

Alternativy, Budoucnost aneb Kam směřuje další vývoj

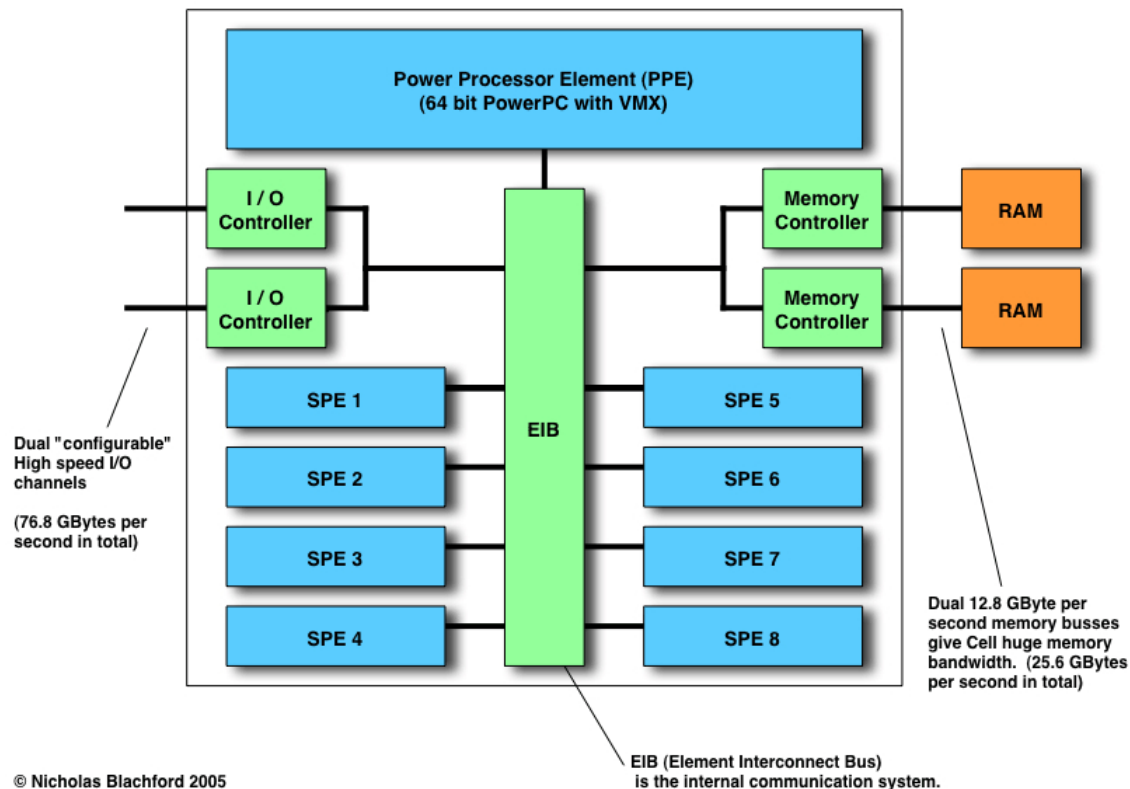
Koprocesory

- Jak získat vyšší výkon, když (general purpose) procesor nestačí?
 - Obecně paralelizací
- Doposud jsme mluvili o *homogenní* paralelizaci
 - Hyperthreading, více jader, paralelní počítače
- Alternativou jsou ale *heterogenní* architektury
 - Použití koprocesorů
 - Nemají charakter general purpose, díky tomu mohou zvládat určité typy výpočtů mnohem efektivněji
 - Známe z historie v podobě numerických koprocesorů (80387)
- Rozdílné přístupy
 - Grafické koprocesory (Cuda/Tesla architektura firmy NVidia)
 - ClearSpeed
 - IBM Cell
 - FPGA (programovatelný hardware, programování „in silico“)

IBM CELL

- Společný projekt Sony, IBM, Toshiba
- 1 Power Processor Element, 8 Synergistic Proc. Elems

Cell Processor Architecture



FPGA

- Programovatelná hradlová pole
- Jednodušší návrh, běží na nižších frekvencích
- **Programovatelnost** – programování “in silico” – FPGA se rekonfiguruje pro konkrétní algoritmus:
 - Circuit diagram v HDL (Hardware Description Language)
 - Následná konfigurace obvodů (může být časově náročnější)
- Hybridní architektury
 - Xilinx – kompletní systém, embedded procesor, (PowerPC) + FPGA
 - Mitrionics – modifikovaný jazyk C (flow-based programming, přesná pravidla ulehčující paralelizaci), odladění a simulace kódu, nakonec “zápis” do hardwaru

Koprocesory – výpočetní modely

- SIMD
 - Návrat k původní ideji, realizovaný ale v rámci jednoho koprocesoru
- Systolické
 - Více jednotek koordinovaně řešící úlohy formou „vlny“. Explicitní prokládání výpočtů a komunikace, vhodné např., pro jednotky (součást jednoho koprocesoru) organizaované do pole
- Dataflow
 - Streaming
- Optimalizace pro heterogenní systémy ještě složitější než pro paralelní počítače
 - Zatím spíše vývojová prostředí, automatická optimalizace v počátcích

Další oblasti

- Gridy
 - Rozsáhlé distribuované systémy
 - Heterogenní, více administrativních domén, nízká spolehlivost komponent
- Fault tolerance
 - Odolnost proti výpadku, ale stále častěji především *Zotavení po výpadku*
 - Znovu v popředí zájmu (rozsáhlé paralelní a zejména distribuované systémy nelze konstruovat bez toho, že komponenty selhávají)
 - Self healing systémy (v podstatě snaha zakomponovat fault toleranci do systémů orientovaných na služby)
 - Standardní přístup – checkpoint a restart – prakticky nefunguje v distribuovaných systémech
- Operační systémy
 - Stále patrnější vliv na skutečnou efektivitu paralelních a distribuovaných systémů
 - Přímá podpora multi-core procesorů
 - Podpora distribuované prostředí v jádře – např. projekt ExtremeOS
- Virtualizace
 - Řešení těžkých problémů – plánování zdrojů, zvýšení odolnosti, ...

Další vývoj

- Začneme pohledem do minulosti
 - Dnešní stolní počítač má výkon předvčerejšího superpočítače
 - Stíhají uživatelé skutečně držet krok?
- A budeme pokračovat novými architekturami
 - A také problémy, které na tyto architektury čekají

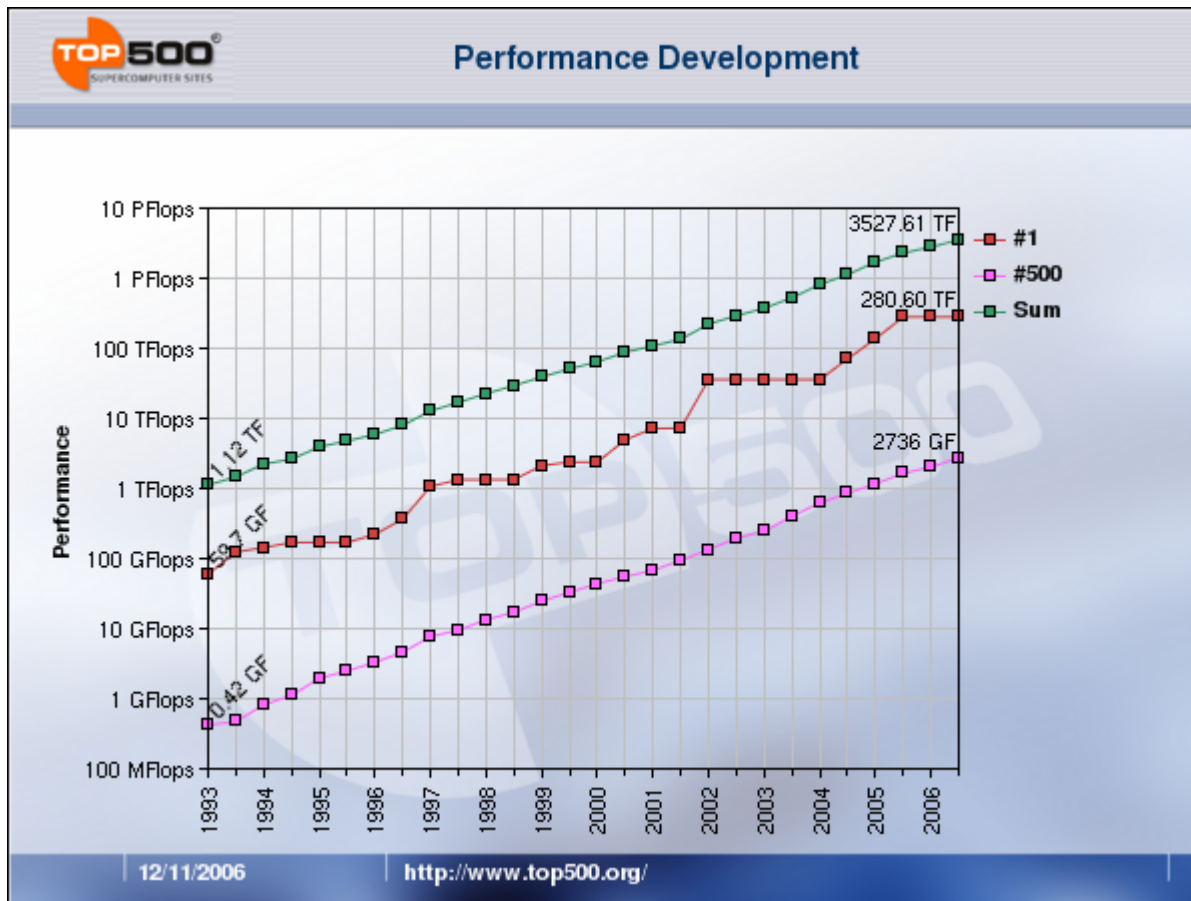
Před 25 lety

- Cray X-MP
- Frekvence: 105 MHz
tj. 9.5 nanosekund
- 2 processory po 200 Mflops
- Paměť 128 MB (16 megaslov)
- Vyroběn 1982 a stál přes 10 milionů dolarů
- Zhruba na úrovni dnešních chytrých mobilních telefonů či PDA



