

Strukturované a nestrukturované P2P sítě, DHT



Přednášky z Distribuovaných systémů
Ing. Jiří Ledvina, CSc.

Omezení modelu klient/server



- Těžko dosažitelná škálovatelnost
- Server představuje úzké místo systému z pohledu chyb
- Server vyžaduje administraci
- Nevyužívá okrajové zdroje

- P2P se snaží eliminovat tato omezení



Definice P2P

- P2P systémy jsou překryvné sítě na aplikační úrovni
- Nemají centrální řízení
- Definice: Je to technologie, která dovoluje spolupracovat dvěma nebo více členům v síti vzájemnou výměnnou odpovídající informací a s využitím komunikačních systémů bez nutnosti centrální koordinace.



Klient-server kontra P2P architektura

- Jednoduché rozlišení
 - Klient-server
 - Počítače se chovají asymetricky vzhledem k prováděným funkcím
 - Peer-to-peer systémy(P2P)
 - Počítače se chovají symetricky vzhledem k prováděným funkcím
- Odlišné architektury – různé výhody a nevýhody
- Čisté P2P systémy jsou raritou
 - Většina P2P sítí provádí některé funkce centrálně (adresáře), některé decentralizovaně (přístup k souborům)



Vlastnosti P2P sítí

- Klienti jsou současně servery i směrovače
 - Uzly obsahují paměť, CPU, úložiště dat
- Uzly jsou autonomní, bez administrativní autority
- Síť je dynamická, uzly se mohou často do sítě připojovat i odpojovat
- Uzly spolupracují vzájemně, nikoliv prostřednictvím nějakých veřejně známých serverů
- Uzly mohou mít velmi proměnlivé schopnosti (heterogenita)



Výhody P2P výpočtů

- P2P počítání je sdílení výpočetních zdrojů a služeb přímou výměnou mezi systémy
- Tyto zdroje a služby zahrnují výměnu informace, výpočetní cykly, vyrovnávací paměti a diskovou paměť pro soubory
- P2P počítání využívá existující výpočetní výkon, paměti a síťová propojení, dovolující uživatelům využívat jejich společný výpočetní výkon jako výhodu pro všechny



Výhody P2P sítí

- Efektivní využití zdrojů
 - Nevyužitá pásma, nevyužitá paměť, nevyužitý výpočetní výkon
- Škálovatelnost
 - Členové jsou si podobní, je možné přidat další členy, rozšíření na rozlehlé sítě
 - Členové jsou jak spotřebiteli zdrojů, tak i jejich zdroji
 - Agregace zdrojů zvyšuje přirozeně jejich využitelnost



Výhody P2P sítí

- Spolehlivost
 - Vytváření replik
 - Geografické rozmístění uzlů
 - Bez existence jednoho chybového místa
 - Internet ani např. web nemají centrální místo.
 - Mnoho internetových služeb používá model klient/server, ale jsou uspořádány tak, že nemají centrální chybové místo
- Jednoduchá administrace
 - Uzly se organizují sami
 - Není třeba rozmisťovat servery tak, aby zajišťovaly průchodnost systému
 - Automaticky zabudovaná odolnost proti chybám, replikace, vyrovnávání zátěže



Hybridní architektura P2P sítě

- Architektura server/klient
 - Slabá místa z hlediska centralizovatelnosti
- Architektura P2P
 - Slabá místa z hlediska distribuovaného řízení
- Hybridní architektura
 - Částečně centralizovaná – řízení
 - Decentralizovaná – škálovatelnost, zálohování, rozdělování zátěže, ...

24.3.2008

Distribuované systémy



Vývoj služeb Internetu

- První generace – čistý Internet
 - Využití přenosových cest
 - Zpracování dat v koncových uzlech
- Druhá generace – webové služby (překryvná síť)
 - Využívání transparentních modulů v síti (proxy, cache, firewall, ...)
- Třetí generace – vytváření nových služeb pro uživatele
 - Využití nevyužitých výpočetních zdrojů sítě v koncových uzlech (ne pouze přenosových cest)
 - Jednoduché zavádění nových služeb
 - Není nutný složitý management
- Na P2P sítě se můžeme dívat jako na novou generaci Internetu

24.3.2008

Distribuované systémy



Klasifikace P2P systémů

- Centralizované systémy – členové jsou připojeni k jednomu serveru, který koordinuje jejich činnost a řídí komunikaci
- Systémy s agenty (brokers) – členové se připojují k serveru aby zjistily ostatní členy, ale samotná komunikace už servery řízena není (brokered P2P síť)
- Decentralizované systémy – členové pracují nezávisle na centrálních službách. Vyhledávání členů je decentralizované a komunikace probíhá mezi členy (Gnutella, Freenet)

24.3.2008

Distribuované systémy



Na co se P2P síť hodí

- Vytváření „sítí“ pro specifické uživatele
 - Snadný a rychlý přístup k informacím
- Vyhledávací stroje
 - Vyhledávání informací v limitovaném (předem daném) prostoru
 - Zvýšení efektivity i rychlosti obnovy informací
- Společný vývoj
 - Společný vývoj programů
 - Společné vytváření dokumentů
 - Spolupráce při složitých výpočtech

24.3.2008

Distribuované systémy



Oblasti aplikací P2P sítí

- Komunikace (AOL Instant Messenger, ICQ)
- Vzdálená spolupráce (sdílené editování souborů, audio a video konference, sdílené kreslení)
- Interaktivní hry (DOOM)
- Streaming (multicast na aplikační úrovni)
- Distribuované výpočty (SETI@home)
- Sdílení souborů (Napster, Gnutella, Freenet, KazaA, ...)
- Ad-hoc sítě

24.3.2008

Distribuované systémy



Sdílení souborů a streaming

- Sdílení souborů
 - Stažení celého souboru a poté jeho zpracování
 - Malé soubory, krátký čas stahování
 - Soubor uložen u jednoho člena – jedno spojení
 - Bez časové krize
- Streaming
 - Zpracování během stahování
 - Velké soubory – dlouhý čas stahování
 - Soubor uložen u několika členů – více spojení
 - Čas je kritický

24.3.2008

Distribuované systémy

Řešené problémy P2P sítí



- Identifikace členů
- Směrovací protokoly
- Topologie sítí
- Vyhledávání členů
- Protokoly pro koordinaci a komunikaci
- Protokoly pro zajištění kvality služeb (QoS)
- Bezpečnost (pasivní napadení, aktivní napadení)
- Sdílení a přidělování zdrojů

24.3.2008

Distribované systémy

Topologie P2P sítí



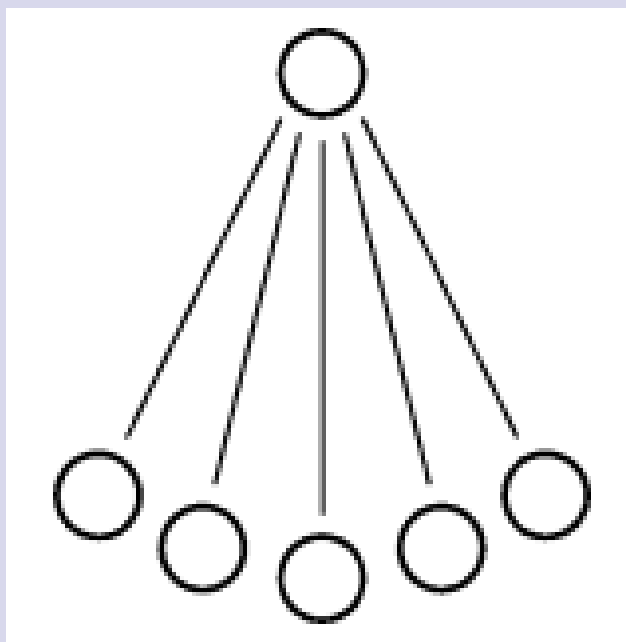
- Centralizovaná topologie
- Kruhová topologie
- Hierarchické uspořádání
- Decentralizované uspořádání
- Hybridní uspořádání

