Strukturno PP, II deo:

- Šabloni rukovanja podacima
- Drugi šabloni

Šabloni serijskog rukovanja podacima

- Šabloni serijskog rukovanja podacima su:
  - Slučajno čitanje i pisanje
  - Dodela steka
  - Dodela memorije (npr. C++ new/delete)
  - Funkcijski objekti
  - Objekti

Slučajno čitanje i pisanje

- Apstrakcija memorije kao niza lokacija
  - Pristup preko adresa, tj. pokazivača
- ALIASI: pokazivači koji pokazuju na isti objekat
- PROBLEM ALIASA, npr. args f-ije engine2
  - Otežava vektorizaciju i paralelizaciju programa
  - Dodatne kopije podataka – ponekad je preskupo
  - Aliasi se zabranjuju – odgovornost na programeru
- Indeksi nizova: malo sigurniji pristup
  - Mogući aliasi, ali oblast memorije je bolje ograničena
  - Drugo: niz se može lakše prebaciti u drugi procesor

Dodela steka

- Stek: prostor za dinamičko smeštanje podataka
  - LIFO organizacija
- Prednosti ove dodele:
  - Efikasna (u konstantnom vremenu)
  - Očuvava lokalnost podataka
- Paralelizacija ovog šablona:
  - Svakoj programskoj niti njen sopstveni stek
  - Cilk Plus generalizuje dodelu steka, u kontekstu funkcijskih poziva, radi očuvanja lokalnosti podataka
  - Tako generalizovan stek se naziva KAKTUS STEK

Funkcijski objekti (FO)

- Funkcijski objekti: rukovanje kao sa podacima
- LAMBDA FUNKCIJE su vrsta funkcijskih objekata
  - Definišu se tamo gde, i onda kada, su potrebni
  - Puno se koriste od strane TBB-a
- Često se koriste za skladištenje stanja referenciranih promenljivih
- FO mogu biti generisani:
  - Statički (jedan nivo indirekcije radi pristupa) ili
  - Dinamički (u ArBB stanje referenciranih prom. u tački konstruisanja – radi optimizacije koda)

Objekti

- Objekti pridružuju podatke funkcijama
  - METODE, ili FUNKCIJE ČLANICE objekta
  - PODKLASE (subclasses), NADKLASE (superclasses)
- C++ podklasa može redefinisati f-ije iz nadklase
  - Ovo redefinisanje zahteva upotrebu pokazivača
  - Pokazivači nisu uvek raspoloživi (stari GPU ih nema)
- Java „synchronized" metode:
  - Implicitno se dodaje brava
  - Brava može negativno uticati na performansu

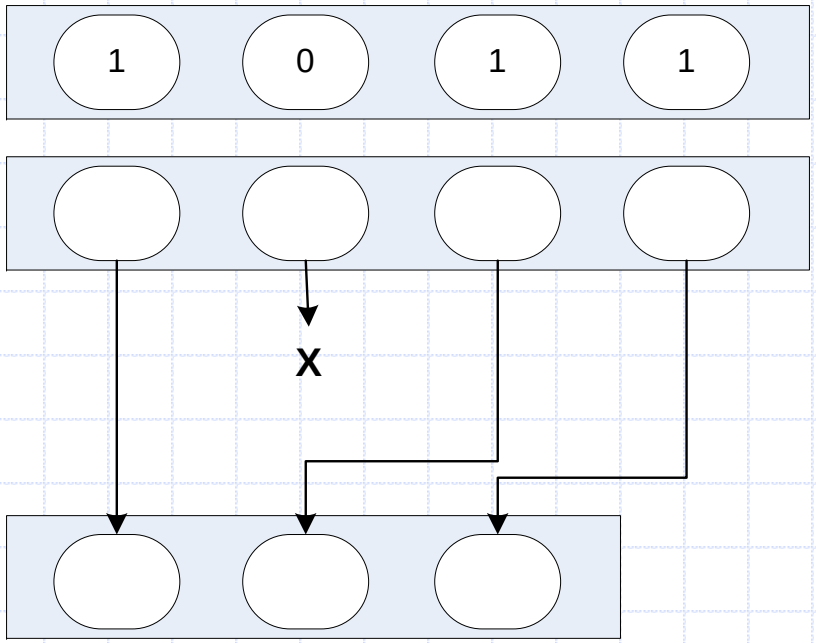
Šabloni paralelnog rukovanja podacima (1/2)

