مجله مدیریت تکنولوژی اطلاعات

نشریه انجمن مدیریت

ندا علی آبادی

آیا databases های توزیع یافته ابزار مناسبی برای افزایش سرعت دستیابی به اینترنت فراهم می کند؟

خلاصه:

به منظور پشتیبانی از عملکرد برنامه های کاربردی تحت وب، بهینه سازی داده های ذخیره شده، که می توانند خلاصه شده و به سوی یک کاربر ارسال شوند، نقش مهمی را ایفا می کند.

تکنولوژی مربوط به هارد دیسک ها که مبتنی بر تکنولوژی مکانیکی است، آهسته ترین بخش در بازیابی اطلاعات می باشد. مانع مکانیکی million-to-one سرمایه گذاری برای بهینه سازی را قابل قبول می کند که این بهینه سازی از طریق ذخیره سازی داده ها در دیسک های چندگانه که در وسایل چندگانه ای توزیع شده است، انجام می گیرد. این متدولوژی منجر به کاهش زمان دسترسی به داده ها می گردد.

رشد مستمر در خدمات اینترنتی و million of hits scenario، تحقیقات به منظور تعیین مزایای عملکرد دیتا بیس های توزیع یافته و مدل ساختاری آن را به امری ضروری تبدیل کرده است.

این مقاله مجموعه ای از تجربیات در اتباط با ۳ نوع متفاوت از الگوریتم ها را به منظور تعیین مزایای بالقوه دیتا بیس های توزیع یافته به کار می برد و نتیجه نشان می دهد که الگوریتم load balansed که شامل ۴ گره یا دیتا بیس می باشد به طور قابل ملاحظه ای عملکرد را بهبود می بخشد.

کلمات کلیدی: دیتا بیس توزیع یافته، تعادل بار کاری، عملکرد IO، الگوریتم های موازی

معرفی: قبل از اینکه برنامه های کاربردی به برنامه های تحت وب تبدیل شود، حداکثر تعداد کاربران بالقوه به سایز شبکه های خصوصی که کاربران به آن وصل می شدند بستگی داشت که اغلب این حداکثر مقدار برابر بود با چند هزار کاربر. با روی کار آمدن برنامه های کاربردی تحت وب انتظار حداکثر سایز چند میلیون کاربر غیر واقعی نیست.

با ظهور صنعت دیسک درایو ها، عملکرد بهینه و مناسب در برنامه های کاربردی متمرکز تحت وب به علت million of this scenario قابل حصول نیست.

و همانگونه که elnikety ذکر می کند به کارگیری مدل دیتا بیس توزیع یافته منجر به افزایش عملکرد ۱۰ درصدی و کاهش زمان پاسخگویی ۱۴ درصدی می گردد. سه متغیر به منظور بهینه سازی برنامه های کاربردی تحت وب در نظر گرفته می شوند.

اگر چه مفهوم دیتا بیس توزیع یافته بیش از بیست سال است که روی کار آمده است، اما هنوز در برنامه های کاربردی مربوط به محیط کسب و کار رواج پیدا نکرده است. پیچیدگی و هزینه ناشی از اضافه کردن گره های اضافی مانع از استفاده و توسعه دیتا بیس توسعه یافته شده است. anthes بیان می دارد که سیستم های دیتا بیس توزیع یافته به سادگی به حوزه برنامه های علمی، مهندسی، ریاضی و آماری گسترش یافته است. همزمان با "millions of hits scenario" و همراه با رشد خدمات اینترنتی، تحقیقاتی به منظور تعیین مزایای دیتا بیس توزیع یافته و مدل پایه ای ساختار و ترکیب آن مورد نیاز است. smith نیز در تأیید این گفته به طور ویژه به بیان نیاز به تحقیقات ارزیابی عملکرد بیشتر در ارتباط با دیتا بیس های بزرگتر پرداخته است.