24.3.2008

Distribované systémy



Centralizovaná topologie

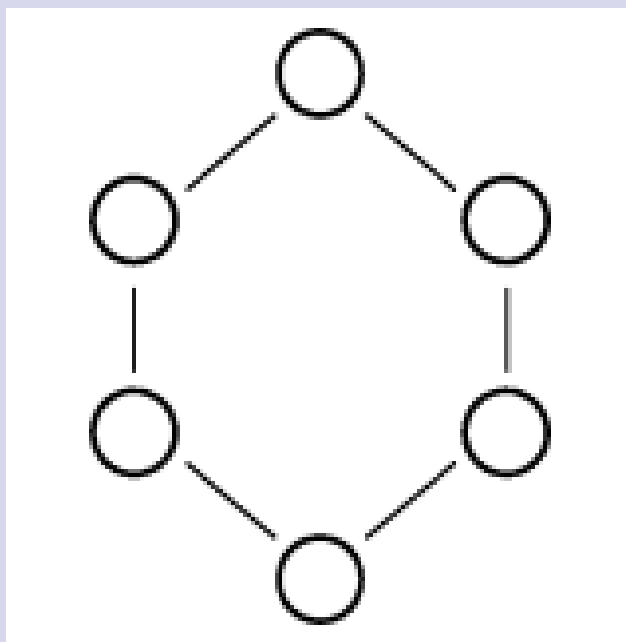


24.3.2008

Distribované systémy



Kruhová topologie

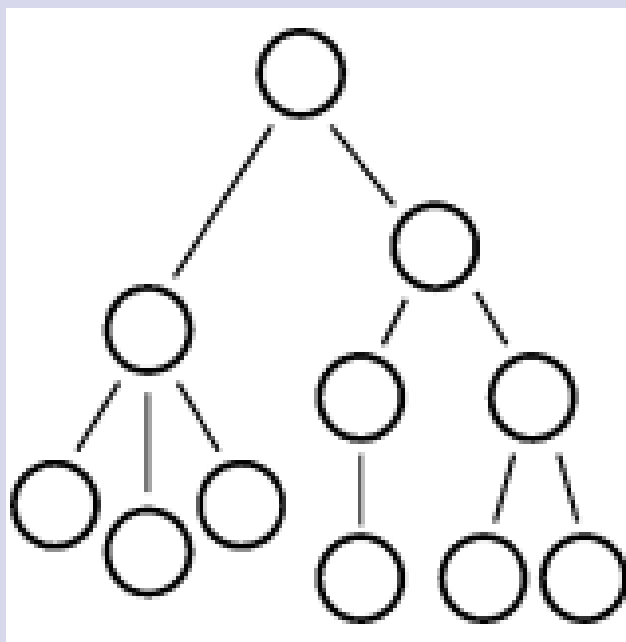


24.3.2008

Distribované systémy



Hierarchická topologie

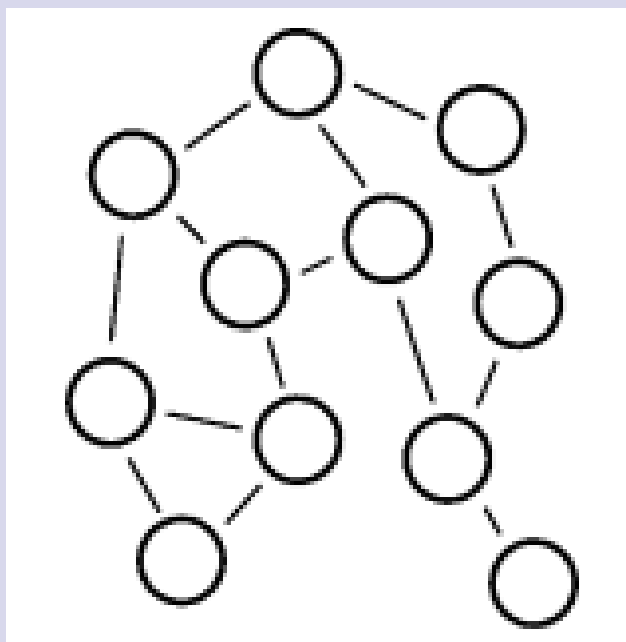


24.3.2008

Distribované systémy



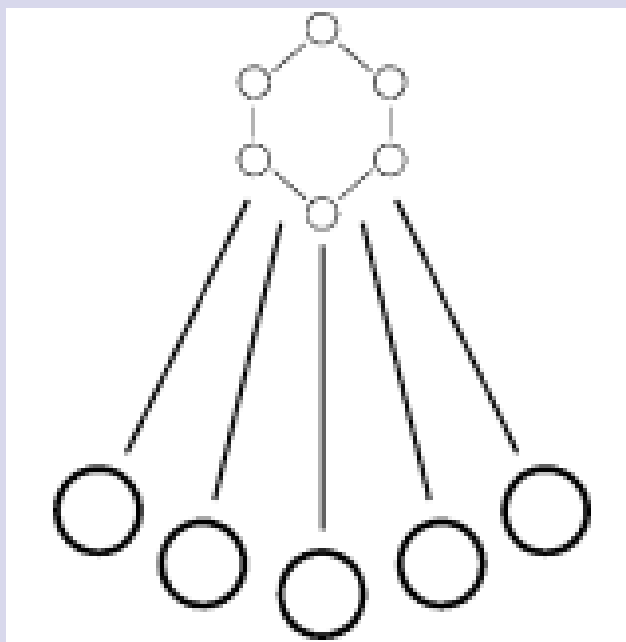
Decentralizovaná topologie



24.3.2008

Distribované systémy

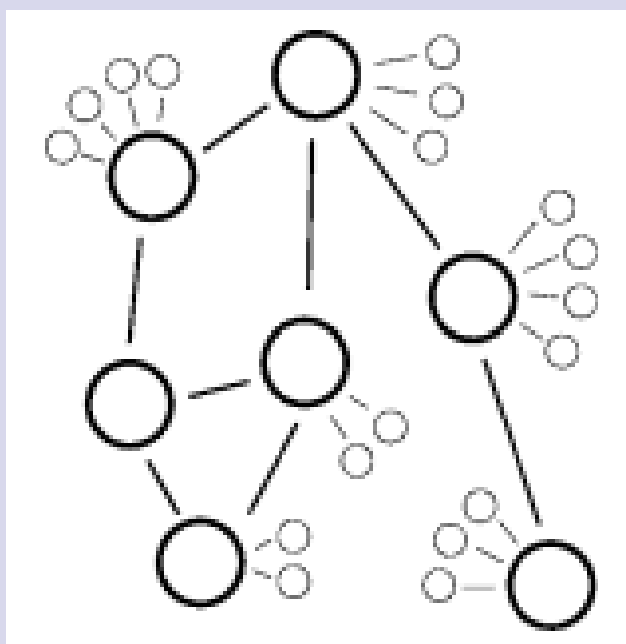
Hybridní topologie (centralizovaná a kruh)