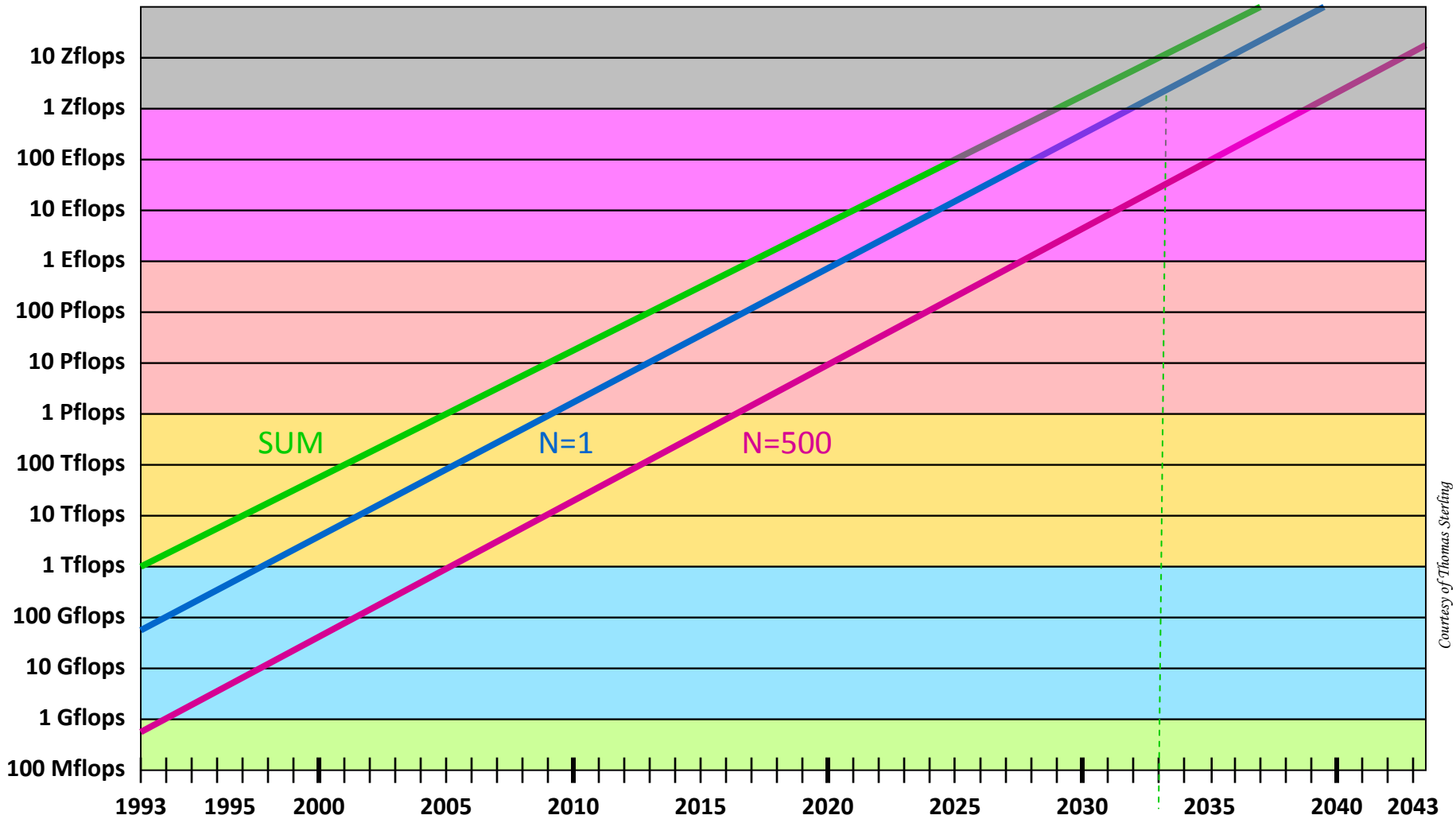
A co bude za 25 let?

Podívejme se na seznam TOP500



- Graf pokrývá 13,5 let
- Výkon nejrychlejšího počítače narostl 4700 krát
- Výkon posledního počítače vzrostl 6514 krát
- Celkový nárůst: 3150x

10 Zetaflops před rokem 2035

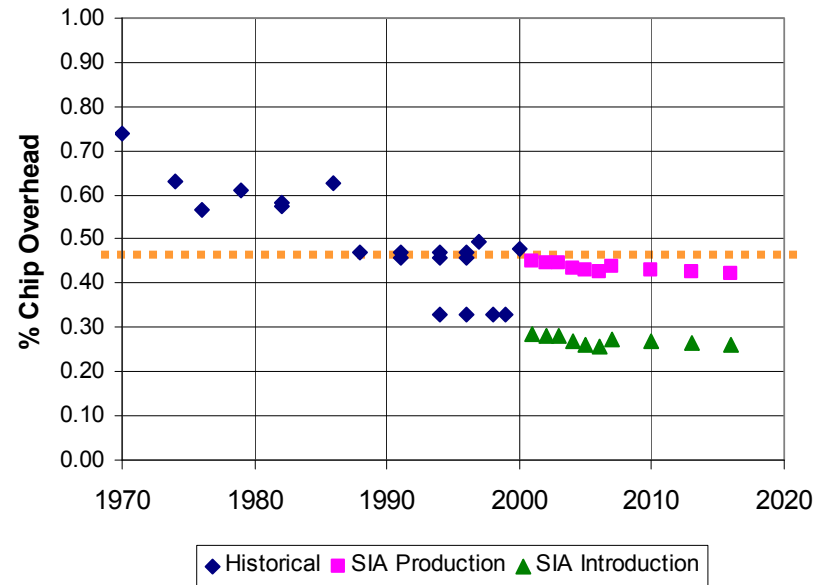
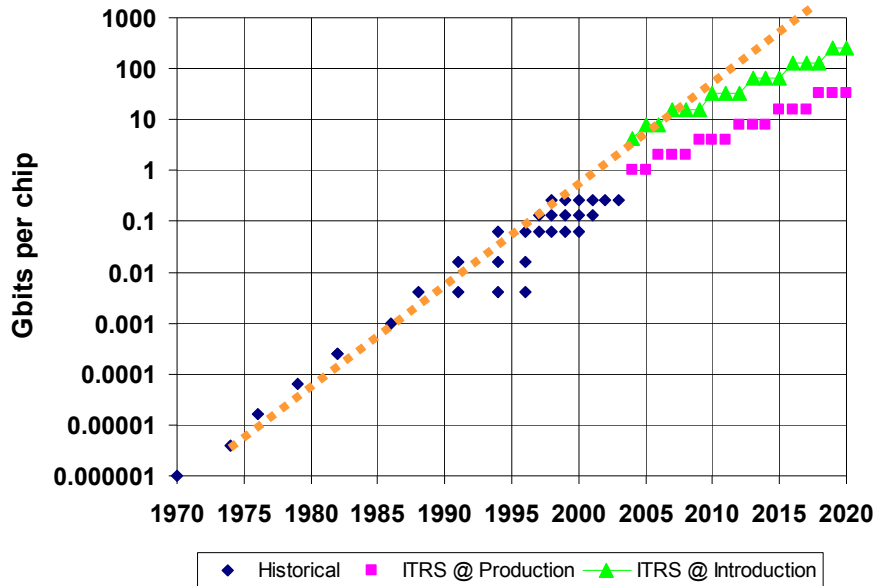


Courtesy of Thomas Sterling

Je tato projekce reálná

- Výkon procesorů
 - Nenarážíme na nějaké fyzikální limity?
- Rychlost paměti
 - Obdobně, je možné dále zvyšovat hustotu?
- Rychlost I/O
 - Zpravidla v pozadí, ale stále více ovlivňuje celkový dosažitelný výkon (jak síť, tak diskový a podobné subsystémy)
- Paralelní výpočetní modely
 - Stále jen počátky, automatická efektivní a obecná paralelizace (natož pak distribuce na heterogenní distribuovaný systém) zůstává stále jen snem