- Problemi, npr. trka do podataka
- Izbegava se menjanje deljenih podataka
  - Izuzetak je šablon Razbacivanje, ali bez problema
- Očuvanje lokalnosti podataka (LP):
  - LP je cilj nekih šablona, npr. šablon Particionisanje
  - Podaci se u nezavisnim zonama mogu bez opasnosti paralelno menjati

Šabloni paralelnog rukovanja podacima (2/2)

- Šabloni paralelnog rukovanja podacima su:
  - Pakovanje
  - Protočna obrada
  - Geometrijska dekompozicija
  - Skupljanje
  - Razbacivanje

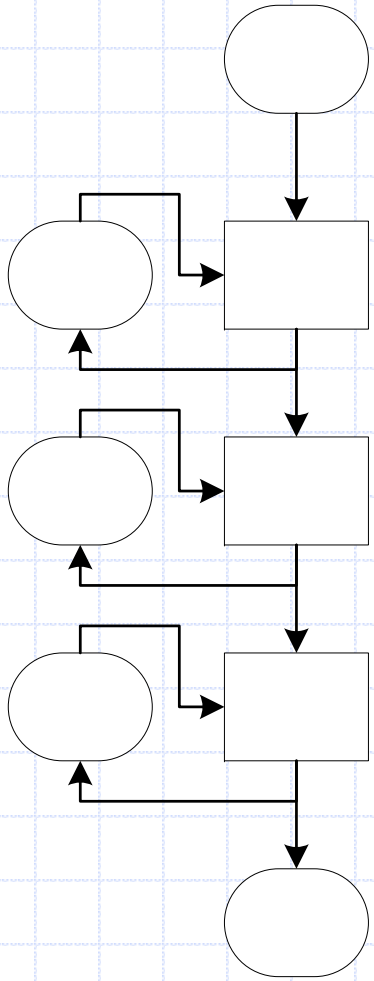
Pakovanje



- Pakovanje se može napraviti kombinacijom
  - šablona Skeniranje i Razbacivanje

- Eliminisanje nekorišćenog prostora u zbirci podataka
  - Preskače elemente označene vrednošću 0
  - Preostali elementi idu u kontinualan prostor
- Posebno koristan u kombinaciji sa drugim:
  - Npr. sa šab. Preslikavanje
  - Izbegava nepotreban izlaz
  - Sužava mem. Throughput
  - Emulira kontrolu toka na SIMD
  - Inverzna operacija je Raspakivanje
  - Obe su deterministične