24.3.2008

Distribuované systémy

Hybridní topologie (centralizovaná a decentralizovaná)



24.3.2008

Distribuované systémy



Hodnocení topologií

- Ovladatelnost
 - Pracnost udržet topologii v provozu
- Informační soudržnost
 - Spolehlivost dodané informace
- Rozšiřitelnost
 - Snadnost rozšíření
- Odolnost proti chybám
 - Snadnost zpracování chyb

24.3.2008

Distribuované systémy



Hodnocení topologií

- Odolnost proti politickým nebo právním vlivům
 - Nový fenomén
 - Jak je obtížné činnost sítě ukončit
- Bezpečnost
 - Jak je těžké síť zničit
- Škálovatelnost
 - Jak dalece může být zvětšována

24.3.2008

Distribuované systémy



Nestrukturované sítě

- Napster
- Gnutella
- Kazaa
- Freenet

24.3.2008

Distribuované systémy



Napster

- 5/1999 – 2001 (soud), 2002 - konec
- sdílení hudebních souborů
- centralizované řešení
- uživatelé zapíší seznam svých souborů do Napster serveru
- další uživatelé pošlou požadavek se seznamem požadovaných souborů do Napster serveru (vyhledávání podle klíčů)
- Napster server pošle seznam IP adres počítačů, které soubory obsahují
- Uživatel se připojí přímo k vybranému uživateli a soubor stáhne

24.3.2008

Distribuované systémy

Napster

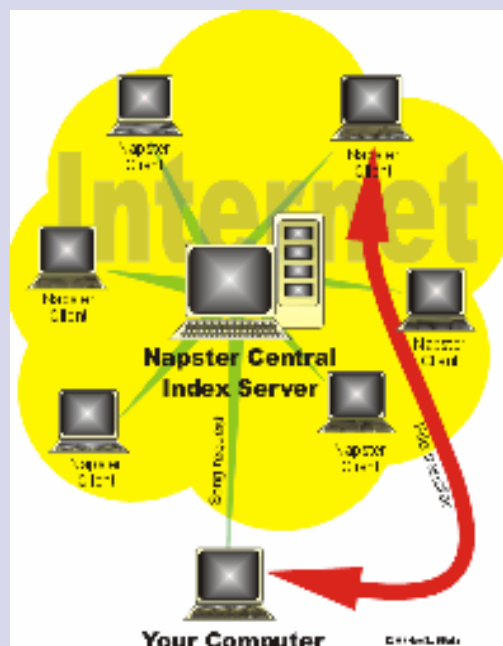


- Centrální Napster server
 - Zajišťuje správné odpovědi
 - Je však úzkým místem z hlediska škálovatelnosti
 - Je také úzkým místem z hlediska chyb
 - Citlivý na DoD
- Prohledávání je centralizované
- Přenos souborů je přímý (peer-to-peer)

24.3.2008

Distribuované systémy

Napster



24.3.2008

Distribuované systémy

Gnutella



- dovede sdílet libovolné soubory
- na rozdíl od Napster provádí decentralizované prohledávání
- dotazuje se sousedů
- sousedé se dotazují svých sousedů, atd.
- počet úrovní prohledávání je dán TTL
- uzly, které obsahují vyhledávané soubory odpoví

24.3.2008

Distribuované systémy

Gnutella



- decentralizované prohledávání
 - odstraněn problém místa citlivého na chyby
 - není tak citlivý na DoS
 - nemůže zaručit správné výsledky
- záplavové dotazování
 - prohledávání je distribuované, ale dosud ne škálovatelné
- Stahování pomocí HTTP GET request

24.3.2008

Distribuované systémy



Gnutella

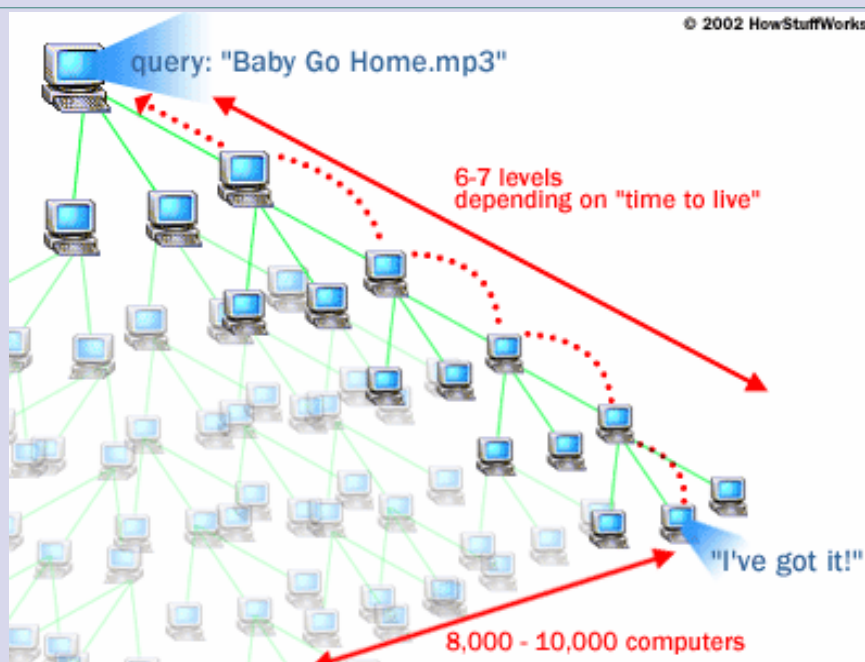
- Zprávy
 - Advertisement
 - Query
 - Push Request
- Formát zprávy
 - Message ID (16) - párování
 - Function ID (1) – typ
 - TTL (1)
 - Hops (1)
 - Payload Length (4)

24.3.2008

Distribované systémy



Gnutella



24.3.2008

Distribované systémy



Kazaa (sít' FastTrack)

- hybrid centralizované Napster a decentralizované Gnutella
- super-peer vystupují jako lokální centra vyhledávání
 - každý super-peer je obdobou Napster serveru pro malou část sítě
 - super-peer jsou vybírány systémem automaticky na základě jejich parametrů (paměť, šířka pásma, ...) a dostupnosti (čas pro připojení)

24.3.2008

Distribované systémy



Kazaa (sít' FastTrack)

- uživatelé přenesou svůj seznam souborů do super-peer
- super-peer si periodicky vyměňují seznam souborů
- uživatelé posílají dotazy do super-peer
- systém slouží ke sdílení souborů
- existuje možnost nedsílet data, pouze je stahovat

24.3.2008

Distribované systémy

Anonymita P2P sítí



- Napster, Gnutella ani Kazaa nezajišťují anonymitu
 - Uživatelé vědí kde co je a kdo co požaduje
- Freenet
 - Navržen mimo jiné k zajištění anonymity