مزایا:

Peddemors بیان کرده است که مزایای زیادی در استفاده از ساختار و معماری دیتا بیس توزیع یافته نهفته است به ویژه زمانی که بار کاری بسیار زیاد است و اضافه می کند که این ساختار به ویژه برای برنامه های کاربردی تحت وب مناسب است. sobol بیان میدارد که افزایش در برنامه های کاربردی مبتنی بر تکنولوژی client-server و telecommunication منجر به تحریک و به جلو راندن سیستم های پردازش توزیع یافته و پراکنده می گردد و در نتیجه منجر به افزایش نیاز به دستیابی کارا به دیتا بیس های سازمانی خواهد شد. این افزایش نیاز به دیتا بیس ها موضوعاتی مانند فضای ذخیره سازی کار و زمان دستیابی را به موضوعات مهمی کرده است. بنابراین معماری جدیدی برای دیتا بیس ها مانند دیتا بیس های توزیع یافته مورد نیاز خواهد بود. ساخت دیتا بیس های توزیع یافته با ساختار client-server در اغلب اوقات موفق بوده است. به طور مثال roussopoulos یک سیستم مدیریت داده پیشرفته را در دانشگاه ماریلند در سال ۱۹۹۳ توسعه داده است. هرچند این واضح است که فزونی برنامه های کاربردی تحت وب و "million of hits scennario" عامل اصلی ظهور و به کارگیری دیتا بیس های توزیع یافته می باشد.

ملاحظات طراحی:

Amiri بیان می دارد که انتخاب معماری دیتا بیس های توزیع یافته مزایای نهفته زیادی را برای خرده فروشان multimedia به همراه دارد. هر چند طراحی سیستم های دیتا بیس های توزیع یافته باید کاملا سنجیده و حساب شده باشد. مشکلی که در این مرحله ظاهر می گردد برنامه ریزی و طراحی و توسعه سیستم های دیتا بیس توسعه یافته از طریق معرفی سرور های جدید و احتمالا از رده خارج کردن برخی از سرورهای موجود است. هدف کاهش هزینه های ارتباط از را دور برای پردازش تقاضای کاربران و دستیابی سرورها به آنها به علاوه بهره برداری و نگهداری در محیط هایی با دوره های زمانی متفاوت که در آن زمان پردازش تقاضای مشتریان متفاوت است. li et al هم بر

اهمیت طراحی مناسب دیتا بیس ها تأکید کرده است.آنها بیان می کنند که علی رغم وجود content delivery network(CDN) بسیاری از برنامه های کاربردی تحت وب که از دیتا بیس ها استفاده می کنند به مراکز داده ای متکی هستند که به منظور عملکرد و اعتبار بیشتر از برنامه های کاربردی و محتوای دیتا بیس ها میزبانی می کنند. هر چند این موضوع باعث مطرح شدن موضوعاتی از قبیل همزمانی مرکز داده/ دیتا بیس، مسیریابی نقل و انتقالات/صف، متعادل کردن بار کاری و دقت و صحت نتایج برنامه های کاربردی می گردد. بنابراین آنهااحساس کردند اگر برنامه های کاربردی تحت وب در ساختار مراکز داده توزیع یافته به صورت موفق اجرا گردد این موضوعات باید مشخص شود. Simba et al دو موضوع اصلی در طراحی دیتا بیس های توزیع یافته را توضیح داده است. اولین مشکل یا مسئله بر می گردد به مشخص کردن تعداد مکان ها یا مراکز متفاوتی که در یک دیتا بیس توزیع یافته می توان به آنها دسترسی پیدا کرد و مسئله دیگر بر می گردد به تعداد موانع دستیابی در یک رابطه. اولین مسئله مستقیملا به موضوع مورد مطالعه ما مربوط است به علت آتکه به تعداد گره ها و الگوی دستیابی مربوط می گردد. دومین مسئله به این بر می گردد که داده ها چگونه در یک گره معین به صورت جزئی تر توزیع می گردند. بازبینی نوشتجات مفاهیمی را در رابطه با حفظ اعتبار آشکار می سازد که پیچیدگی اضافی و زایدی را به دیتا بیس های توزیع یافته می بخشد. Xiong etal این موضوع را بیان کرده است. تکرار داده ها می تواند به سیستم های دیتا بیس کمک کند تا محدودیت های زمانی شدید برنامه های کاربردی real-time موجود را برآورده سازد به ویژه دفترچه های راهنمای تحت وب و خدمات تجاری بازرگانی. هرچند پیش نیاز مورد نیاز برای دستیابی به مزایای تکرار، توسعه یک مکانیزم کنترل همزمانی با عملکرد بالاست. این به معنای این است که کلیه گره هایی که از اطلاعات استفاده می کنند باید همزمان باشند و به روز گردند. Wu et al با اهمیت اعتبار و اطمینان موافق است بنابراین به ایجاد پروتکلی که مسائل را نشان دهد علاقه نشان می دهد. این مقاله یک ساختار و طرح نو را برای اجرا یک پروتکل کنترل المثنی انعطاف پذیر در سیستم های دیتا بیس توزیع یافته ارائه می دهد. این ساختار طرح به تعداد گره های کمتری نیاز دارد تا بسته شود و عملیات نوشتن/خواندن را انجام دهد. این موضوع نه تنها عملکرد بهتری را فراهم می کند بلکه به طراحان سیستم انعطاف پذیری بیشتری را به هنگام اجرای پروتکل ارائه می دهد.