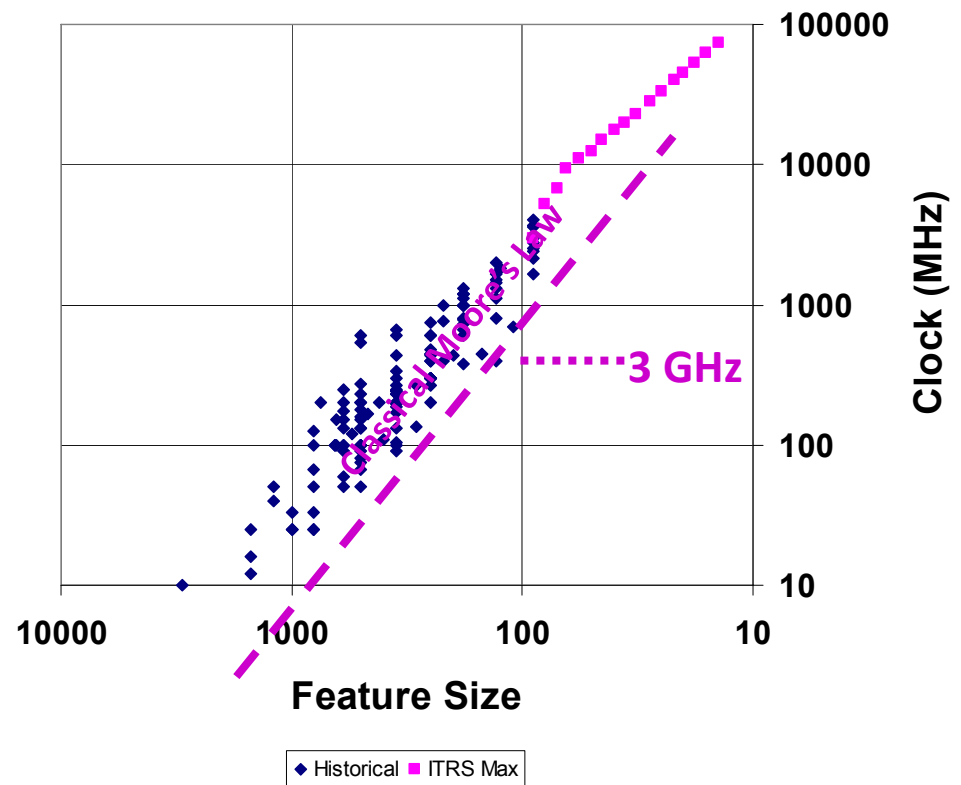
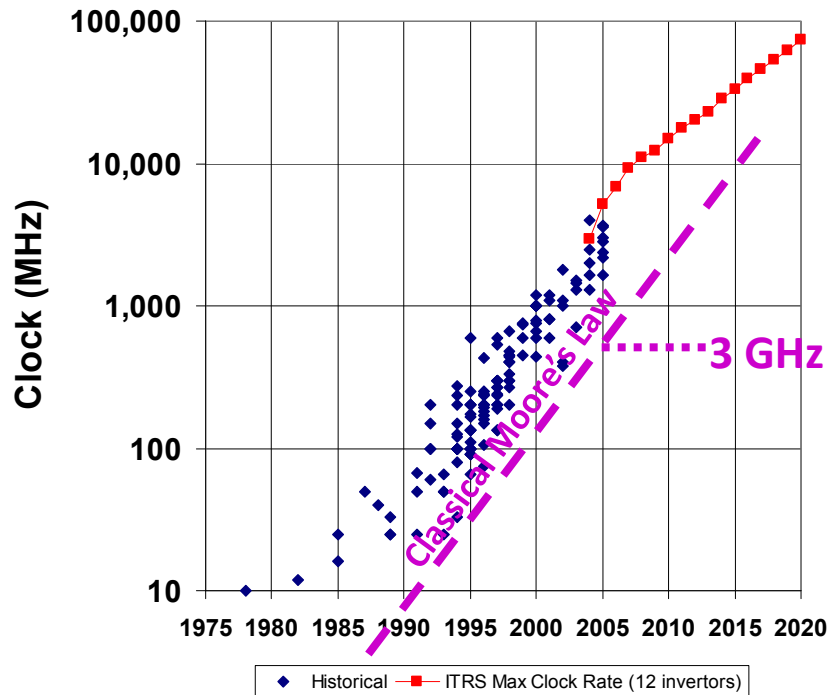
Klasická DRAM paměť



Viditelné zpomalování nárůstu hustoty

Příliš velká část chipu (cca 45%) *není* paměť

Frekvence komponent



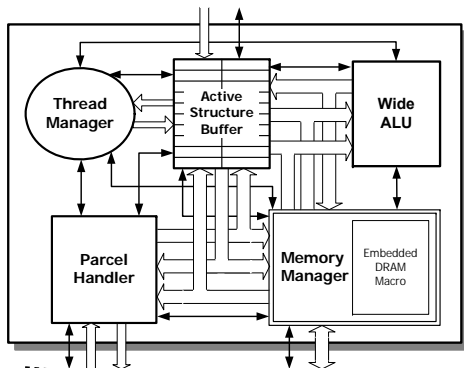
Pro rok 2008 byla v roce 2005 predikována základní frekvence 5,2 GHz, predikce ale selhala. Vyráběné systémy se zastavily na hranici cca 3 GHz. Podrobnější analýza ukáže, že hlavním problémem je energie, resp. její přílišná koncentrace (na 1mm²) při vyšších frekvencích – **neumíme teplo dostatečně rychle odvádět**

Překonání hranice

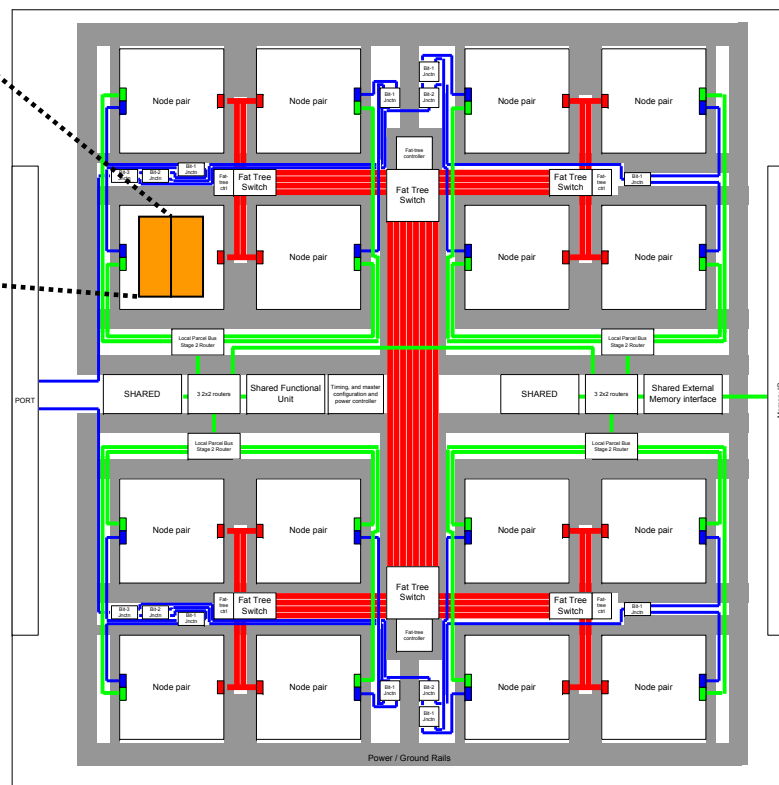
- V podstatě zásadní změnou architekturu
- Rozložení paměti více přímo mezi výpočetní komponenty, proložení více a méně zatížených komponent tak, aby se usnadnil rozvod tepla
 - Teplá místa menší, obklopena chladnějšími – výsledkem je vyšší škálovatelnost, tj. i vyšší dosažitelná frekvence
- Vyšší integrace komponent
 - Složitější pro vývoj
- MIND prvky a z nich skládané systémy (Dodecatron, VISION2020)