24.3.2008

Distribované systémy

Freenet



- 6/1999
- data jsou přenášena v opačném směru než dotaz
 - není možné zjistit, je-li uživatel iniciátorem nebo pouze data přenáší dál
 - není možné zjistit, jestli uživatel data posílá dál nebo je také spotřebovává
- chytré dotazy
 - požadavky jsou směrovány do správného uzlu postupně

24.3.2008

Distribované systémy

Freenet

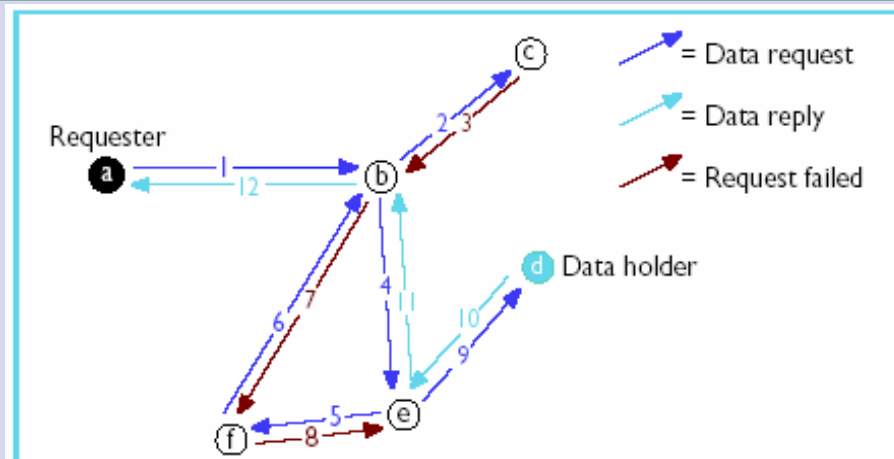


Figure 1. Typical request sequence. The request moves through the network from node to node, backing out of a dead-end (step 3) and a loop (step 7) before locating the desired file.

24.3.2008

Distribované systémy

Freenet



- Nestrukturovaná P2P síť
- P2P indexovací a vyhledávací služba
- P2P stahování souborů
- Soubory jsou obsluhovány po stejných cestách jako vyhledávání (nepoužívá přímá propojení koncových bodů)
 - Zavedeno kvůli zachování anonymity

24.3.2008

Distribované systémy

Freenet



- Kompletně anonymní pro zdroje i konzumenty informací
- Odolný vůči pokusům třetí strany zabránit přístupu k informacím
- Cíle
 - Anonymita pro poskytovatele i konzumenty
 - Možnost zapírání pro úložiště dat
 - Odolnost proti útokům na znepřístupnění služeb
 - Efektivní ukládání dat a směrování
 - Nezajišťuje
 - Trvalé ukládání souborů
 - Vyrovnávání zátěže

24.3.2008

Distribuované systémy

Mechanismus vyhledávání souborů: Chain Mode

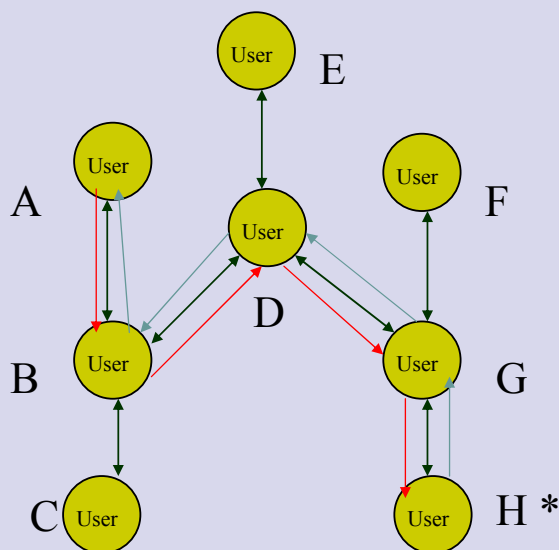


- Požadavek na vyhledání souboru může být poslán mnoha různým uzlům
- Jestliže uzel požadovaný dokument nemá, posílá požadavek některému ze svých sousedů, u kterého je pravděpodobnější, že dokument má
 - Zprávy vytváří řetězec spojující uzly, přes které byl dotaz přenášen
- Zprávy jsou rušeny po přenosu přes předem daný počet mezilehlých uzlů, takže délka řetězce je omezená
- Řetězec končí pokud je vyčerpán počet opakování nebo pokud je soubor nalezen

24.3.2008

Distribuované systémy

Mechanismus vyhledávání souborů: Chain Mode



Počítač A pošle dotaz svému sousedu B, který jej pošle svému sousedu D, který jej dále pošle sousedovi G a nakonec dotaz obdrží H, který má požadovaná data

Odpověď je posílána zpět přes všechny uzly, které přenášely dotaz

24.3.2008

Distribované systémy

Výhody a nevýhody tohoto režimu pro prohledávání



- Výhody
 - Pro průměrný případ rychlé vyhledávání s minimálním zatížením sítě
 - Vyhledávání je ukončeno jakmile je soubor nalezen
 - Dobře škálovatelné
- Nevýhody
 - Pro nejhorší případ pomalé vyhledávání
 - Soubor nemusí být nalezen – náhodný výběr dalšího souseda

24.3.2008

Distribované systémy



Operace Freenet

- Dotazy jsou posílány jednomu serventu (server/klient). Pokud se neobjeví kladná odpověď z tohoto řetězu dotazů, je vybrán jiný servent.
- Avšak po dané cestě je poslán celý nalezený dokument i když může být hodně velký
- Servent(y) mají vyrovnávací paměti, do kterých často vyhledávané dokumenty ukládají na omezenou dobu. Pokud není dokument po delší dobu požadován, je vymazán



Dosažení anonymity

- Jak se dosáhne anonymity
 - Uzly posílají požadavky náhodně a nelze rozpoznat jsou-li zdrojem nebo cílem požadavku
 - Hodnoty doby života požadavku jsou neurčité
 - Není možné vysledovat dokument do jeho počátečního uzlu
 - Podobně není možné vysledovat ve kterém uzlu byl dokument vložen



Strukturované P2P sítě

- Chord
- Pastry
- CAN
- BitTorrent

24.3.2008

Distribuované systémy



Klasifikace P2P systémů pro sdílení souborů

- Hybridní (se zprostředkováním pomocí brokera)
 - Nestrukturované a centralizované
 - Napster
 - Nestrukturované a se znalostí super člena (super peer)
 - KazaA, Morpheus
- Nestrukturované decentralizované (volně řízené)
 - Soubory mohou být kdekoliv
 - Podporuje částečná jména a dotazování podle klíče
 - Neefektivní prohledávání (heuristické metody)
 - Negarantuje nalezení informace
 - Gnutella

24.3.2008

Distribuované systémy

Klasifikace P2P systémů pro sdílení souborů

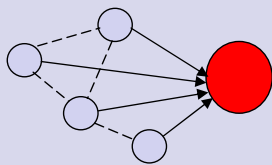


- **Strukturované** (pevně řízené, DHT)
 - Soubory jsou přiřazeny specifickým uzlům podle přísných pravidel
 - Lze zajistit efektivní prohledávání a garantovat nalezení souboru
 - Postrádá částečná jména a vyhledávání podle klíčových slov
 - Příklady – Chord, CAN, Pastry, Tapestry