Protočna obrada

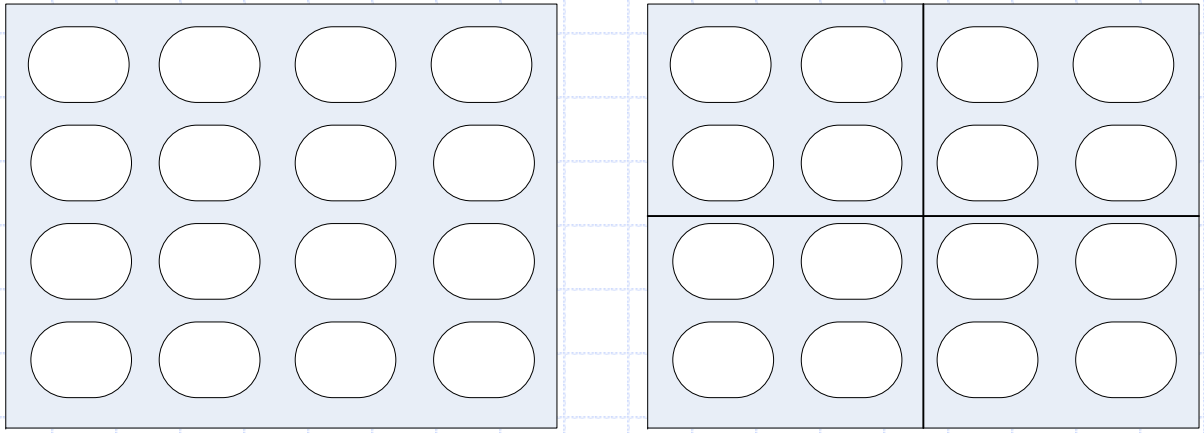


- Zadaci u relaciji proizvođač-potrošač
  - Svi stepeni istovremeno aktivni, i imaju stanje
  - Osnovni šablon je Linearna protočna obrada
  - U opštijem slučaju skup stepeni može biti povezan u USMERENI ACIKLIČNI GRAF (DAG)
- Korisne za serijski zavisne zadatke
  - Npr. kodovanje i dekodovanje videa i audia
- Prave se funkcionalnom dekompozicijom zadataka u aplikaciji
- Mana: relativno mala skalabilnost
  - Ipak, korisne kada se komponuju sa drugim šablonima

Geometrijska dekompozicija (1/2)

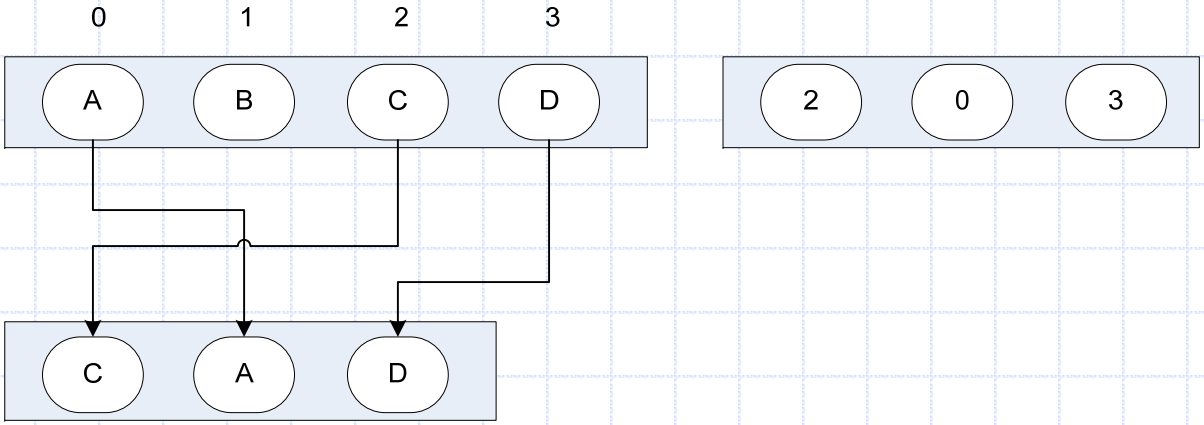
- Razbija zbirku podataka na skup zbirki
- Šablon PARTICIONISANJE:
  - Slučaj podele na zone bez preklapanja
  - Paralelni zadaci mogu da rade nezavisno
  - Algoritmi podeli-i-zavladaj i Obrada suseda
- Obrada suseda: preklapanje traka na ulazu, bez preklapanja na izlazu
- Problemi: granični uslovi, pločice iste veličine
- Umesto pomeranja podataka, alternativni pogled na organizaciju podataka

Geometrijska dekompozicija (2/2)



