BigData

Historie

- Ti, kteří se nepoučí z historie, jsou odsouzeni k tomu, aby opakovali její chyby – George Santayana

- Wolfenstein3D, Retaliator F29 (vč. multiplayeru), LHX – staré 3D hry, které plynule běhaly na 286 s 1MB RAM a bez grafického akcelerátoru
    - Kolik dnešních programátorů by to dokázalo?

- Na kolik by se lišil referát vytisknutý z MS Windows 8 s MS Office 2013 od referátu vytisknutého z MS Windows 3.1 s MS Office 4.0 / AmiPro?
    - A kolik funkcí je pro to skutečně nezbytných až v nové verzi? Žádná?
    - A proč instalace, která se původně vešla na diskety, dnes zabírá několik gigabytů?

- Jsou procesory jako 286 a 386 stále relevantní?
    - Ano, ve specifických prostředích, kde není možné provozovat novější procesory
        - Např. ve vesmíru, kde by novější procesory nevydržely kvůli záření
        - I když důvodem je i to, že než se systém otestuje s daným procesorem, tak se už na trhu dávno prodává novější typ

- V IT došlo k několika věcem, které ač jsou na jednu stranu prospěšné, na kvalitě programátorů si vybírají svou daň

    - Zvyšuje se výkon a počet procesorových jader v běžně dostupném počítači, stejně tak jako množství dostupné RAM, místa na disku a přítomnost GPGU

        - Žijete-li v přepychu, začnete plýtvat – tj. psát neefektivní kód

    - Garbage Collector - Java umožnila programovat i těm lidem, kteří by nezvládnuli psát programy s manuální správou paměti

        - Když vás programovací jazyk odstíní od hardwaru, na jednu stranu vám sice umožní více se soustředit na vlastní problém (a nechat ho vyřešit víc lidí), ale na druhou stranu lze takové programy těžko nazvat efektivními

            - Respektive to ze svého subjektivního pohledu dokáže např. Java programátor, který v životě nenapsal efektivní program v C/C++/Asm

Are you quite sure that all those bells and whistles, all those wonderful facilities of your so called powerful programming languages, belong to the solution set rather than the problem set?

— Edsger W. Dijkstra

- Průmysl – v příliš mnoha firmách rozhodují lidé zodpovědní za peníze způsobem podobným jednomu z následujících scénářů:

  - Scénář A:
    - Neděláme software pro bojový letoun poslední generace, ale nástroj do kanceláře
      - Anebo nějakou malou utilitku, která se nebude pouště často,
      - Atd.

    - Když to uděláme v Javě/C#, tak se nám urychlí vývoj, protože se nemusíme ladit s chybným přístupem do paměti
    - A když to bude pomalé, tak prostě koupíme výkonnější hardware

If we'd asked the customers what they wanted, they would have said "faster horses"

— Henry Ford

    - A abychom to neprogramovali celé sami, použijeme už hotové knihovny/komponenty
      - Takže se pak v praxi klidně použije jakýsi moloch, jenom pro jednu funkci, kterou by nakonec bylo i vhodnější si napsat vlastnoručně

Most software today is very much like an Egyptian pyramid with millions of bricks piled on top of each other, with no structural integrity, but just done by brute force and thousands of slaves.

— Alan Kay

Compatibility means deliberately repeating other people's mistakes.

— David Wheeler

- Scénář B:
  - Máme problém, na který neexistuje už hotový software
  - Protože čas jsou peníze, potřebujeme programátora, který si nebude „mazlit kde jakou řádečku kódu“, ale někoho, kdo nám dá k dispozici něco funkčního co nejdříve
  - Jinými slovy, hledají kutila namísto profesionála

- Takže pak vznikají naprosto neuvěřitelně tristní řešení, kdy např. na práci zvládnutelnou 286 (i s oněmi 640kB) v reálném čase potřebujete procesor poslední generace s bůhví kolika jádry a gigabyty RAM a diskového prostoru
  - A jejich autoři se stále považují za programátory
    - Nebuďte takoví i vy!

The computing scientist’s main challenge is not to get confused by the complexities of his own making.

— E. W. Dijkstra

At first I hoped that such a technically unsound project would collapse but I soon realized it was doomed to success. Almost anything in software can be implemented, sold, and even used given enough determination. There is nothing a mere scientist can say that will stand against the flood of a hundred million dollars. But there is one quality that cannot be purchased in this way -and that is reliability. The price of reliability is the pursuit of the utmost simplicity. It is a price which the very rich find most hard to pay.

— C.A.R. Hoare

Increasingly, people seem to misinterpret complexity as sophistication, which is baffling - the incomprehensible should cause suspicion rather than admiration. Possibly this trend results from a mistaken belief that using a somewhat mysterious device confers an aura of power on the user.