مسایل مربوط به عملکرد:

(cannataro)از طرفداران پردازش توزیع یافته پراکنده است،آنها بیان می کنند که یکپارچگی محیط های محاسباتی پراکنده در آینده بهبود عمده ای را عملکرد برنا مه های کاربردی محاسباتی data intensive ایجاد خواهد کرد.در حقیقت مقاله اولیه آنها نگاه اولیه آنها به موضوعات اصلی مربوط به محاسبات داداه های متمرکز به صورت موازی در برنامه های کاربردی علمی و تجاری را فراهم خواهد کرد. این مقاله هم چنین خواننده تشویق می کند تا به دنبال

مقالات عمیق تر و تخصصی تری باشد که بعد ها در ژورنـال هـای موضـوعی خـاص چـاپ مـی گـردد jutla et al احساس می کندکه این برای کاربران نهایی بـسیار مهـم اسـت نـه بتوانـد عملکـرد بلقـوهDB هـای e-commera می باشد. آنها بیان می دارند که به علت جنبه های multidis ciplinary ،تجارت الکترونیک و ظهـور مـدل هـای متنوع و متمایز در عرصه e-commerce امکان پذیر نیست، گذشته از این نیازهای متنـوع شـرکت هـای متوسـط و کوچک و کسب و کار های بزرگ، لزوم داشتن یک الگوی مناسب برای e-commerce را شدیدتر می کند.

Rajamani بیان می کند که امروزه مسئله اصلی در فراهم کردن عملکرد مناسب در برنامه های کاربردی تحـت وب مسئله، ازدیاد درخواست های دیتا در یک زمان می باشد. به ویژه اکنون که وب سـایت هـا کـم کـم بـه سـمت ارائـه صفحات خاص هر مشتری به جای یک ثابت و عمومی پیش می روند کـه اطلاعـات خـاص هـر user و بـسیاری از خدمات آنلاین مورد نیاز آنها در آن ارائه می شود. وب سـایت هـای multi-taiered database-driven سـاختار غالبی را برای رویکردهای ساختار یافته و ثابت به منظور ارائه اطلاعات پویا، فراهم می سازد.

هر چند این رویکرد scalable (تقاضاهای مبتنی بر زمان و نیاز بالا به وب سایت هایی با محتوای پویا) در مقایسه با وب سایت هایی با محتوای ثابت منجر به تأخیر زمانی بالاتر و سطح عملکرد پایین تر می گردد و نیازمند این است که به خوبی سنجیده گردد و طرحی مناسب برای آن ارائه شود. Kanitkar بیان می کند که روش توزیع درخواست هـا در میان گره ها تأثیر عمده ای بر زمان پاسخگویی به آنها دارد. به منظور حل این مشکل توزیـع ،او طـرح جدیـدی را بـه منظور زمان بندی درخواست ها و نقل و انتقالات ارائه می دهد که در آن اولویت بالاتر را به نقل و انتقالاتی مـی دهد که اطلاعات مورد نیاز آنها در دسترس تر است و نیز به منظور بهبود عملکرد دیتا بیس هایپراکنده، یـک مکـانیزم تقسیم بار کاری را پیشنهاد می کند که در آن به هماهنگ کردن انتقال دیتـا و نقـل و انتقـالات مـی پـردازد کـه ایـن بالاترین احتمال برای اجرای موفق آنها را ارائه می دهد.