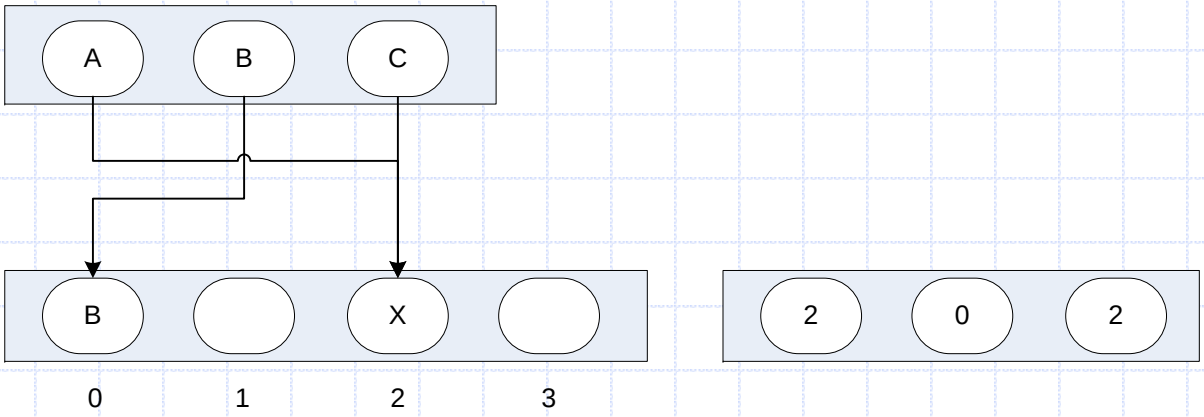
- Podela u regularne podnizove
  - Ili podzbirke različitih veličina, podzbirke koje su učešljane
  - Podela grafa: povezani čvorovi ili na druge načine
  - Distribucija: komunikacija samo preklopljenih domena
- Primene: JPEG i druge makroblok kompresije
  - množenje matrica tipa podeli-i-zavladaj, itd.

Skupljanje



- Izlazna zbirka podataka na osnovu
  - ulazne zbirke podataka sa indeksima
  - Kombinacija operacija Preslikavanja i Slučajno čitanje
- Optimizacije: niz indeksa fiksan, ili neki šablon
  - Npr. pomeranje podataka ulevo ili udesno putem vektorskih operacija, optimizacija Obrade suseda, itd.
- Primene: retke matrice, rač. grafika, bliskost/kolizija

Razbacivanje



- Inverzan od šablona Skupljanje
  - Problem: Šta ako dva upisa idu u istu lokaciju?
  - Dakle, moguća je trka do podataka – KOLIZIJA
- Potpune def. Razbacivanja – neko rešenje kolizije:
  - Korišćenje asocijativnih operatora za kombinovanje elemenata
  - Nedeterminističko biranje jednog od više elemenata
  - Pridruživanje prioriteta pojedinim elementima