MIND prvky

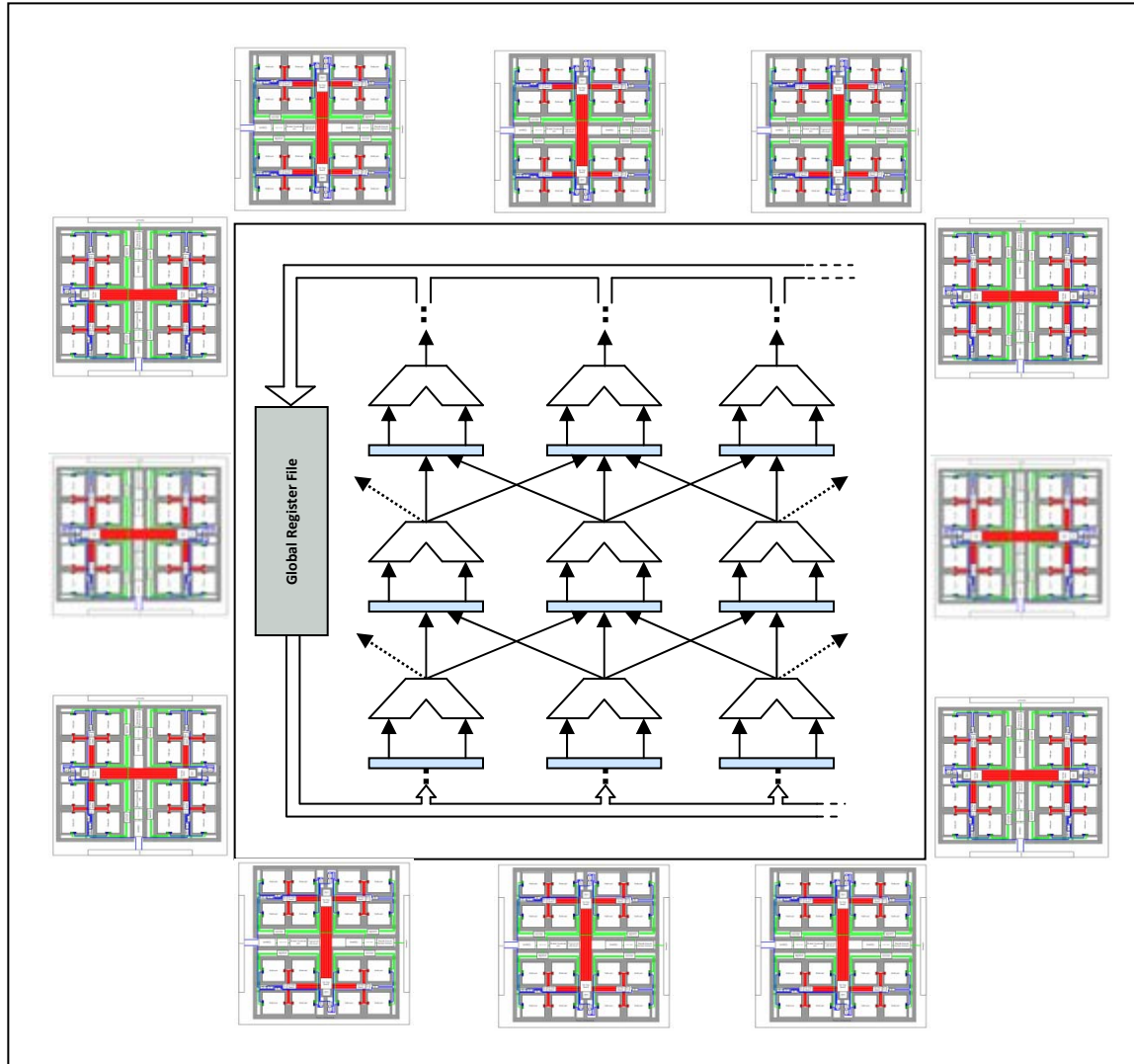
MIND memory accelerator



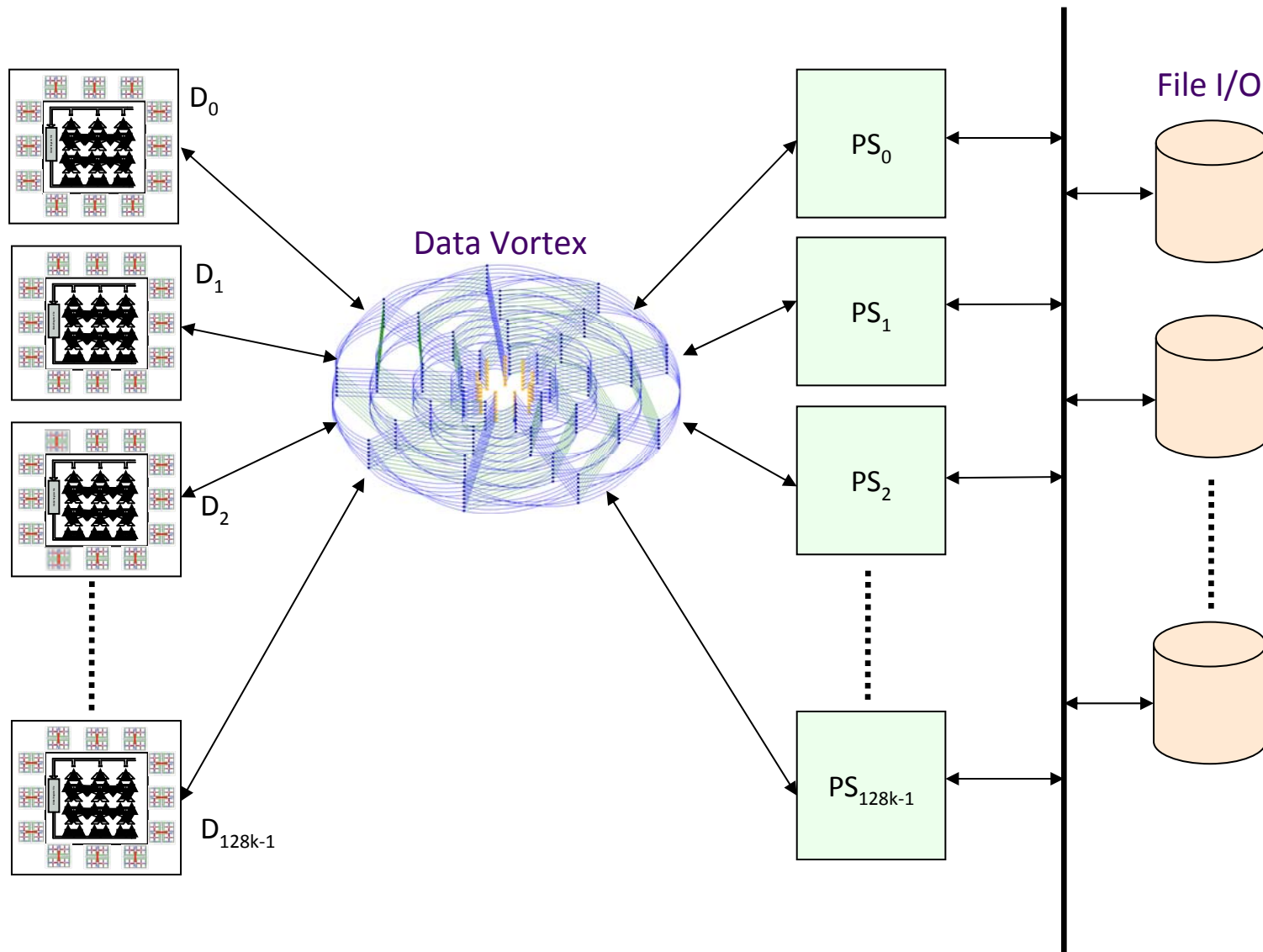
Gilgamesh



Dodecatron



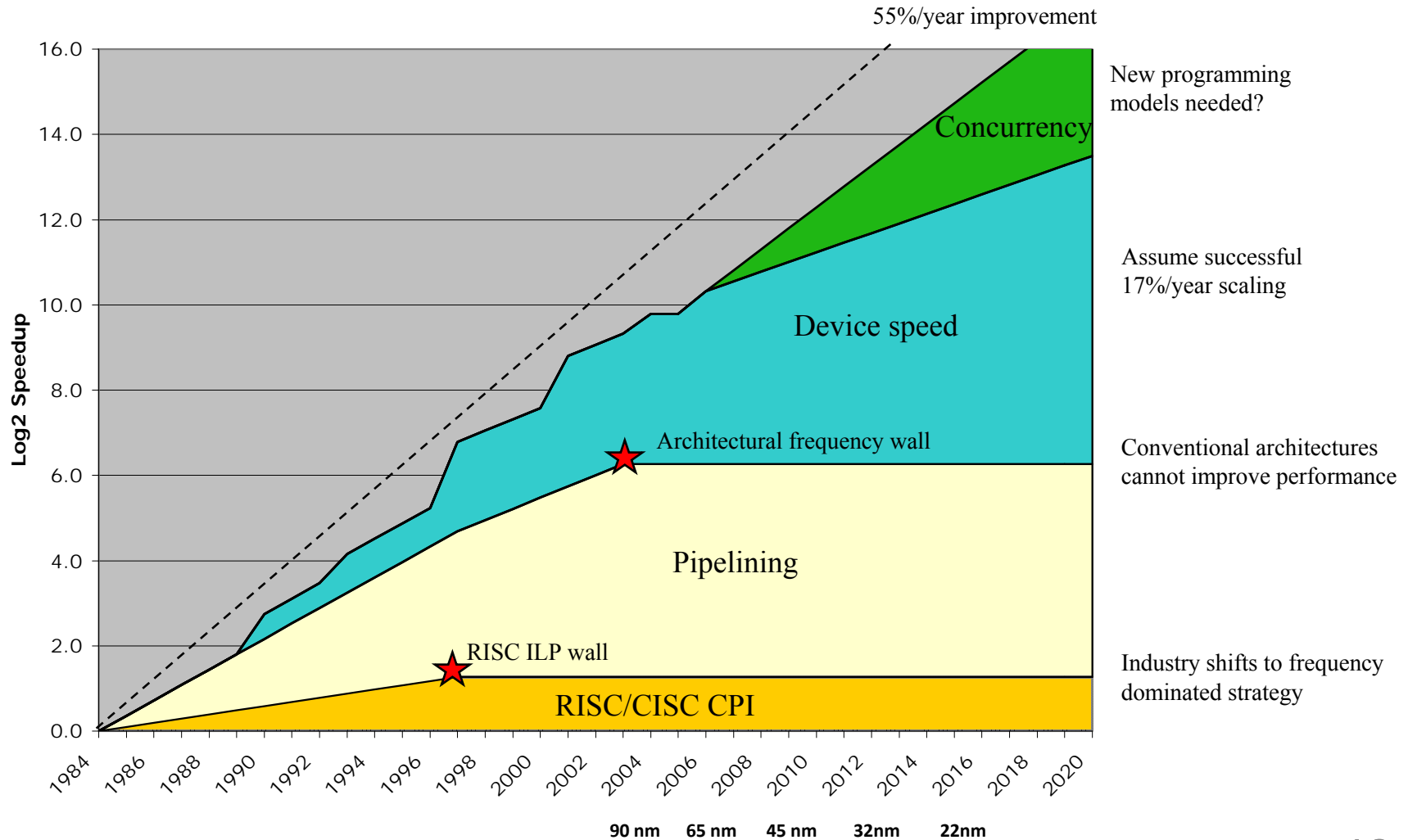
Vision2020 System



Škálování výkonu

Příspěvky jednotlivých architektonických řešení

Single-processor Performance Scaling



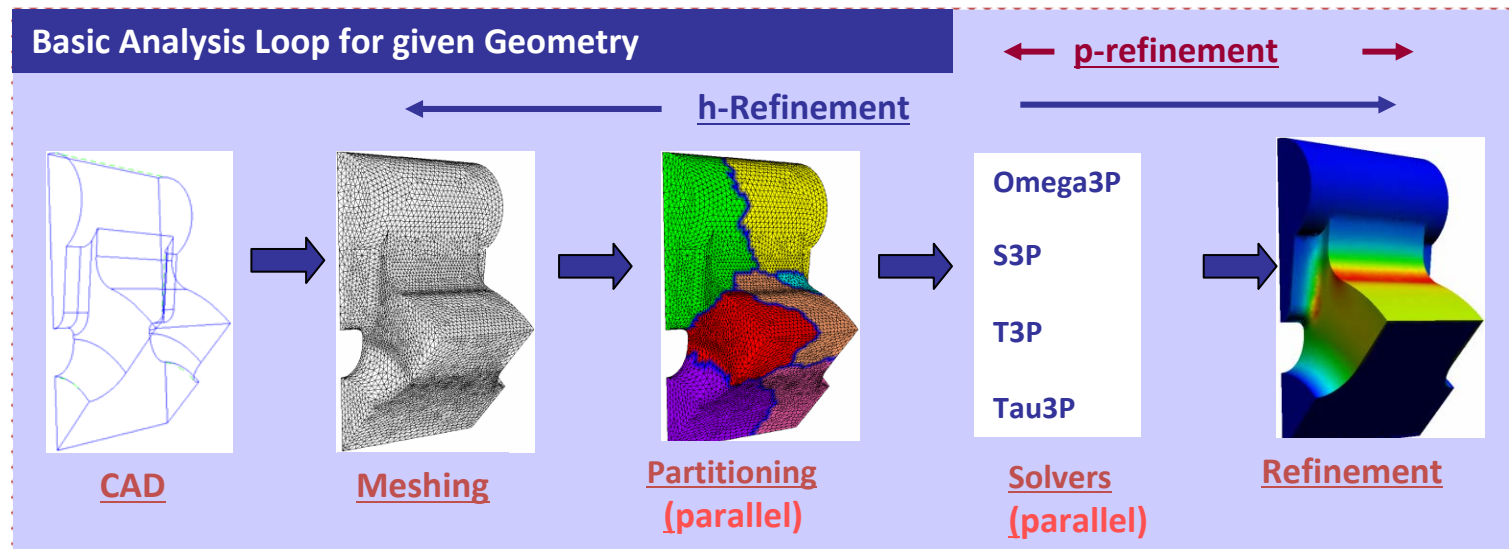
Další rozvoj

- Nové fyzikální jevy
 - Josephsonův tunelovací efekt – základ SFQ architektur
 - Experimenty s frekvencí až 770 GHz
 - Radikálně odlišná výroba, průmyslové nasazení zatím nejasné
- Nová architektura
 - Od striktně lineární k 2D a 3D
 - Přímé mapování výpočtu /(včetně explicitních komunikací) na 2D/3D výpočetní substrát
 - Současné zkušenosti s FPGA (mapování výpočtu na hardware) však ukazují na nedostatek vývojových a optimalizačních nástrojů a přístupů
- Radikální změny
 - Kvantové tečky
 - Kvantové počítače

A k čemu to bude?

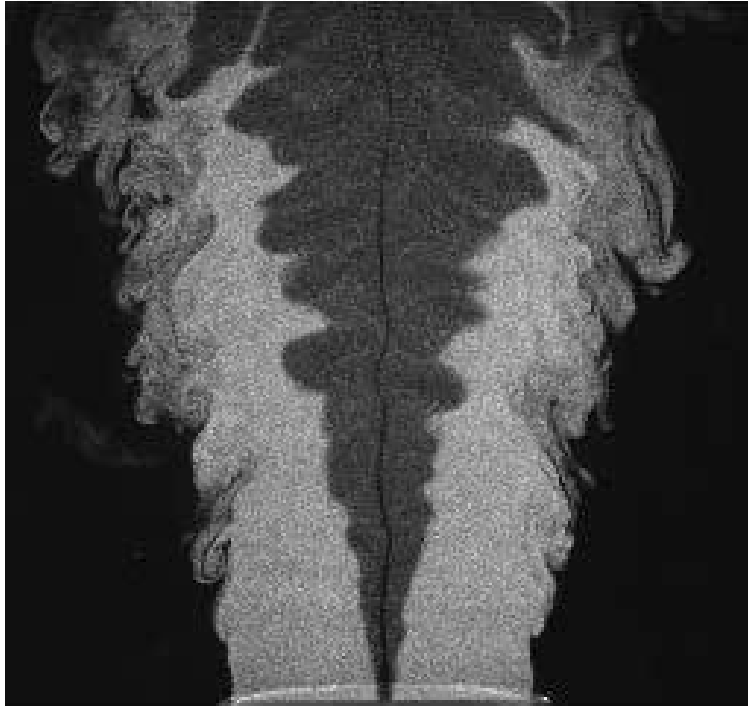
- Existují skutečně problémy, které využijí na úrovni Zetaflops+?
- Jednoduchá odpověď zní: **Ano**
- V oblasti modelování složitých dynamických systémů jsme dosud na počátku, na seriózní modely „in silico“ nestačí výpočetní výkon
 - Samozřejmě nejde jen o hrubou sílu, studium vlastního problému a návrh nových algoritmů může urychlit simulace mnohem více než vývoj na rovní hardware
 - To však nezpochybňuje vlastní zájem o prostředí pro náročné výpočty a další zvyšování jeho výkonu
- Společný problém: současné modelování přes řadu časových škál
 - Např. simulujeme proces, který trvá řádově hodiny, pomocí dějů na molekulární úrovni, které mají reakční dobu na úrovni nanosekund až mikrosekund

Příklad 1: Computer Aided Design



- Výše uvedený cyklus není možné příliš často opakovat (výpočetní náročnost)
- Optimalizace výsledného tvaru (např. drobné změny tvaru povrchu, kombinace konvexních a konkávních oblastí) by však vyžadovala stovky tisíc až miliony průchodů

