دانشگاه صنعتی امیرکبیر
دانشکده مهندسی کامپیوتر و فناوری اطلاعات

معماری سوئیچ و روترهای با کارآیی بالا

تمرین اول

پرهام الوانی

۲۴ آبان ۱۳۹۶

۱ سوال اول

در این سوال فرض می‌کنیم عناصر یکسان هستند و تفاوتی با یکدیگر ندارند.

هر عنصر می‌تواند در هر یک از ۱۰۰۰۰ بلوک موجود قرار بگیرد. فرض می‌کنیم یکی از ۱۰۰۰۰ بلوک خالی باشد، به این ترتیب هر عنصر ۹۹۹۹ انتخاب خواهد داشت. باید به این تکته توجه داشت که برای جلوگیری از شمارش جندباره‌ی حالت‌ها نیاز است که مطمئن شویم سایر خانه‌های حداکثر یک عنصر خواهند داشت. بنابراین احتمال خالی بودن یکی از بلوک‌ها به شرح زیر است:

$$\frac{\binom{10^4}{1} * \binom{5000-1}{10^4-2}}{\binom{5000+10^4-1}{10^4-1}} \quad (۱.۱)$$

همانطور که مشخص حاصل رابطه‌ی بالا برابر صفر است زیرا امکان ندارد بتوان با ۵۰۰۰ عنصر ۱۰۰۰۰ خانه را طوری پر کرد که دقیقا یک خانه خالی باشد.

اگر فرض کنیم متغیر تصادفی x نشان‌دهنده‌ی تعداد بلوک‌های خالی باشد، امید ریاضی x متوسط تعداد بلوک‌های خالی خواهد بود به این ترتیب داریم:

$$E[x] = \sum_{n=0}^{10^4} n \frac{\binom{10^4}{n} * \binom{5000-1}{10^4-n-1}}{\binom{5000+10^4-1}{10^4-1}} \quad (۲.۱)$$

ابتدا دو عنصر را انتخاب کرده و در یک بلوک مشخص قرار می‌دهیم و سایر عناصر ۹۹۹۹ انتخاب خواهند داشت. اگر فرض کنیم که بلوک دو عنصری از پیش مشخص شده است خواهیم داشت:

$$\frac{\binom{5000-1+10^4-1}{10^4-1}}{\binom{5000+10^4-1}{10^4-1}} \quad (۳.۱)$$

اگر فرض کنیم متغیر تصادفی y نشان‌دهنده‌ی تعداد بلوک‌های دو عنصری باشد متوسط تعداد بلوک‌های دو عنصری امید ریاضی y خواهد بود. نکته‌ی اصلی در این روش محاسبه‌ی تعداد بلوک‌های دو عنصری بدون چندباره شماری است، برای این منظور می‌بایست مطمئن شویم که سایر بلوک‌ها دو عنصری نخواهند بود، یعنی صفر، یک یا بیش از دو عنصر دارند.

۲ سوال دوم

برای جستجو در اولین گام بلوک مورد نظر مشخص می‌شود و سپس جستجو در همان بلوک صورت می‌پذیرد. از آنجایی که جستجو در بلوک صورت می‌پذیرد پس زمان آن به

اندازه‌ی بلوک وابسته است، اندازه‌ی هر بلوک در صورتی که تعداد عناصر آن‌ها با یکدیگر برابر باشد n/m می‌باشد.