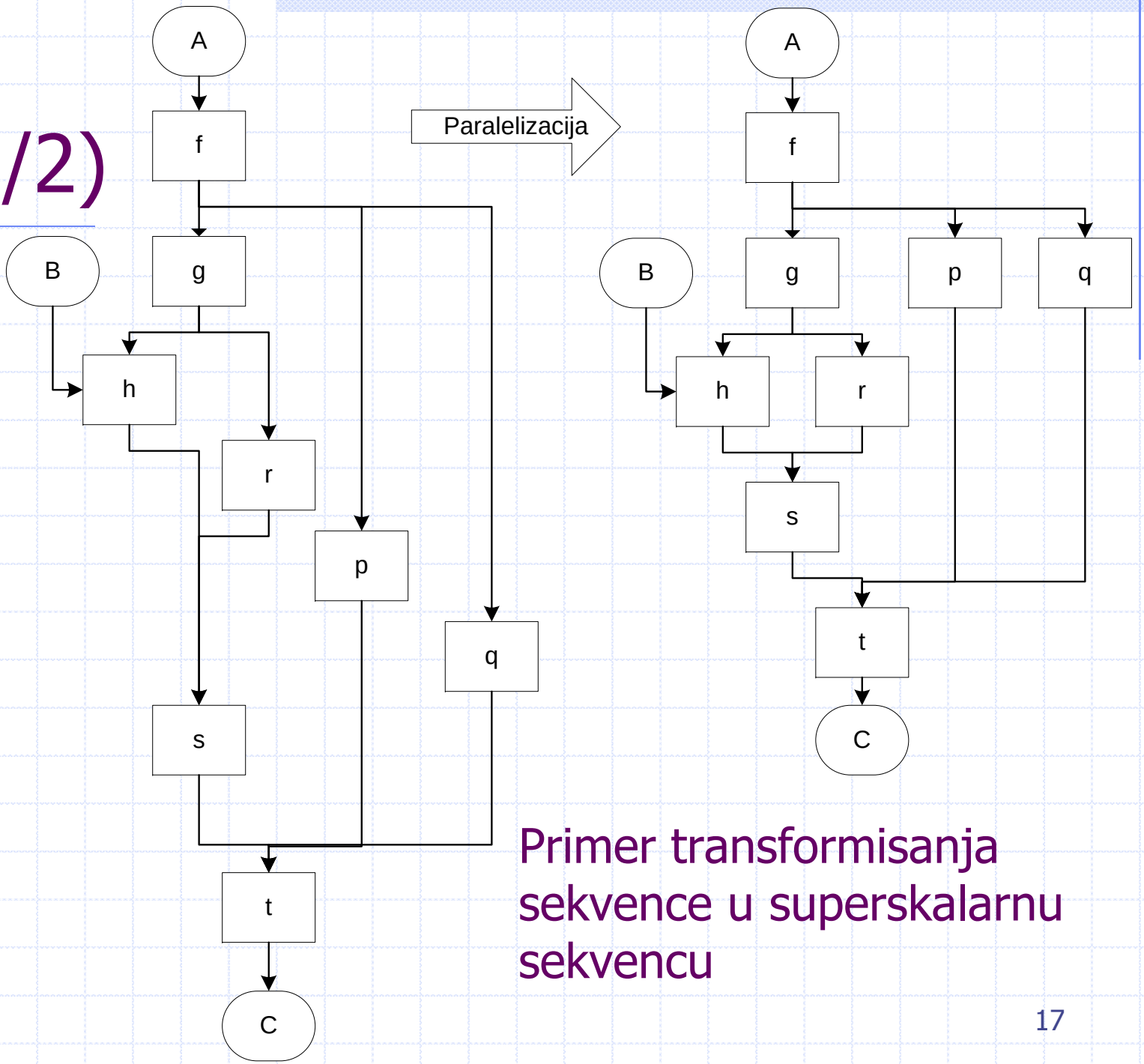
Drugi paralelni šabloni

- Drugi paralelni šabloni:
  - Superskalarna sekvenca
  - Buduće vrednosti
  - Spekulativni izbor
  - Gomilanje posla
  - Pretraga
  - Segmentacija
  - Proširivanje
  - Redukcija kategorija
  - Prepisivanje delova grafa

Superskalarna sekvenca (Ss) (1/2)

- Za razliku od serijske sekvence
  - Redosled zadataka određen zavisnostima podataka
  - Ako nema ivičnih efekata: zadaci teku paralelno
  - ili u redosledu koji je različit od onoga u izvornom kodu programa
  - Zavisnosti podataka moraju biti vidljive raspoređivaču
- Ovaj šablon ima veze sa šab. Buduće vrednosti:
  - Ss nema eksplicitnog rukovanja ili čekanja zadat.
  - Ss samo mora biti serijski konzistentna
- Sledi primer

Ss (2/2)

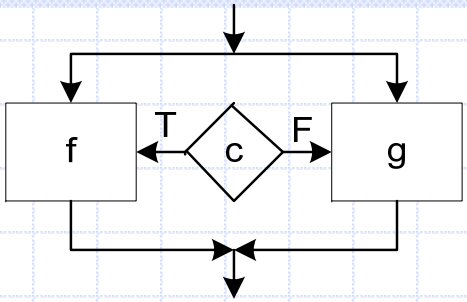


Primer transformisanja sekvence u superskalarnu sekvencu

Buduće vrednosti

- Ovaj šablon je kao Grananje-Pridruživanje (G-P)
  - ali zadaci ne moraju biti hijerarhijski ugnježdeni
  - Mrešćenje vraća objekt – BUDUĆA VREDNOST (BV)
  - Operacija nad BV: čekanje na završetak zadatka
- Za implementaciju opštijih grafova zadataka
  - G-P i BV kao stek i heap u memoriji
- Operacija otkazivanja = uništavanje zadatka:
  - Može se iskoristiti za implementaciju drugih šablona
  - Nedetrministički šab. Grananje i ograničavanje ili šab. Spekulativni izbor