Příklad 2: Řízené spalování



Experimental PIV measurement

Instantaneous flame front imaged by density of inert marker

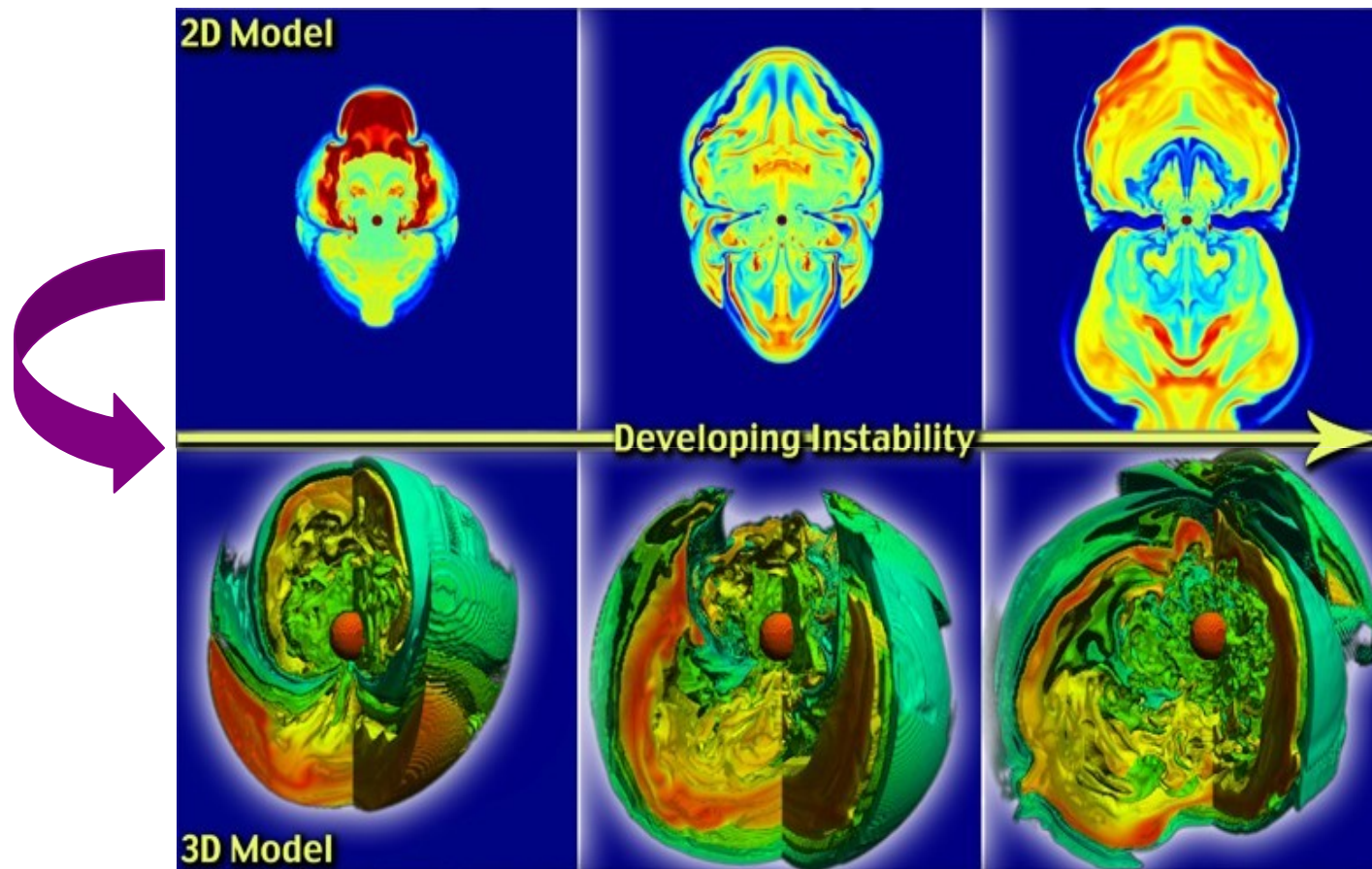


Simulation

Instantaneous flame front imaged by fuel concentration

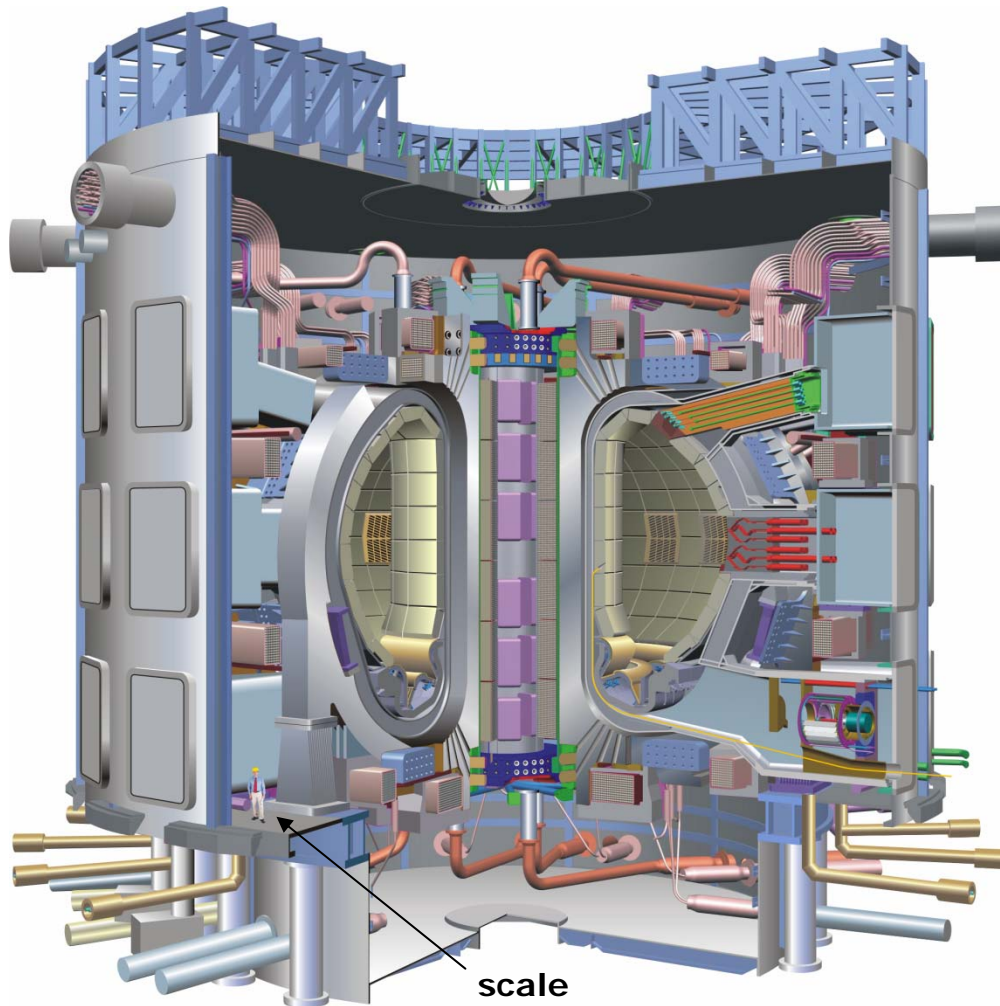
Images c/o R. Cheng (left), J. Bell (right), LBNL, and NERSC
2003 SIAM/ACM Prize in CS&E (J. Bell & P. Colella)

Příklad 3: Výbuch supernovy



Základní problém je v dynamice a nestabilitě počáteční fáze, která ale určuje výsledný tvar a směr vyzařované energie supernovy. Jednodušší modely (dnešní) poskytují základní představu o dějích při výbuchu supernovy, ale teprve plný model – od okamžiku výbuchu po expanzi – by umožnil generovat radiační stopu supernovy a tu porovnat s reálným pozorováním.

Příklad 4: Jaderná fúze, projekt ITER



Projekt ITER (International Thermonuclear Experimental Reactor) bude mít za úkol vybudovat fúzní reaktor (Tokamak).