دامنه مطالعه:

به منظور اینکه این مقاله را ملموس تر و قابل فهم تر بنماییم در اینجا به توضیح چند پارامتر می پردازیم که بـه درک بهتر آزمایش انجام شده کمک می کند.

انتخاب سرور عملیاتی:

سرور عملیاتی انتخاب شده برای این پروژه انتخاب شده لینوکس است. به علت آزادی و درجه ی بالای انعطاف پذیری آن ، لینوکس به علت سر بار کمتر و استفاده از کد بهینه سطح عملکرد بالا تری را ارائه می دهد.

انتخاب نرم افزارDB:

DBمـورد انتخـاب mysql اسـت ایـن نـرم افـزار بـل سیـستم عملیـاتی hinuxسـازگار اسـت و از زبـان استانداردsqdاستفاده می کند.

ساختار DB:

Dbمحدود به یک جدول نیست.هدف از این بررسی دستیابی بهbaselineاز طریق ایجاد تغییرات در تعداد گـروه هـا، میزان بار کاری و الگوریتم توزیع است.

Workloud generator: siegeبه عنوان WGانتخاب شده است به علت اینکه طراحی آن بـه توسـعه دهنـدگان اجازه می دهد تا سطح عملکرد برنامه های خودشان را بر مبنای siege یا duress اندازه گیری کنند.

الگوریتم توزیع: اگر چه الگوریتم های زیادی موجود است اما این مقاله بر سه تـای آنهـا تمرکـز کـرده اسـت: ترتیبـی، تصادفی وloab balanced (LB)

مدل ترتیبی تقاضا ها را به ترتیب به گره ها تخصیص می دهد برخلاف مدل تصادفی که تقاضا را به صورت اتفاقی بـه گره ها تخصیص می دهد. مدل LB ، هر کدام از گره ها را چک می کند تا مطمئن شود که سطح بهره برداری از آنهـا از سطح آستانه و حد نهایی قابلیت فعالیت آنها پایین تر است.

اندازه خوشه و الگوی پیمایش: حداکثر تعداد گره های بهینه دیتا بیس ها به ۴ تا محدود می گردد. بر طبـق پیمـایش ، این معمول است که به منظور دستیابی به عملکرد از ۱،۲،۴،۸،۱۶ پردازنده استفاده گـردد. الگـوی پیمایـشی " دو برابـر سازی (doubling)" به طور گـسترده در بقیـه مطالعـات بـه کـار گرفتـه شـده اسـت و از نقطـه نظـر سـازگاری و transferability، با این مطالعه نیز سازگار است. این محدودیت به این منظور ارائه شده تا مطالعه را کنترل پـذیر تـر نماید و اجرا و ارزیابی نتایج را برای خوانندگان ساده تر نماید.

متدولوژی و نتایج:

این مقاله بوسیله آزمایش ترکیبات مختلفی از متغیر ها که در بالا لیست شده است، به بررسی اثربخشی دیتا بیس های پراکنده می پردازد. سوالات زیر مورد بررسی می باشد.

- چگونه شدت بار کاری بر نیاز و سطح عملکرد دیتا بیس های پراکنده تأثیر می گذارد؟ عامل اصلی collection در این محیط یک بسته sniffer است که توسط tcppump تولید می گردد. این عامل داده های حاصله از آزمایش های تجربی گوناگون را جمعآوری می کند.
- چگونه تعداد گره های دیتا بیس بر زمان دستیابی به داده ها تأثیر می گذارد؟

- چگونه الگوریتم به کار گرفته شده به منظور تخصیص درخواست های مشتریان به گره های خاص بر زمان دستیابی به داده ها اثر می گذارد؟

این سؤالات به گونه زیر اصلاح می گردد تا سه فرضیه صفر ایجاد کند که می تواند بوسیله آزمایشات تست گردد.