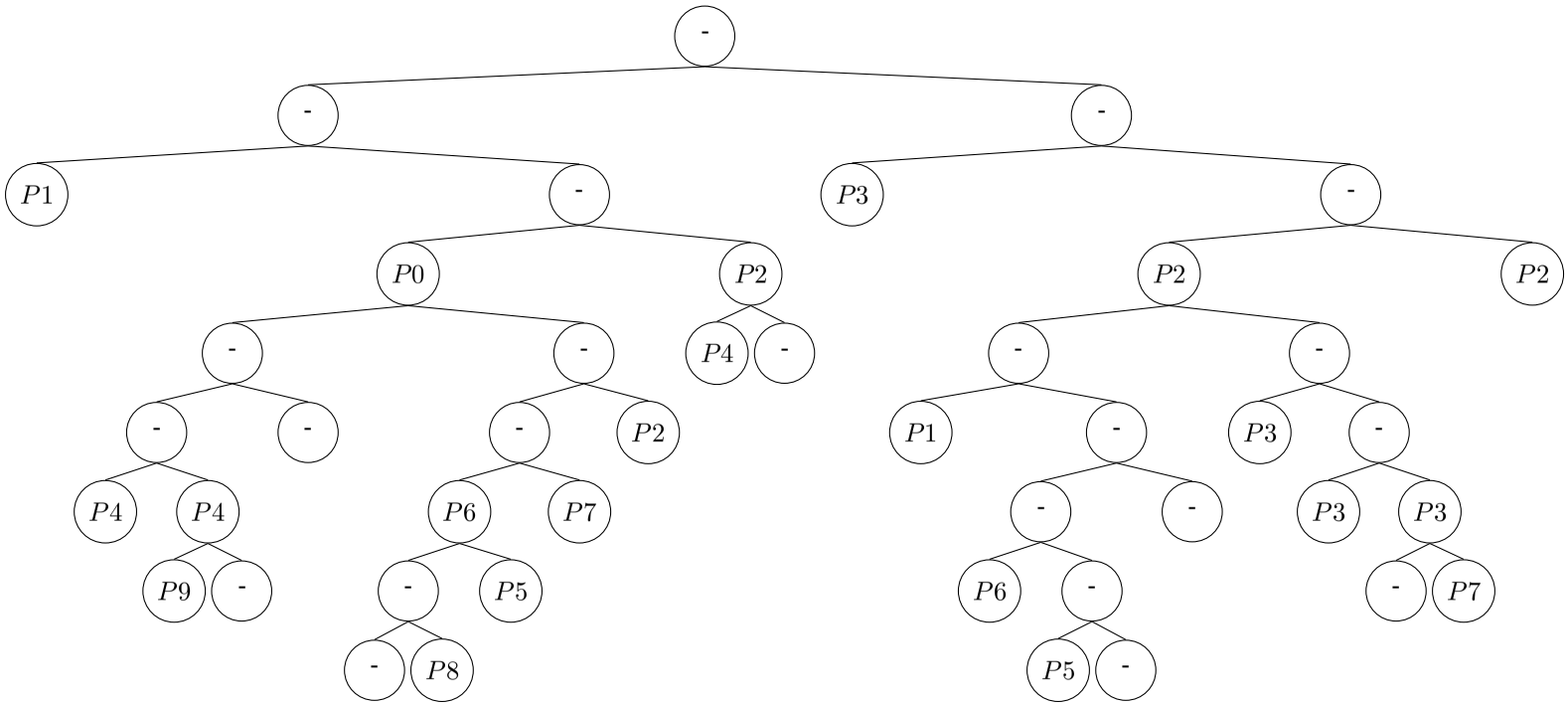
$$search = O(n/m)$$

حافظه‌ی مصرفی این روش اگر تعداد عناصر همه‌ی بلوک‌ها با یکدیگر برابر باشد به اندازه‌ی عناصر موجود خواهد بود ولی در صورتی که این امر اتفاق نیافتد حافظه‌ی مصرفی این روش از تعداد عناصر موجود بیشتر خواهد بود.