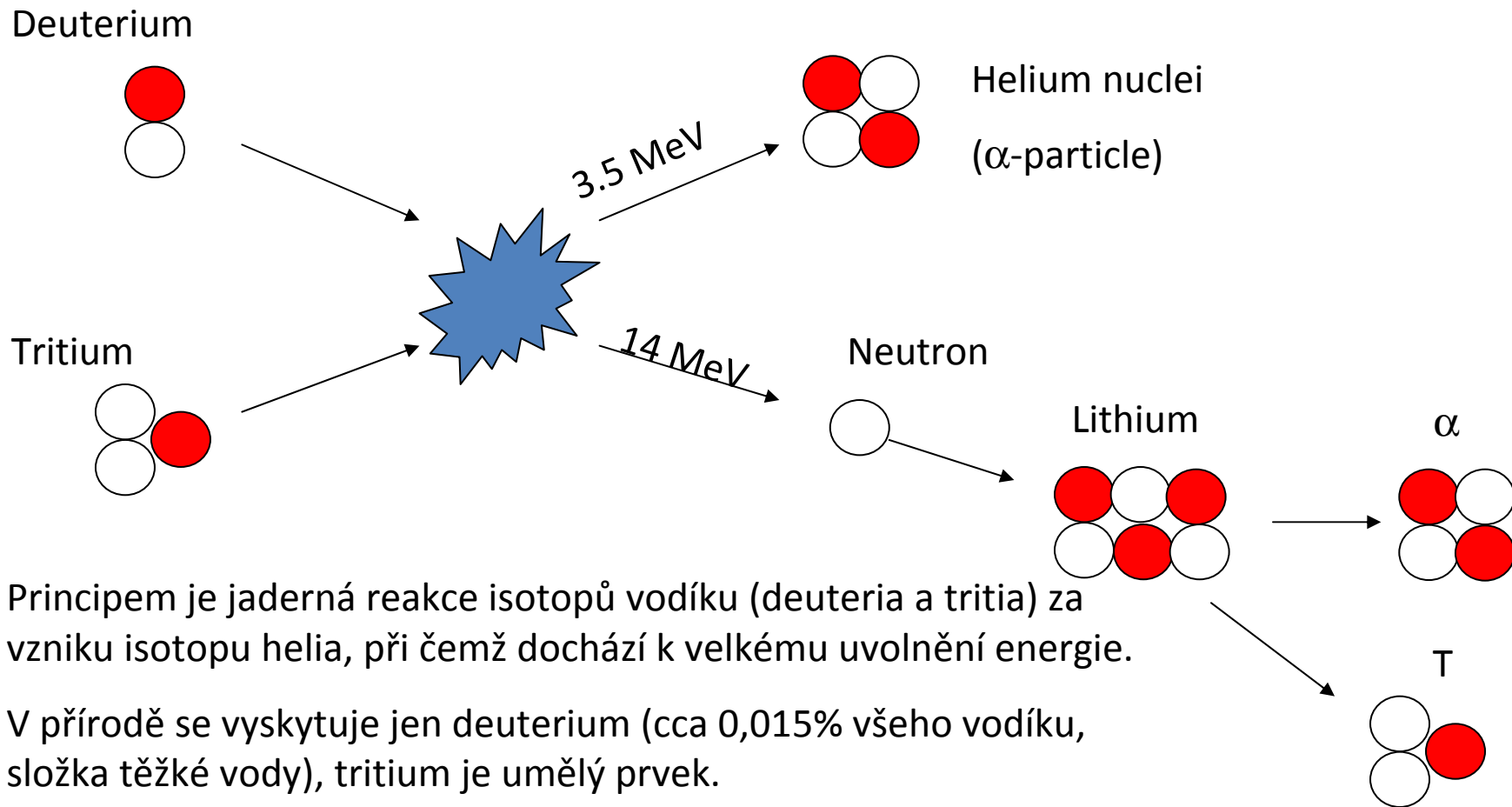
ITER bude umístěn v Toulouse (Francie), jedná se ale o globální projekt (podobně jako CERN)

Zapojené země/oblasti:

- Evropská unie
- Japonsko
- USA
- Rusko
- Korea
- Čína

- Předpokládaný výkon: 500 MW
- Cena: cca 12 miliard Euro
- Dnes rozhodnuto o místě, spuštění 2015

Příklad 4: Jaderná fúze II



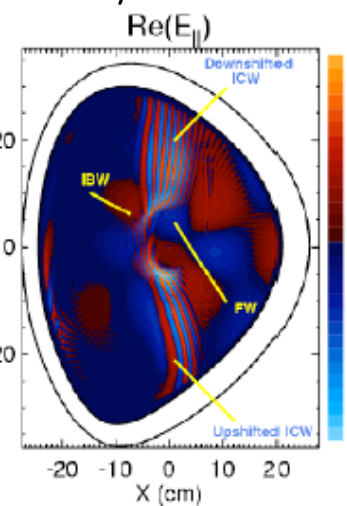
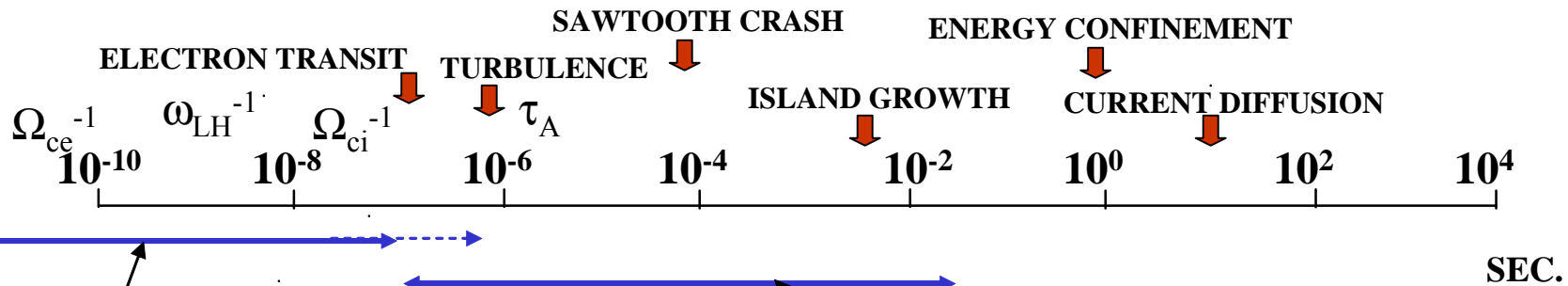
Principem je jaderná reakce izotopů vodíku (deuteria a tritia) za vzniku isotopu helia, při čemž dochází k velkému uvolnění energie.

V přírodě se vyskytuje jen deuterium (cca 0,015% všeho vodíku, složka těžké vody), tritium je umělý prvek.

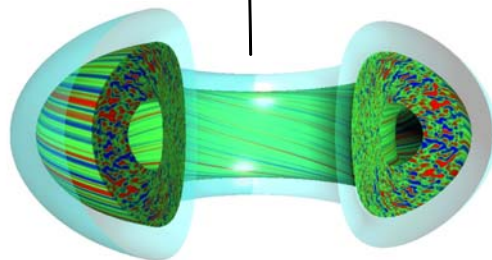
Reakce je ale trvale udržitelná kombinací s další jadernou reakcí vyzářeného neutronu s lithiem (potřebný isotop opět rovněž běžný přírodní prvek).

Příklad 4: Jaderná fúze III

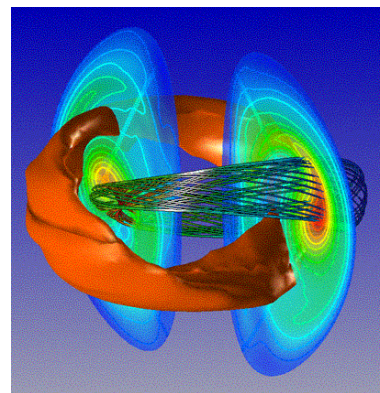
- Simulace dějů ve fúzním reaktoru má 4 fáze:
 - Vlastní jaderná reakce
 - Mikroturbulence
 - Magnetohydrodynamické děje (jaderná plazma je držena v magnetické nádobě, protože žádný pevný materiál nevydrží fyzický kontakt)
 - Konvence a transport tepla
- Každá fáze má jinou časovou škálu
 - Dnes řešeno odděleně
 - Neexistuje výpočetní model celého Tokamaku