24.3.2008

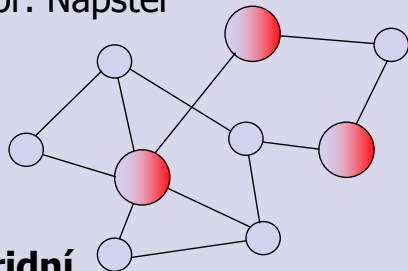
Distribuované systémy

Porovnání některých metod pro sdílení souborů



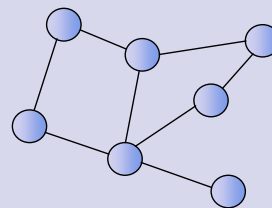
Centralizované

- jeden nebo pár koordinátorů
- např. Napster



Hybridní

- „Super peers“ obsahují směrovací informace
- např. FastTrack (Kazaa, Morpheus), odvozeniny z Gnutella



Zcela decentralizované

- všichni členové (nebo žádný) obsahují směrovací informace e. g. Freenet, Gnutella

24.3.2008

Distribuované systémy



Strukturované P2P sítě

- Druhá generace P2P překryvných sítí
- Samo se organizující
- Podporují vyrovnávání zátěže
- Odolné proti poruchám
- Garantují maximální počet přeskoků při dotazování
 - Hlavní rozdíl od nestrukturovaných systémů
- Založené na distribuovaných hashovacích tabulkách
 - Definují jednoduché rozhraní pro přístup
 - Umožňují oddělit vyhledávací metody od metod přenosových

24.3.2008

Distribuované systémy



Distribuované hashovací tabulky (DHT)

- Distribuovaná verze datové struktury hashovací tabulka
- Ukládá pár (klíč, hodnota)
 - Klíč odpovídá jménu souboru
 - Hodnota odpovídá obsahu souboru
- Cíl: vytvořit efektivní operace pro vkládání, vyhledávání a rušení položek (klíč, hodnota)
- Každý člen ukládá u sebe podmnožinu párů (klíč, hodnota)
- Základní operace: nalezení uzlu který odpovídá klíči
 - Mapování klíče na uzel
 - Efektivně směřovat požadavky (vložit, vyhledat, vymazat) k tomuto uzlu

24.3.2008

Distribuované systémy

Aplikace DHT



- Nad rozhraním DHT může být vybudováno mnoho služeb
 - Sdílení souborů
 - Archivní úložiště souborů
 - Databáze
 - Vyhledávání služeb podle jmen
 - Chatovací služby
 - Komunikace založená na randevous
 - Publikování

Požadované vlastnosti DHT



- Klíče jsou rovnoměrně mapovány na všechny uzly sítě
- Každý uzel udržuje informaci pouze o malém množství ostatních uzlů
- Zprávy mohou být do uzlů směrovány efektivně
- Přidání nebo odebrání uzlu má vliv pouze na několik uzlů

Směrovací protokoly DHT



- DHT je pouze základní rozhraní
- Existuje několik implementací toho rozhraní
 - Chord [MIT]
 - Pastry [Microsoft Research]
 - CAN (Content Addressable Network) [Berkeley]
 - SkipNet [Microsoft Research]
 - Kademia [New York University]
 - Viceroy [Israel, UC Berkeley]
 - P-Grid [EPFL Switzerland]
 - Freenet [Ian Clarke]

24.3.2008

Distribuované systémy

Strukturované překryvné sítě



- Vlastnosti
 - Mají topologii determinovanou přesnými pravidly
 - Dobře definovaná pravidla určují ke kterým uzlům bude uzel připojen
 - Soubory jsou ukládány do přesně definovaných míst
 - Hashovací funkce mapuje jména souborů na uzly (reprezentované síťovými adresami)
 - Škálovatelnost směrování je založena na attributech vyhledávání

24.3.2008

Distribuované systémy

Chord



- Chord
 - Zajišťuje P2P vyhledávání pomocí hashovací funkce
 - Převádí Lookup(key) → IP adresa
 - Chord neukládá data
 - Efektivnost $O(\log N)$ zpráv pro jedno vyhledání
 - N je celkový počet serverů
 - Škálovatelnost: $O(\log N)$ stavů pro uzel
 - Rozsáhlé změny při změně členství

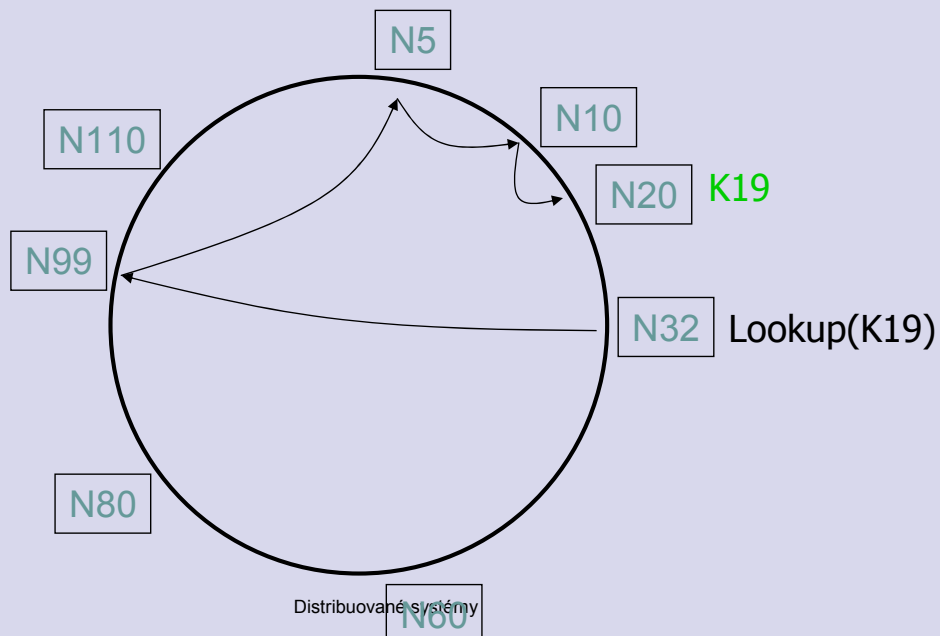
24.3.2008

Distribuované systémy

Chord: Lookup Mechanism



- Lookups take $O(\log N)$ hops



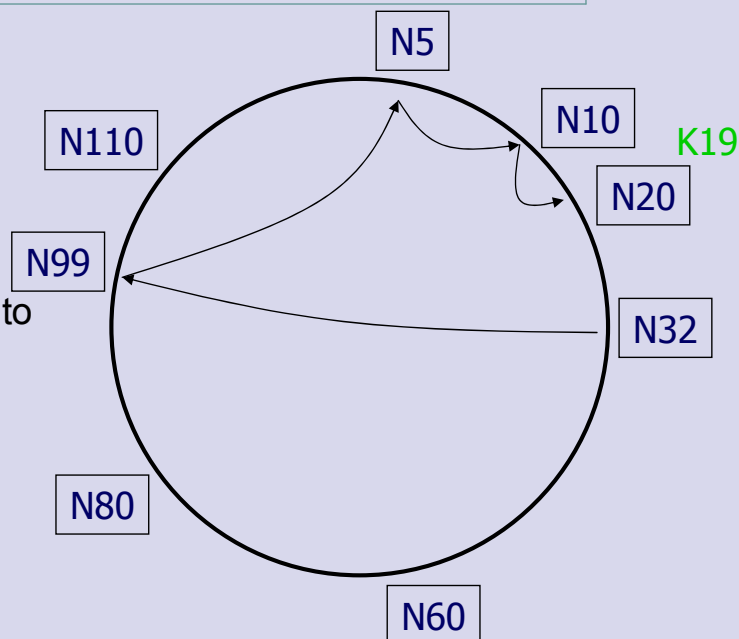
24.3.2008

Distribuované systémy

Document Routing – Chord



- MIT project
- Uni-dimensional ID space
- Keep track of $\log N$ nodes
- Search through $\log N$ nodes to find desired key