$$memroy = \Omega(n/m)$$

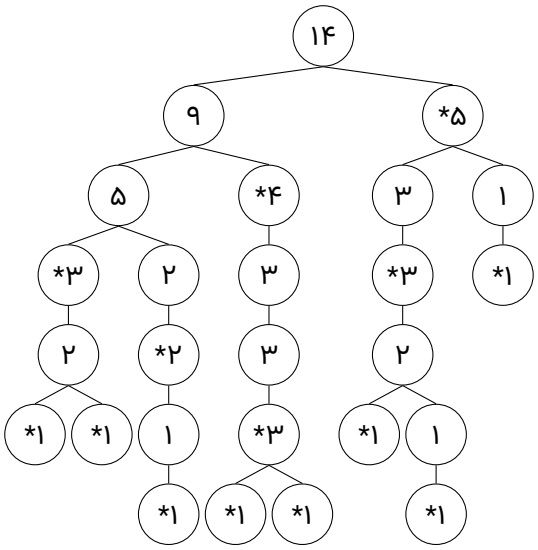
برای به روزرسانی تنها نیاز است که بلوک موردنظر پیدا شود و در ادامه محتوای آن به روزرسانی شود بنابراین زمان اجرای به روزرسانی با جستجو تفاوتی ندارد.

۳ سوال سوم


-
-
-
P1
-
P3
-
P0
P2
P2
P2
-
-
P4
-
-
-
-
-
P1
-
P3
-
P4
P4
-
P2
P6
P7
-
-
P3
P3
P9
-
-
P5
P6
-
-
P7
-
P8
P5
-

۴ سوال چهارم

در ابتدا درخت دودویی که با تعداد prefixها شماره‌گذاری شده است، رسم می‌کنیم.



روش subtree-spliting

Index	Bucket Prefixes	Bucket Size	Covering Prefix
000^*	$000^*, 00010^*, 00011^*$	3	000^*
00^*	$0011^*, 001110^*$	2	—
$0*$	$01^*, 01101^*, 011010^0, 011011^*$	4	—
10^*	$100^*, 10010^*, 100110^*$	3	1^*
$*$	$1^*, 111^*$	2	—

ورش post-order spliting

Index	Bucket Prefixes	Bucket Size	Covering Prefix
$000^*, 00111^*$	$000^*, 00010^*, 00011^*, 001110^*$	4	$000^*, 0011^*$
$00^*, 011^*$	$0011^*, 01101^*, 011010^*, 011011^*$	4	01^*
$0^*, 10^*$	$01^*, 100^*, 10010^*, 100110^*$	4	1^*
$*$	$1^*, 111^*$	2	—

۵ سوال پنجم

۱.۵

امروزه با افزایش داده‌ها در دیتاسنترها نیاز به توان پردازشی بیشتر و ارتباطات داخلی سریعتری است. استفاده از تکنولوژی‌های الکتریکی برای انتقال داده محدودیت‌های زیادی دارد، مانند دخالت درونی[۱]. یک پاسخ طبیعی به این مشکل استفاده از تکنولوژی‌های نوری است، این تکنولوژی‌ها توانایی سوئیچینگ و نرخ ارسال بالایی دارند.

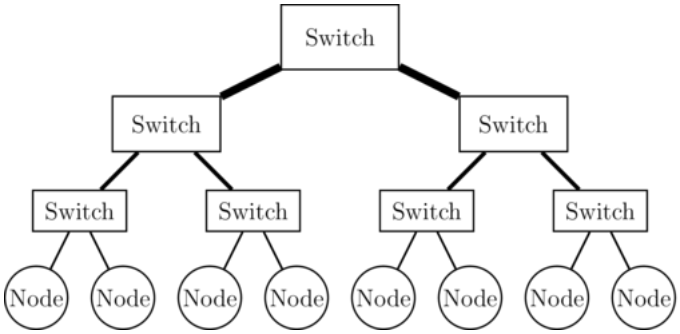
استفاده از تکنولوژی‌های نوری مصرف انرژی و فضای بیشتری نسبت به تکنولوژی‌های الکتریکی دارد. مطالعات زیادی برای ریلکس کردن این موارد صورت گرفته است.

این مقاله در نهایت به دنبال یافتن پاسخ برای افزایش نرخ داده‌ها در دیتاسنترها به وسیله‌ی شبکه‌های نوری و حل مشکلات این نوع شبکه‌ها در دیتاسنترها است.

