executed the distributed computation model in the BOINC system. Specifically, it is a simple computation in home network. A Server provides simple application and distributes work units. The client computer downloads this application, uses it to the computation and returns results that are stored on the server. Then there are noted server options, client options and a project web site. In the last part of this essay are described current distributed computing research projects.