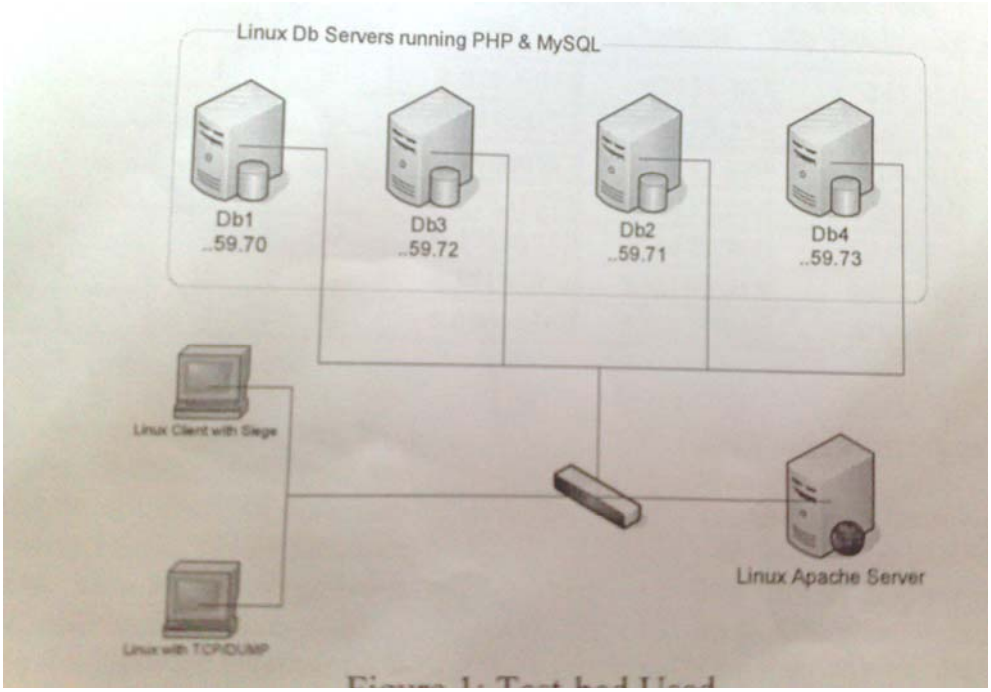
H_1 – شدت بار کاری اثری بر زمان بازیابی داده ها از دیتا بیس پراکنده ندارد از این رو تأثیری بر تأخیر و انتظار کاربران ندارد.

H_2 – تعداد گره های دیتا بیس اثری بر زمان پاسخگویی به کابران ندارد.

H_3 – الگوریتم مورد استفاده به منظور درخواست ها به گره های موج.د در دیتا بیس پراکنده اثری روی میزان و انتظار کاربران نهایی ندارد.

به منظور جمع آوری داده ها برای آزمون این فرضیه یک test bed data base مورد استفاده قرار گرفته است که در آن بار کاری برای هر تعداد از کاربران همزمان شبیه سازی می گردد. الگوریتم توزیع متفاوت است و تعداد گره های به کار رفته در دیتا بیس از ۱ تا ۴ متغیر است.

شکل زیر این test bed در زیر نمایش داده شده است.



URL مورد استفاده قرار گرفته است تا سه مدل ترتیبی و تصادفی و LB را امتحان کند. اطلاعات بیشتر در رابطه با این مدل ها در سایت زیر موجود است.

http://web.stcloudstate.edu/chrisb/thesis-20050501.doc

سرور apache سپس باید دوباره خروجی ها را بر مبنای الگوریتم که قبلا برای هر متد تعریف شده،سازماندهی کند. متغیرهای زیر در هر بسته ثبت شده، وجود دارند.

علامت زمان، مک آدرس مبدأ، مک آدرس مقصد، اندازه بسته، آدرس پورت شبکه مبدأ و آدرس پورت شبکه مقصد. این متغیر ها زمانی که پردازش می شوند پارامترهای زیر را فراهم می سازد: شدت ترافیک داده ها، سطح عملکردی، زمان انتظار کاربران.

پردازنده های مدل بالا که از لینوکس استفاده می کنند تولید بار کاری می نمایند. نرم افزاری که در آنها مورد استفاده قرار می گیرد siege است که قادر است تا جریان های ترافیک با شدت های متفاوتی را در وب ها تولید کند. برای مثال، جریان یا ترافیک ۸ که بوسیله ۳ گره متوالی از ۵۰، ۱۰۰ و ۲۰۰ کاربر در سه آزمایش جداگانه ایجاد شده است.

درخواست های مشتریان از طریق یک شبکه اینترنت با سرعت ۱۰۰mbps به یک وب سرور لینوکس رانده می شود.