24.3.2008

Distribované systémy

Výpočet ID



- m bit prostor identifikátorů pro klíče i uzly
- identifikátor klíče = $\text{SHA-1}(\text{key})$
Key="LetItBe" $\xrightarrow{\text{SHA-1}}$ ID=60
- identifikátor uzlu = $\text{SHA-1}(\text{IP address})$
IP="198.10.10.1" $\xrightarrow{\text{SHA-1}}$ ID=123
- oba jsou distribuovány stejně
- problém mapování ID klíčů na ID uzlů

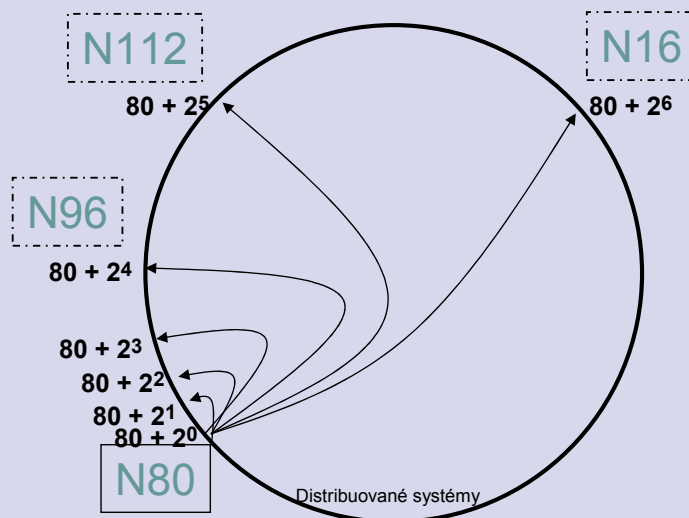
24.3.2008

Distribované systémy

Vyhledávací tabulky (Finger tables)



- každý uzel zná m ostatních uzlů v kruhu
- vzdálenost se zvětšuje exponenciálně

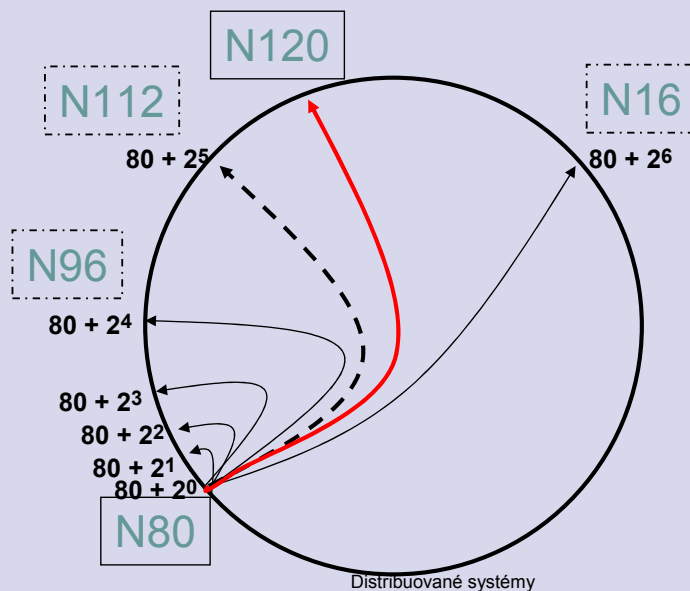


24.3.2008

Vyhledávací tabulky (Finger tables)



- ukazatel i ukazuje na následníka $n+2^i$

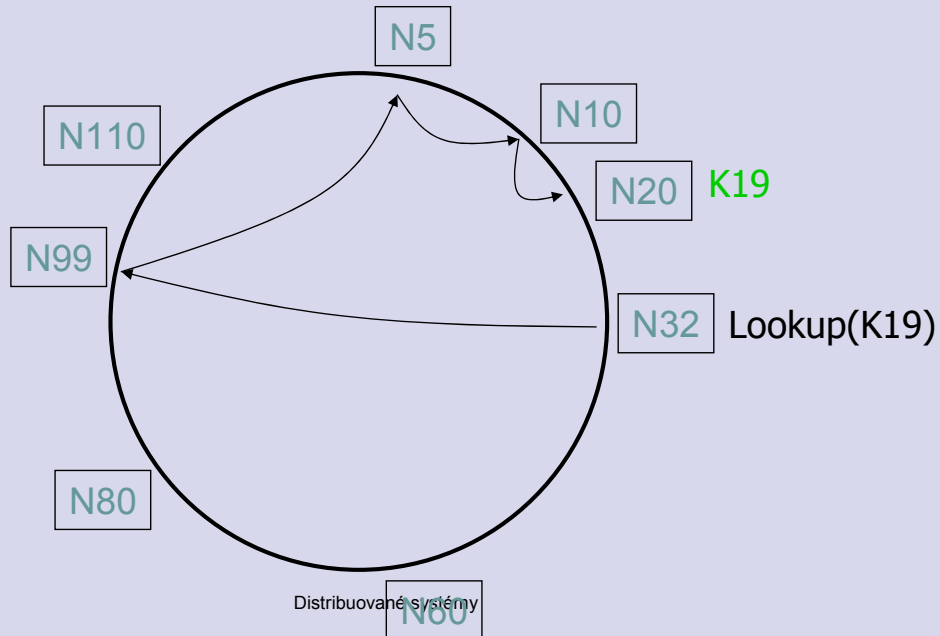


24.3.2008

Vyhledávací tabulky (Finger tables)



- vyhledávání představuje $O(\log N)$ kroků



24.3.2008

Distribuované systémy

Vyhledávací tabulky - připojení se do kruhu



- Agresivní algoritmus:
 - inicializace všech položek v uzlu
 - oprava položek v existujících uzlech
 - přenos klíčů z následníka do uzlu
- Méně agresivní algoritmus (oprava na pozadí):
 - inicializace položky na následující uzel
 - periodická verifikace bezprostředního následovníka a předchůdce
 - periodická oprava položek tabulky

24.3.2008

Distribuované systémy

Pastry



- rozhraní podobné jako Chord
- předpokládá lokalizaci sítě pro minimalizaci přeskoků
- nový uzel potřebuje znát nejbližší uzel, aby bylo dosaženo lokality

Content-Addressable Network (CAN)



- Typická metoda pro vyhledání dokumentu v síti
- Virtuální kartézský prostor pro vyhledávání
- Celý prostor je rozdělen mezi všechny uzly
 - Každý uzel vlastní zónu jako část celkového prostoru
- Abstrakce
 - Data ukládáme do bodů prostoru
 - Mezi libovolnými body prostoru můžeme vytvářet cesty
- Bod = uzel který vlastní určitou oblast