۲.۵

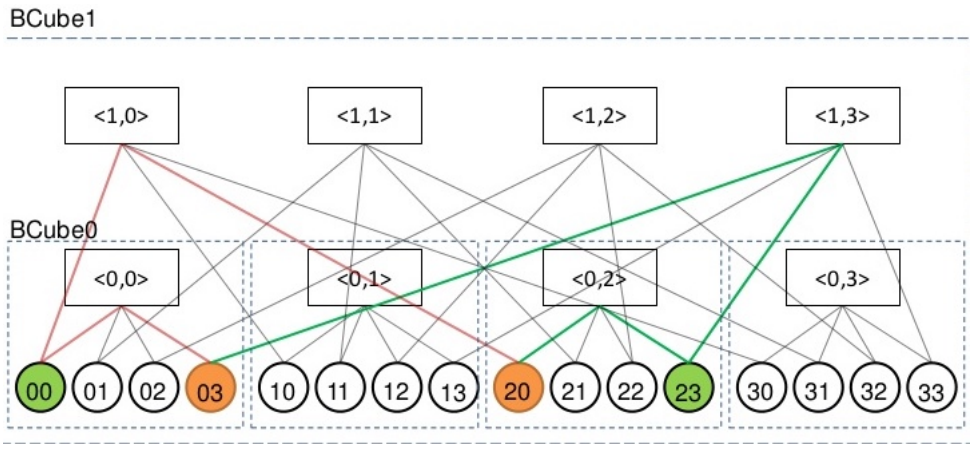
در معماری switch-centric تنها ارتباطات سوئیچ-سوئیچ و سرور-سوئیچ وجود دارد و ارتباطات سرور-سرور وجود ندارد مانند fat-tree.

[۱] interference inter-symbol



این روش سرعت پردازش زیادی داشته ولی مصرف توان و هزینه‌ی زیادی دارد. این روش برنامه‌پذیری کمی دارد.

در معماری server-centric تنها ارتباطات سرور-سرور و سرور-سوئیچ وجود دارد و ارتباطات سوئیچ-سوئیچ وجود ندارد مانند BCube.



در این معماری تاخیر پردازشی بیشتر از روش switch-centric است ولی قابلیت برنامه‌پذیری بیشتری نسبت به آن روش وجود دارد.

۳.۵

۱. بالاترین سطح ترافیک نماینده‌ی ترافیک بین رک‌ها است، طول لینک در این ارتباطات بین چند متر تا چند صد متر می‌باشد.

۲. ارتباطات درون تجهیز رک می‌توانند طول لینکی بین ۱۵ سانتی‌متر تا چند متر داشته باشند.

۳. ارتباطات بین چیپ‌ها در یک ماژول که طول لینکشان کمتر از ۲۵ سانتی‌متر می‌باشد.

۴. ارتباطات روی چیپ‌ها که طولشان زیر ۲ سانتی‌متر است.

ارتباطات در ابعاد مختلف مس‌توانند از معماری‌ها و تکنولوژی‌های مختلفی استفاده کنند زیرا پارامترها و اهداف طراحی آن‌ها با یکدیگر متفاوت است.

۴.۵

مشکل مهمی که مقیاس‌پذیری و مدیریت سیستم‌های مقیاس-بزرگ را محدود می‌کند تعداد زیاد لینک‌های لازم برای ارتباطات داخلی است که باعث تعداد زیاد کابل‌ها می‌گردد. این مشکل عموما به عنوان wiriing problem شناخته می‌شود.

روش 2D Torus از نظر تعداد لینک‌ها با توجه به مقایسه‌ای که در مقاله صورت گرفته است مقیاس‌پذیری بیشتری دارد.

۵.۵

معماری سوئیچینگ banyan یک معماری چند مرحله‌ای است که دارای blocking است.

معماری سوئیچینگ cantor یک معماری چند مرحله‌ای است که دارای blocking نیست. در این معماری مستقل از اینکه ارتباطات چگونه ساخته می‌شوند شبکه‌ی حاصل جهت سوئیچینگ دارای blocking نخواهد بود.