در حقیقت وب سرور یک صفحه به نام درخواست های ورودی/خروجی را برای ۱، ۲ یا هر ۴ دیتا سرور که از برنامه my SQL استفاده می کنند، ایجاد می کند که شامل یک جدول فهرست شده با ۲۹ فیلد است که ۱۱۵۵۲ رکورد را در بر دارد. در مواردی که دیتا بیس سرورهای چندگانه استفاده می شوند، یک دیتا بیس جایگزین دیتا بیس های هر گره می گردد. بنابراین تقاضا یا درخواست برای داده بوسیله هر کدام از ۴ دیتا بیس می تواند اجرا گردد و نتیجه یکسانی را ارائه دهد. روش های متفاوتی به منظور تعیین دیتا بیس سروری که یک درخواست را دریافت می کند وجود دارد.(اگر چند دیتا بیس سرور مورد استفاده قرار گیرد.)

در روش ترتیبی، درخواست ها یک سیر ترتیبی را طی می کنند. اول سرور اول بعد دومی بعد سومی و بعد چهارمی و سپس دوباره اولی. در مدل تصادفی یک عدد تصادفی به منظور انتخاب یک سرور از میان چند سرور مورد استفاده قرار می گیرد. مدل LB سیستم عملیاتی هر گره در هر دیتا بیس را چک کرده و میزان بار کاری آنها را در همان زمان مشخص می کند. دیتا بیس سروری که بار کاری سنگینی دارد و قادر به ارائه پاسخ در زمان معین نیست را اصطلاحا دیتا بیس سروری گویند که در بهره برداری ۱۰۰٪ است. انتخاب در این مدل بر مبنای کمترین سطح بهره برداری است. اطلاعات جمع آوری شده در مجموعه ای از جداول گزارش داده می شود. یکی از این جداول در زیر نمایش داده شده است.

Table 1: 8 Consecutive Iterations of 50 Concurrent Sessions.

Query Distribution Type	Sequential Iterations	Server Nodes	Clients	Average Delay (ms)	Throughput (bytes/s)	Packet Intensity (packets/s)
N/A	8	1	50	2.07193316	92758.467	241.321
Sequential	8	2	50	0.74688173	191275.131	669.450
Sequential	8	4	50	0.39466387	312233.329	1266.901
Random	8	2	50	0.71621023	167432.264	698.119
Random	8	4	50	0.47683332	275472.700	1048.584
Load Balanced	8	2	50	0.17195826	252105.315	2907.682
Load Balanced	8	4	50	0.08090560	522526.965	6180.042

داده هایی که از ۵۰ کاربر جمع آوری شده است در جدول ۱ نمایش داده شده است.هر کدام از آزمایش ها در سطح ۵۰ کاربر برای هر کدام از روش ها و هر ترکیب از گره های موجود در دیتا بیس ها به طور جداگانه اجرا شده است. زمـانی که بار کاری افزایش می یابد، تاوت در سطح عملکرد واضح تر می گردد، به عنوان نتیجه با اضافه شدن هر گره به دیتا بیس سرور سطح عملکردی بالاتر می رود.

- اولین ستون جدول روش تخصیص درخواست ها به گره های دیتا بیس را نشان می دهد. این مفهوم زمـانی کـه فقط یک دیتا بیس داریم معنایی ندارد.
- ستون دوم سطح ترافیک را نشان می دهد.
- ستون سوم تعداد دیتا بیس سرورهای به کار گرفته شده را توصیف می کند.
- ستون چهارم تعداد کاربران حاضر در آزمایش را که تولید بار کاری می کنند را نشان می دهد.
- ستون پنجم نشان دهنده میانگین زمان انتظار هر کاربر برای دریافت پاسخ است.
- ستون ششم نشان دهنده سطح عملکردی با واحد bps است.
- ستون آخر نشان دهنده شدت ترافیک بسته ها است.

Table 2: 8 Consecutive Iterations of 100 Concurrent Sessions.

Query Distribution Type	Sequential Iterations	Server Nodes	Clients	Average Delay (ms)	Throughput (bytes/s)	Packet Intensity (packets/s)
N/A	8	1	100	3.88490715	44360.966	128.703
Sequential	8	2	100	0.71031160	197973.048	703.916
Sequential	8	4	100	0.37551237	361269.535	1331.514
Random	8	2	100	0.63998721	202004.413	781.266
Random	8	4	100	0.44903326	312168.375	1113.503
Load Balanced	8	2	100	0.13790886	318776.122	3625.583
Load Balanced	8	4	100	0.08812921	484603.770	5673.488

Table 3: 8 Consecutive Iterations of 200 Concurrent Sessions.