Základní koncepce CAN



- Data uložená v CAN jsou adresována podle jména (tj. klíče), ne podle svého uložení (tj. IP adresy)
- Úlohou směrování je nalézt místo uložení dat

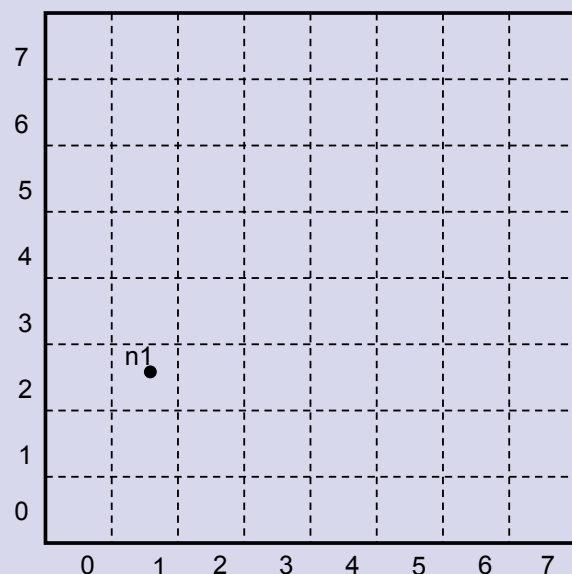
24.3.2008

Distribuované systémy

CAN Example: Two Dimensional Space



- Space divided between nodes
- All nodes cover the entire space
- Each node covers either a square or a rectangular area of ratios 1:2 or 2:1
- Example:
 - Node n1:(1, 2) first node that joins → cover the entire space



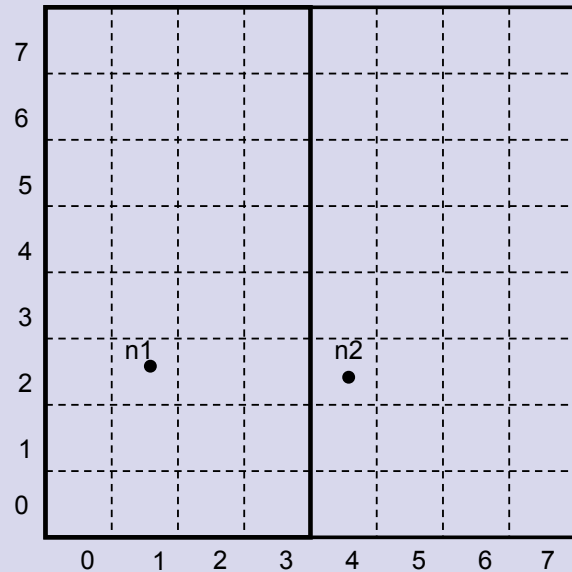
24.3.2008

Distribuované systémy

CAN Example: Two Dimensional Space



- Node $n2:(4, 2)$ joins \rightarrow space is divided between $n1$ and $n2$



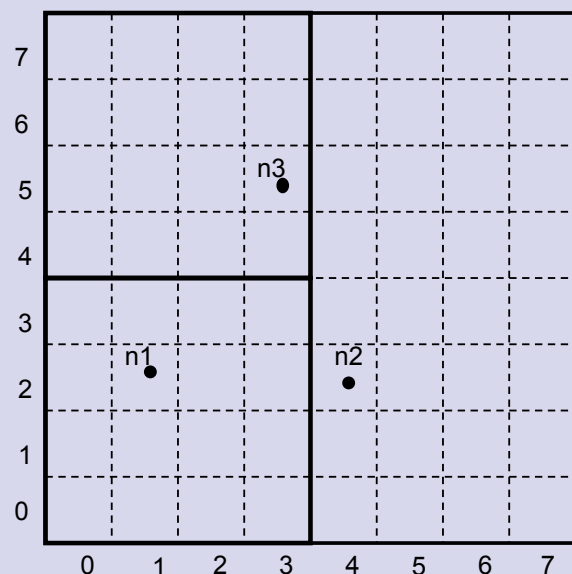
24.3.2008

Distribované systémy

CAN Example: Two Dimensional Space



- Node $n3:(3, 5)$ joins \rightarrow space is divided between $n1$ and $n3$



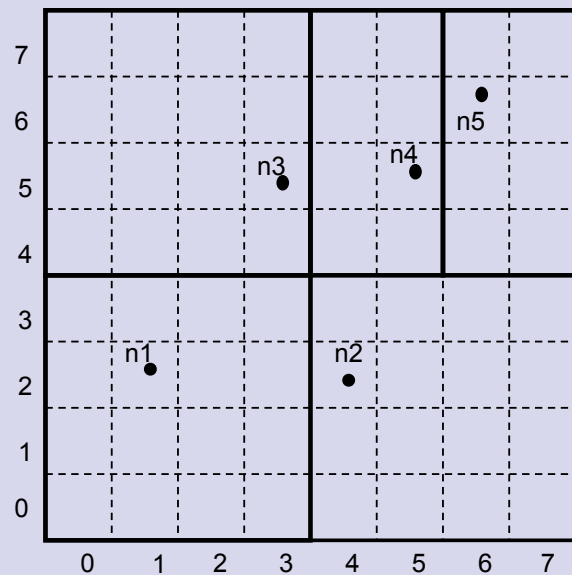
24.3.2008

Distribované systémy

CAN Example: Two Dimensional Space



- Nodes $n_4:(5, 5)$ and $n_5:(6,6)$ join



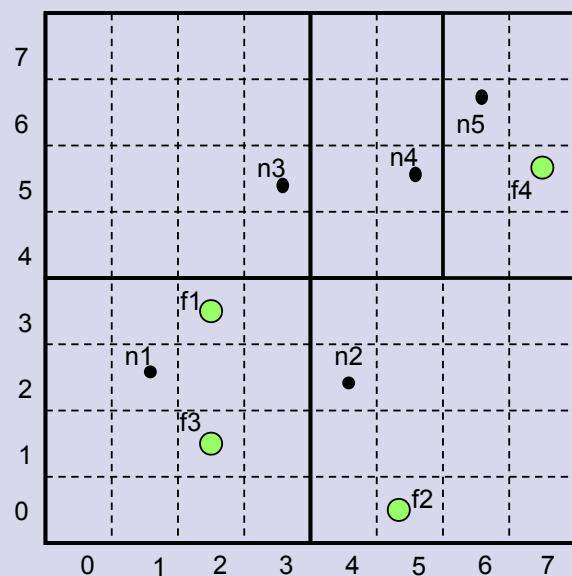
24.3.2008

Distribované systémy

CAN Example: Two Dimensional Space



- Nodes: $n_1:(1, 2)$; $n_2:(4,2)$; $n_3:(3, 5)$; $n_4:(5,5)$; $n_5:(6,6)$
- Items: $f_1:(2,3)$; $f_2:(5,1)$; $f_3:(2,1)$; $f_4:(7,5)$;