Spekulativni izbor (1/2)



- Generalizuje sekvencijalni izbor, tako da
  - obe alternative mogu da se izračunavaju paralelno
  - Nakon određivanja uslova, suvišna grana se otkazuje
  - Vraćanje bilo kakvih ivičnih efekata
- Ovaj šablon je inherentno rasipnički
  - Uvek povećava ukupnu količinu posla
- Otkazivanje može biti skupo
  - Pogotovo ako je potrebno zakasniti ivične efekte
- Model prog. mora podržavati otkazivanje zadatka, npr. TBB model podržava

Spekulativni izbor (2/2)

- Šablon paralelizma najfinije skale, u dva slučaja:
  - I: Radi skrivanja kašnjenja na nivou instrukcija
  - II: Radi simuliranja više niti na SIMD jedinicama
- Sl. I: spekulativni izbor ili van redosleda
  - Spekulativni izbor ne mora biti rasipnički
  - Realizuje kompajler ili procesor
- Sl. II: niti se emuliraju korišćenjem maskiranja
  - Upis u mem.samo na SIMD trakama gde je dozvoljen
  - Sličan pristup za emuliranje iteracije na SIMD jedinicama - završetak petlje: maska sve 1-ce/0-le

Gomilanje posla

- Generalizacija šab. Preslikavanje
  - Svaka instanca elementne f-je može generisati više stavki, i dodati ih na gomilu posla
  - Npr. u rekurzivnoj pretrazi stabla, po jedna instanca za obradu svakog potomka
- Za razliku od šablona Preslikavanje
  - Ukupan broj instanci osnovne f-ije nije poznat
  - Niti je regularna struktura posla (nije po mustri)
  - Teže se vektorizuje od šablona Preslikavanje