Query Distribution Type	Sequential Iterations	Server Nodes	Clients	Average Delay (ms)	Throughput (bytes/s)	Packet Intensity (packets/s)
N/A	8	1	200	4.80601741	17878.602	104.036
Sequential	8	2	200	5.19456850	21524.983	96.254
Sequential	8	4	200	0.34005095	330987.386	1470.368
Random	8	2	200	13.61430900	8529.467	36.726
Random	8	4	200	0.89513973	157894.511	558.572
Load Balanced	8	2	200	0.10743538	424712.763	4653.961
Load Balanced	8	4	200	0.05969465	724683.596	8375.961

مقایسه مقادیر موجود در سطح های مختلف مشتریان به طور گرافیکی بهتر نشان داده مـی شـود و شـکل هـای ۲-۱ مقادیر مشاهده شده مربوط به متغیر های، زمان تأخیر میانگین، شدت ترافیک پکت ها، توان عملیاتی را نـشان خواهـد داد.

شکل های ۴-۲ میانگین زمان تأخیر مربوط به سه مدل ترتیبی، تصادفی و LB را به ترتیب نشان مـی دهـد. جزئیـات مربوط به زمان های جلسه و حداکثر بار مفید پکت ها در مدل های ترتیبـی و تـصادفی و LB بـا بـار ۵۰، ۱۰۰ و ۲۰۰ کاربر به طور همزمان موجود است.