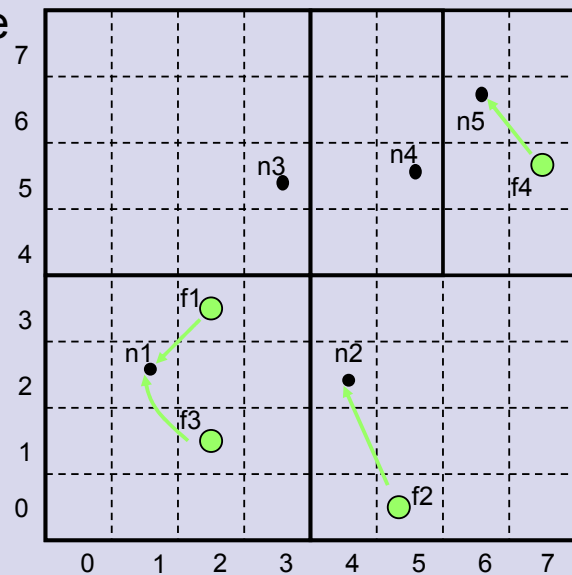
24.3.2008

Distribované systémy

CAN Example: Two Dimensional Space



- Each item is stored by the node who owns its mapping in the space



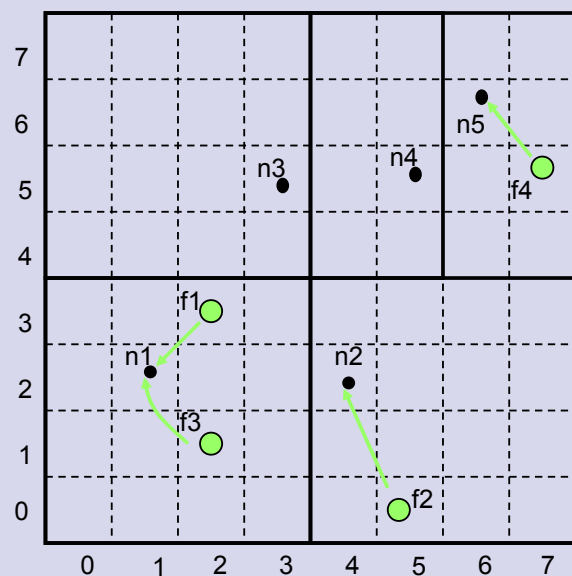
24.3.2008

Distribované systémy

CAN: Query Example



- Each node knows its neighbours in the d -space
- Forward query to the neighbour that is closest to the query id
- Example: assume Node n1 queries File Item f4



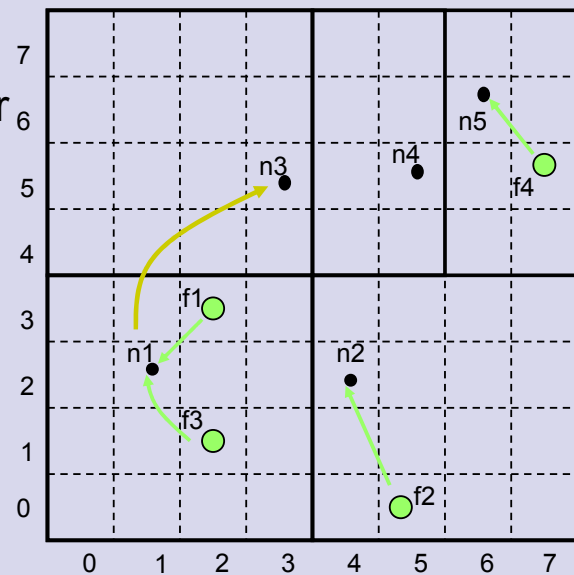
24.3.2008

Distribované systémy



CAN: Query Example

- Each node knows its neighbours in the d -space
- Forward query to the neighbour that is closest to the query id
- Example: assume Node $n1$ queries File Item $f4$



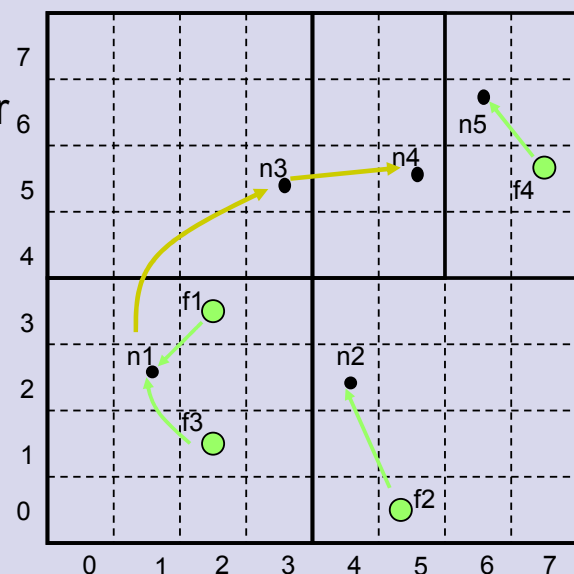
24.3.2008

Distribované systémy



CAN: Query Example

- Each node knows its neighbours in the d -space
- Forward query to the neighbour that is closest to the query id
- Example: assume Node $n1$ queries File Item $f4$



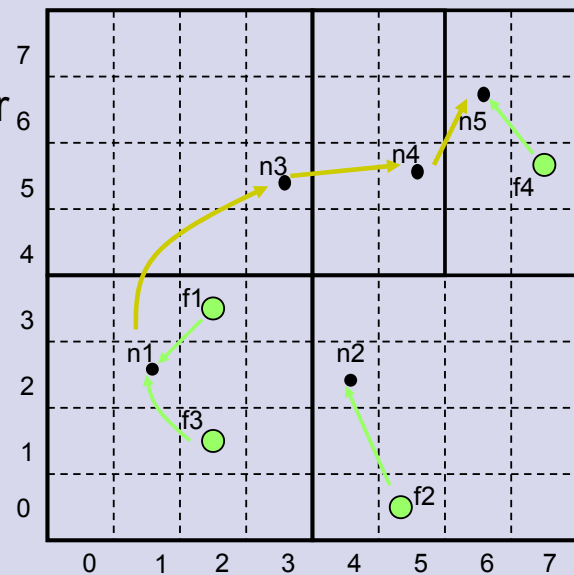
24.3.2008

Distribované systémy

CAN: Query Example



- Each node knows its neighbours in the d -space
- Forward query to the neighbour that is closest to the query id
- Example: assume Node $n1$ queries File Item $f4$



24.3.2008

Distribované systémy

Bit Torrent



- Bram Cohen
- TCP pro přenosy
- Dělení souboru na kousky (16kB)
 - Každý kousek zabezpečen SHA-1
- Stahování přes Web server
 - Supernova.org (+zrcadla)
 - Obsahuje metadata soubor (.torrent)
 - SHA-1 pro všechny kousky souboru
 - Mapování kousků do souboru
 - Odkaz na tracker (umístění částí souborů)

24.3.2008

Distribované systémy

Bit Torrent



- Seed (zdroj) – vytváří .torrent soubor
- Tracker – centrální server udržující seznam členů ve swarmu
- Swarm – soubor uzlů zúčastněných na distribuci souborů
- Člen se spojí se SWARMem aby získal TRACKER a z něho seznam členů, ke kterým se pak připojí

24.3.2008

Distribované systémy

Bit Torrent



- Překrývání částí souborů
 - Rovnoměrné překrytí
 - Dobré využití šířky pásma
 - Záloha souborů, distribuované kopie
 - Náhodný výběr člena pro stahování
 - Odolný vůči výpadkům členů

24.3.2008

Distribované systémy