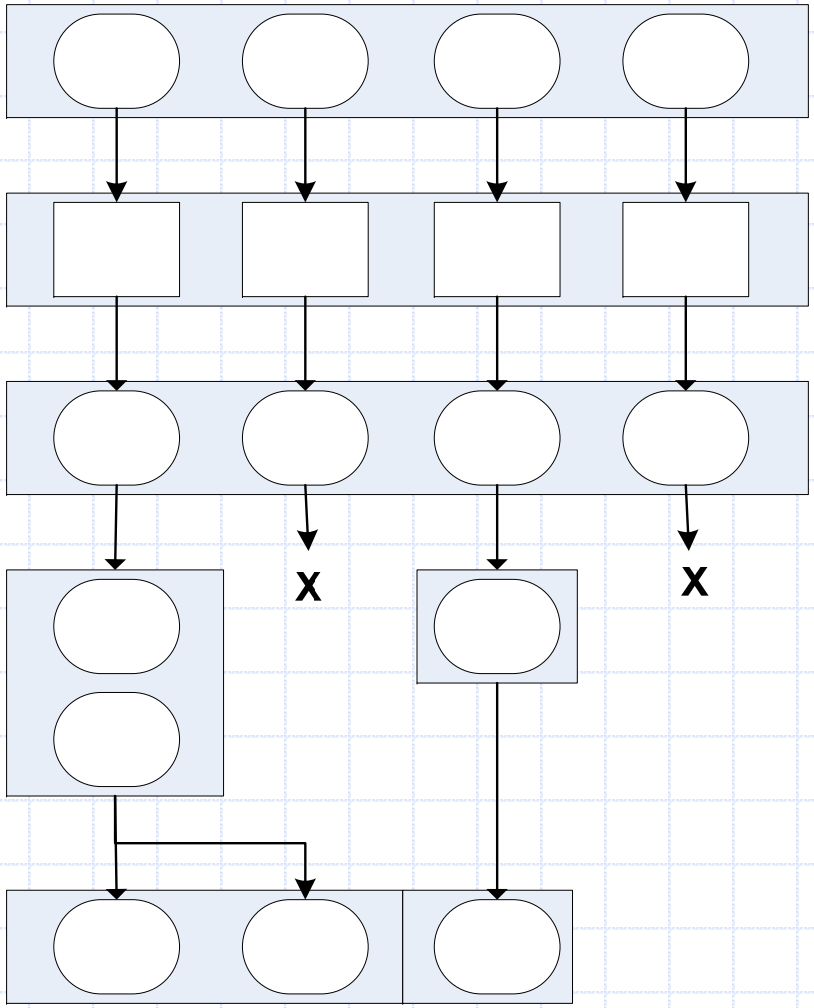
Pretraga

- Pronalazi podatak unutar zadate zbirke podataka
  - koji zadovoljava neki kriterijum
  - Npr. poklapanje sa zadatim ključem
  - Ili skup logičkih i aritmetičkih ograničenja
- Pretraga je obično povezana sa sortiranjem
- SQL se može posmatrati kao model PP
  - LINQ firme Microsoft koristi generalizovane pretrage kao osnovu za svoj model programiranja

Segmentacija

- Generalizovane operacije na zbirkama
  - SEGENTIRANE ZBIRKE su 1D nizovi podeljeni u nepreklapajuće i neuniformne particije
  - Skeniranje i Redukcija nad svakim segmentom
  - Preslikavanje nad segmentom ili elementom
- Segmentirane radnje nad zbirkama su skuplje
  - ali se lako vektorizuju i uravnotežuje opterećenje
- Primene:
  - Qicksort: PRVO-U-ŠIRINU se može vektorizovati
  - Analize vremenskih serija podataka (finansije, itd.)

Proširivanje

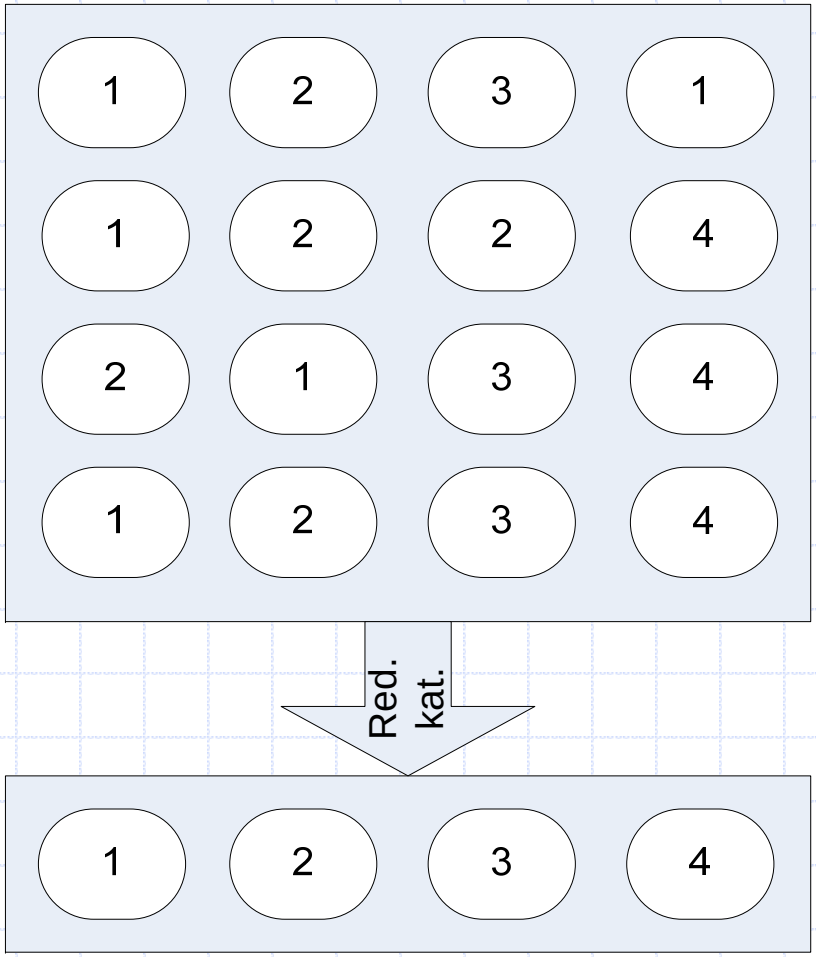


- Spoj šab. Preslikavanje i šab. Pakovanje
  - Svaki element preslikavanja može proizvesti nula ili više elemenata na izlazu
  - Pakovanje po poziciji preslikavanja i redosledu proizvođenja
- Primene:
  - Testiranje parova radi detekcije kolizije
  - kompresije audia/videa sa promenljivom brzinom