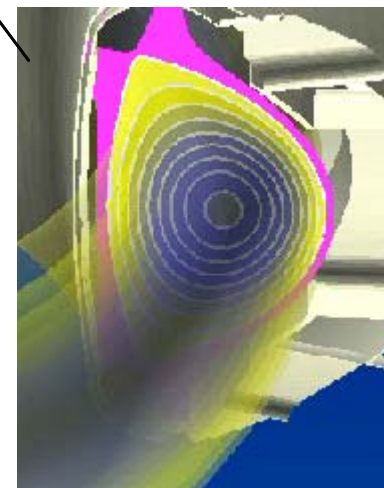
(a) RF codes



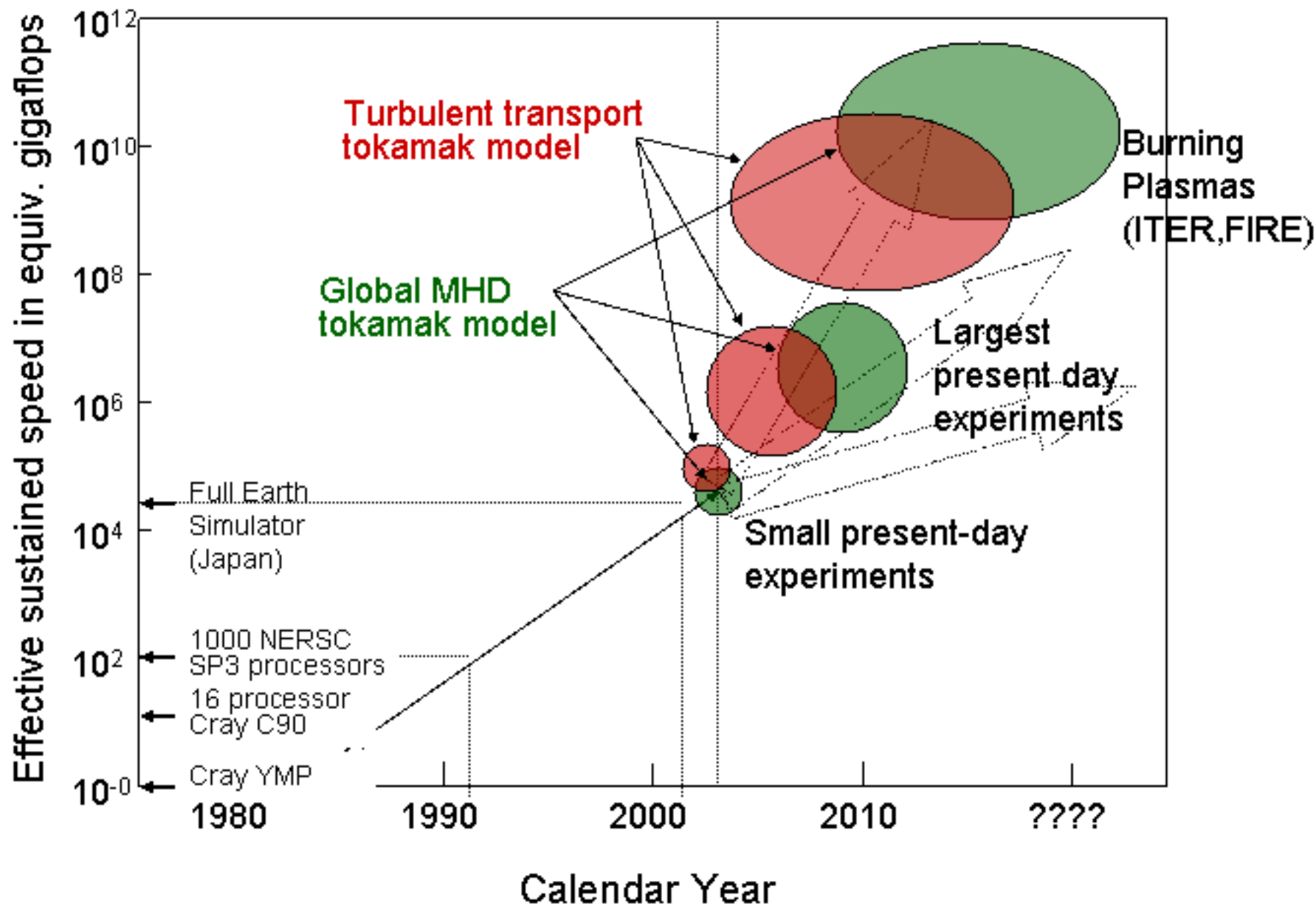
(b) Micro-turbulence codes



(c) Extended-MHD codes



(d) Transport Codes



Příklad 5: Výpočetní chemie

- Určení přesné 3D struktury bílkovin a nukleových kyselin ze znalosti jednotlivých stavebních kamenů
 - Vliv prostředí, to se ale velmi těžce modeluje (zpravidla jen jako kontinuum)
- Dynamika poškození DNA
 - Procesy stárnutí, vliv radiace (i přirozené), opravné procesy
- Simulace dějů v buňce
 - V podstatě „továrna“, ovlivněná vnitřními podmínkami a vnějším stavem (např. transport živin a signálů membránou, oddělené části uvnitř buňky (jádro, vlastní buněčný prostor, mitochondrie, ...))
- Materiálové vědy
 - Studium vlastností materiálů ze základních fyzikálních principů
 - Vnesení časové složky, sledování změn v čase (kvantová chronodynamika)
- ...

Příklad 6: Grafové algoritmy

- Reprezentují úlohy s enormními nároky na operace s pamětí
- Avšak velké množství reálných problémů má vyjádření v podobě grafových algoritmů
 - Problémy jsou často enormně veliké (miliony, miliardy a více vrcholů a hran)
 - Nalezení efektivních architektur pro grafové algoritmy bude mít přímý dopad na řadu oblastí (výzkum, obchod, průmysl, finance, ...)
- Využití ve vědě a průmyslu
 - Přímé řešení řídkých (sparse) problémů
 - Přenos tepla radiací
 - Generace mřížek
 - Plánování
 - Doprava
 - Bioinformatika, ...
- Informatics
 - Kódování entit a vztahů
 - Rozpoznávání obrazů (a subgrafů)

Charakteristika grafových algoritmů

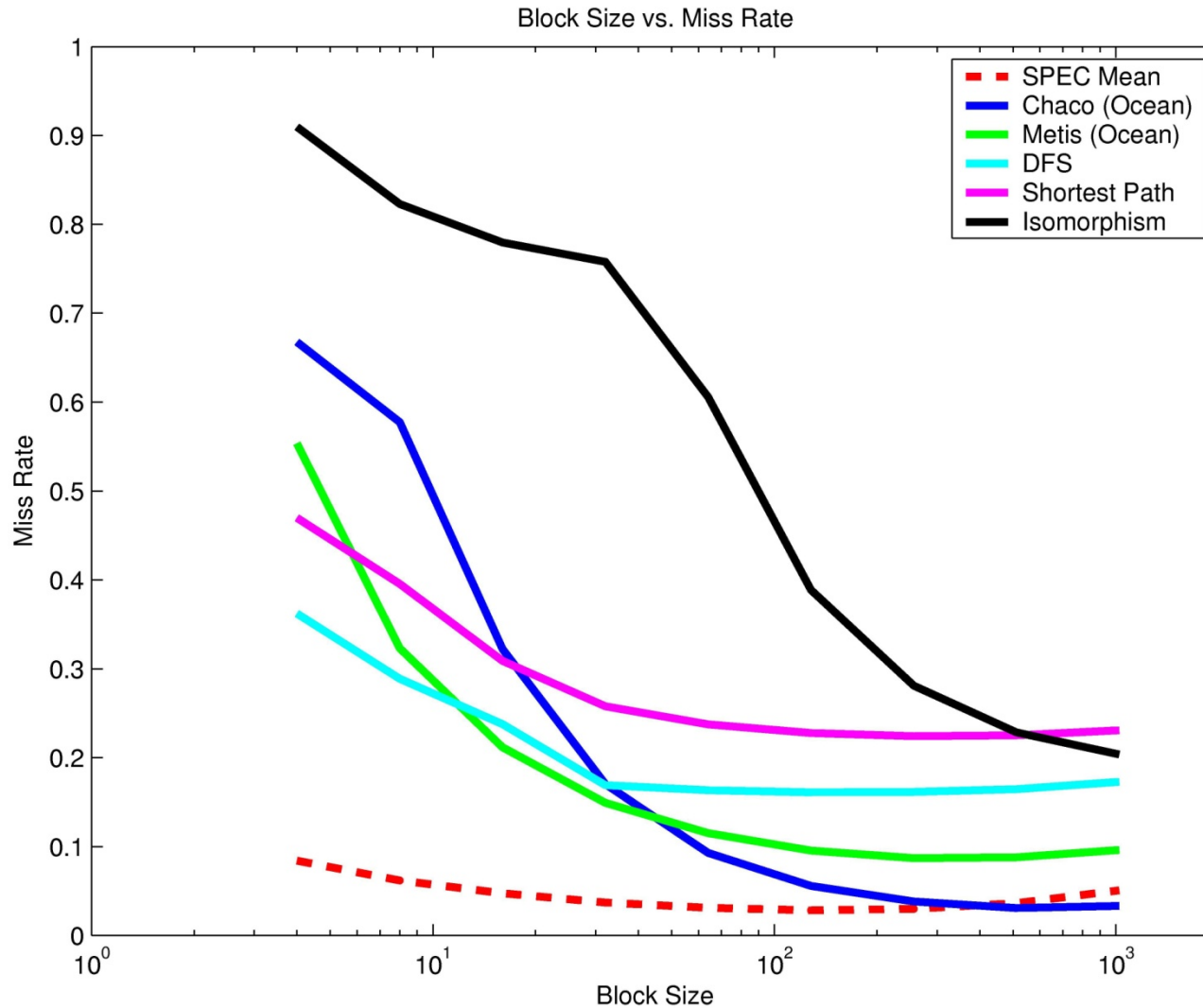
- **Struktura dat**
 - Více méně uspořádané pro vědecké/průmyslové aplikace
 - I nestrukturované mřížky lze mapovat na regulární grafy
 - Vysoce nestrukturované u informatických problémů
 - Podobné náhodným (power law) sítím
 - Prakticky je nejde rozdělit do menších částí
- **Vlastnosti algoritmů**
 - Zpravidla se prochází hranami
 - Často možné procházení více hran současně – potenciál pro paralelizaci
 - Vysoké nároky na paměťový subsystém
 - Náhodné přístupy do globální paměti, čtou/zapisují se zpravidla jen malé objemy dat
 - Přístupu lineární (výsledek aktuálního přístupu určuje pozici dalšího)
 - Zanedbatelný objem výpočtů

Vhodná architektura/model

- Nízká latence / vysoká propustnost
 - Nezbytná podpora malých zpráv
- Nenáročné mechanismy pro synchronizaci
- Tolerance latence
- Globální adresní prostor

- Není snadné upokojit současnými architekturami
 - Komunikaci nelze prokládat se zanedbatelnými výpočty
 - Cache/prefetching nepomáhají při náhodném přístupu k malému objemu dat

Příklad malého přínosu cache



Závěr

- Přes dosažený pokrok je stále více otevřených než vyřešených problémů
- Kombinace postupného pokroku se zcela novými přístupy
 - Z dlouhodobého pohledu jsou disruptivní (nové) přístupy a technologie důležitější
- Aplikace čekají
 - Ale k průlomu může dojít i u nich (nové modely a algoritmy)