— Niklaus Wirth

Schopný programátor

- Rozumí problému, který se řeší – např. co znamenají naměřená data a chápe, jak byl odvozen matematický model, který je popisuje
  - Nechápe-li, neví jak stanovit okrajové podmínky, které mají vliv na výpočetní náročnost
- Rozumí použitým výpočetním postupům
  - Např. chápe význam stacionárních bodů při hledání extrému funkce více proměnných při výpočtu optimálních parametrů účelové funkce
- Rozumí tomu, co dělá překladač a hardware s jeho kódem
  - I v C++ se dá „kutit“ namísto programovat
- Chápe, že návrh datových struktur také určuje náročnost

Q: What is the most often-overlooked risk in software engineering?

A: Incompetent programmers. There are estimates that the number of programmers needed in the U.S. exceeds 200,000. This is entirely misleading. It is not a quantity problem; we have a quality problem. One bad programmer can easily create two new jobs a year. Hiring more bad programmers will just increase our perceived need for them. If we had more good programmers, and could easily identify them, we would need fewer, not more.

— David Parnas

My definition of an expert in any field is a person who knows enough about what's really going on to be scared.

— P. J. Plauger, Computer Language, March 1983

- Ego – jestli si myslíte, že jste napsali efektivní (paralelní) program, tak to vůbec nemusí být pravda

Reálná náročnost výpočtů v praxi
Aneb, opravdu pracujeme s BigData?

- Na většinu normálních, byť výpočetně náročných, problémů by měl stačit SMP
  - Máme-li k dispozici schopného programátora

- Některé problémy lze urychlit použitím GPGPU
  - Nicméně pozor na to, že urychlení nemusí být tak zázračně velké, jak se může v počátečním nadšením zdát
    - Např. viz (pozor: střet zájmů Intel a nVidia) http://www.hwsw.hu/kepek/hirek/2010/06/p451-lee.pdf
  - Opět to vyžaduje schopného programátora

- Pokud se však stále jedná o problém, na jehož řešení to nestačí, pak by mělo stačit použití buď (zastaralého) PVM, nebo novějšího MPI s tím, že programy jednotlivých procesů by měl opět napsat schopný programátor
  - Normálně byste se neměli dostat do situace, kdy na něco nebude stačit MPI cluster

- V případě, že se skutečně jedná o skutečně rozsáhlá data, např. tzv. BigData, existují distribuované frameworky MapReduce
  - Jsou určené pro „serverové farmy“
  - Pro true-BigData jako řešení jako je např. Square Kilometre Array to ale nakonec nejspíš stejně skončí proprietární technologií

BigData

- Definice BigData je vágní – říká, že jde o takový objem dat, který nezpracujete se stávajícím vybavením v rozumně krátkém čase
  - Definice neřeší neefektivní datové struktury ani špatně napsaný program ani ne/schopnost programátora, které určují čas výpočtu
- Bohužel se svým způsobem o jedná o další buzzword, který neřeší smysluplnost bigdata – tj. zda nahromadění velkého objemu dat je vždy ku prospěchu věci, a zda to vůbec nějakým způsobem překračuje stávající hranci poznání o distribuovaných výpočtech

Cloud

- Řekneme-li, že cloud je po gridu jenom další marketingové buzzword v řadě, ve skutečnosti znamenající starý dobrý distribuovaný systém, pak budeme pravdě vzdáleni s dostatečně malým epsilon
- V podstatě se jedná o vylepšení gridu v tom smyslu, že si dynamicky pronajímáte hardware a na něm instalovaný software, přičemž obojí je „někde v cloudu“

… what society overwhelmingly asks for is snake oil. Of course, the snake oil has the most impressive names — otherwise you would be selling nothing — like “Structured Analysis and Design”, “Software Engineering”, “Maturity Models”, “Management Information Systems”, “Integrated Project Support Environments” “Object Orientation” and “Business Process Re-engineering” (the latter three being known as IPSE, OO and BPR, respectively).

— *Edsger W. Dijkstra — EWD 1175: The strengths of the academic enterprise [Today we could add ‘Extreme Programming’, ‘Agile Software Development’ and many more.]*

Distribuované MapReduce

- V počítači se sdílenou pamětí můžeme říci, že map by odpovídalo plánovači, a reduce by odpovídalo redukční funkci, jak to říká její název
  - Pokud bychom neuvažovali TaskStealing plánovač, pak by šlo o kombinaci SetThreadAffinity a CreateThread, která bere jako parametr funkci s kódem vlákna

- V distribuovaném prostředí map odpovídá procesu rozdělení dat na části a jejich distribuci na jednotlivé uzly
  - Které je pak mohou ještě dále rozdělit

- Reduce pak znamená provedené nějaké operace nad danou částí dat na příslušném uzlu
  - A následné zkombinování mezivýsledků do finálního výsledku

- Ač se používá pro zpracování rozsáhlých dat, než urychlení, řeší se spolehlivost výpočtu s ohledem na riziko selhání výpočetního uzlu
- Uživatel v podstatě napíše jenom operace map a reduce
- Použitý framework pak může dosáhnout dalšího urychlení ještě tím, jak překrývá výpočetní čas s komunikačním časem

- MapReduce v podstatě poskytují i databázové servery, když zpracovávají SQL dotazy