Redukcija kategorija (1/2)

- Za zadatu zbirku pod. sa pridruženim labelama
  - Pronalazi sve elemente sa istom labelom i
  - redukuje ih na jedan el. korišćenjem asocijativnog (i možda komutativnog) operatora
  - Kombinacija šab. Pretraga i segmentirana Redukcija
- Problem: paralelizacija pretrage i poklapanja
- Google rešenje: Hadoop MapReduce:
  - Preslikavanje generiše izlazne podatke i skup labela
  - Redukcija kategorija kombinuje i organizuje izlaz iz preslikavanja (Google to zove kratko redukcija)

Redukcija kategorija (2/2)



- Redukcija kategorija:
  - Pronađu se svi el. sa istom labelom
  - Redukuju se zadatim operatorm
- Primene:
  - Matrični proračuni na delovima slike
  - Hiljade aplikacija koje su implemenirane u modelu MapReduce

Prepisivanje delova grafa

- Pronalazi sve instance podgrafa TERM
  - i zamenjuje ih instancama novog podgrafa
  - Zamene se ponavljaju iterativno
    - sve dok više nema mogućnosti zamene
- Šab. ekvivalentan Lambda računu
  - za definisanje semantike funkcionalnih jezika
- Paralelno u različitim delovima grafa:
  - Ako je operacija konfluentna (ne zavisi od redosleda)
- Primene:
  - funkcionalni jezik Concurrent Clean, sinteza FPGA, itd.

Nedetirminstički šabloni

- Nedeterminizam otežava testiranje programa
  - Ali može biti koristan
- Nedetirminstički šabloni:
  - Grananje i ograničavanje
  - Transakcije
- Apstrakcija može biti deterministička
  - a njena implementacija interno nedeterministička
  - Treba razumeti kada nedeterminizam može biti sadržan unutar neke apstrakcije

Grananje i ograničavanje

- Često se koristi u implementaciji pretrage
- Paralelna pretraga:
  - Skup stavki se podeli i podskupovi pretraže paralelno
  - Čim se pronađe stavka, ostale pretrage se otkazuju
- Super-linearna ubrzanja
  - Otkazivanje zadataka mora biti efikasno
- Primene: Matematička optimizacija:
  - Algoritam nedetirministički, rezultat deterministički
  - Slične tehnike za pretraživanje prostora-stanja u veštačkoj inteligenciji (npr. Alfa-beta potkresivanje)

Transakcije (1/2)

- Kada centralnu bazu treba višestruko ažurirati
  - Pri čemu redosled nije važan
  - ali se baza mora održavati u konzistentnom stanju
- Primer Banka:
  - Koristi asocijativne operacije (+ i -), pa je rezultat faktički deterministični
- Primer 2: tabele sa dir. pristupom (hash table)
  - Skup kontejnera (bucket), npr. skup lista
  - Pomoću transakcija više zadataka može paralelno i konzistentno da umeće nove elemente

Transakcije (2/2)

- Redosled ne mora biti isti u svakom izvršenju
- Ali, program može biti determ. ako unutrašnji nedeterminizam nije izložen izvan implementacije
- Tada će pretrage tabele uvek vraćati iste rezultate

- Problem: dodati dva elementa sa istim ključima
  - Ako se zadržava zadnji: nedeterministčka tabela
  - Ako se uzima npr. veći: deterministčka tabela
- Implementacija transakcija:
  - Može pomoću brava, ali nije skalabilno, bolje pomoću
  - Protokola ZAVRŠI I PONOVI (commit and rollback)
    - HW/SW Transakcione Memorije (TM), npr. Intel Haswell