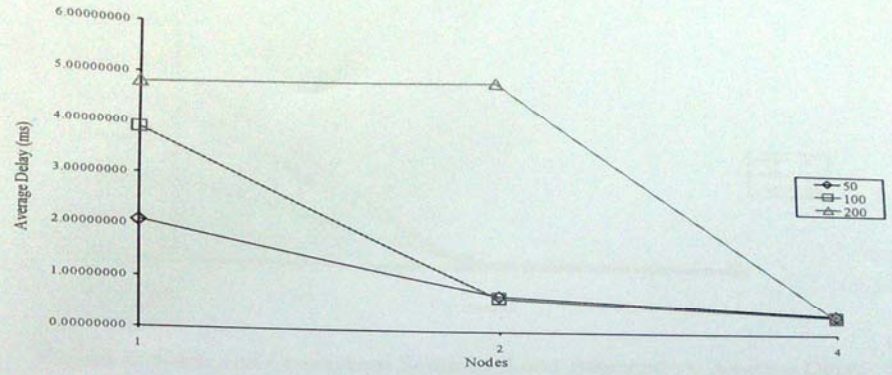


Figure 2: Series of Concurrent Sessions. Sequential Nodes vs. Average Delay

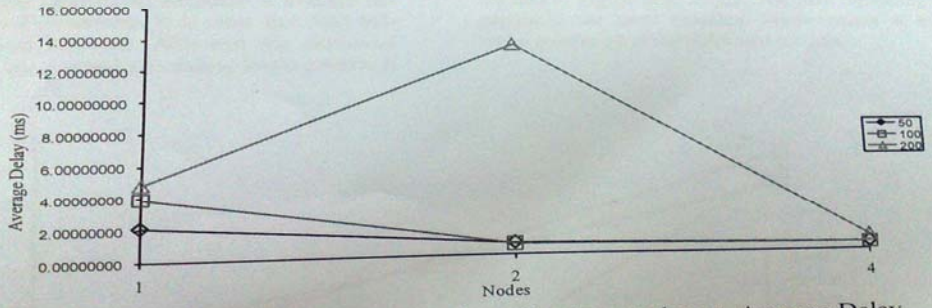


Figure 3: Series of Concurrent Sessions. Random nodes vs. Average Delay

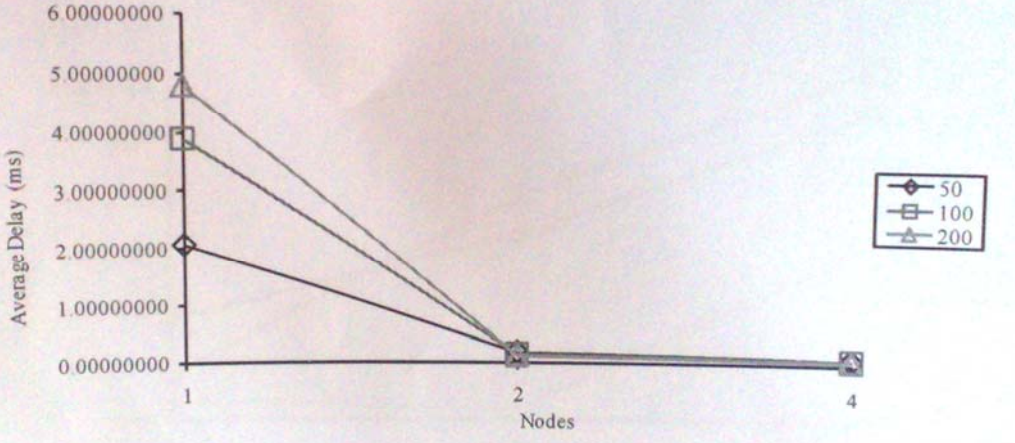


Figure 4: Series of Concurrent Sessions. Load Balanced vs. Average Delay

در تمامی مدل ها، زمان تأخیر زمانی که دیتا بیس ها افزایش می یابد،کاهش می یابد هر چند در مواردی در مدل های ترتیبی و به طور گسترده در مدل های تصادفی، زمان تأخیر به هنگام حرکت از ۱ سرور به ۲ تا افزایش می یابد. با به کار بردن ۴ دیتا بیس سرور در زمان تأخیر بهبود مشاهده می گردد. این کاملا واضح است که مدل LB کاراترین مدل است. اگر چه مدل ترتیبی الگوی کاهش خطی مطلوبی را نشان می دهد اما به اندازه مدل LB شناخته شده نیست.مدل تصادفی بر اساس محاسبات انجام شده در سطح ۲ دیتا سرور از لحاظ ضررو زیان کاراتر است اما در سطوح بالاتر از آن نسبت به دو مدل دیگر از کارایی کمتری برخوردار است. مدل LB در مقایسه با بقیه روش ها بهبود چشم گیری را در تمامی سطوح ایجاد می کند.

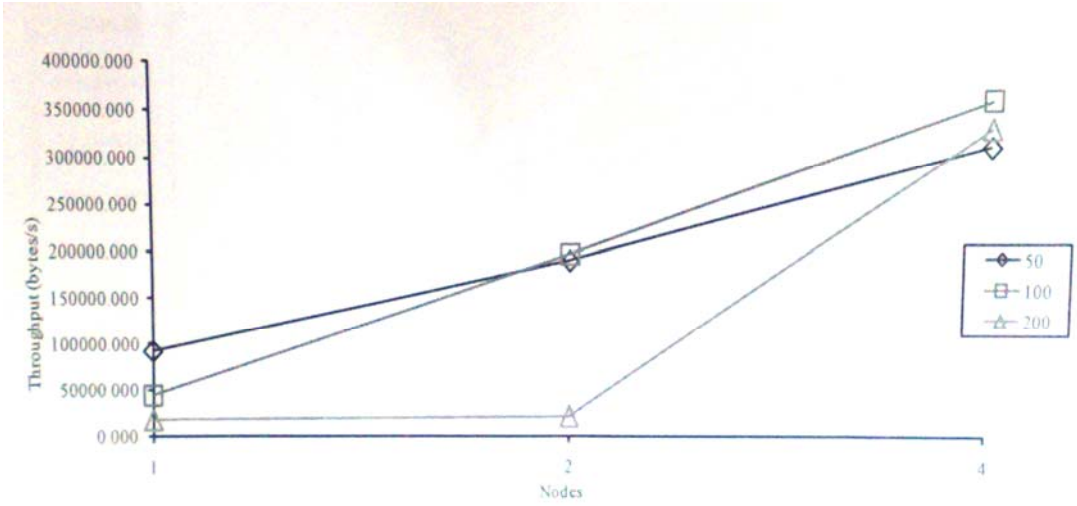


Figure 5: Series of Concurrent Sessions. Sequential Nodes vs. Throughp

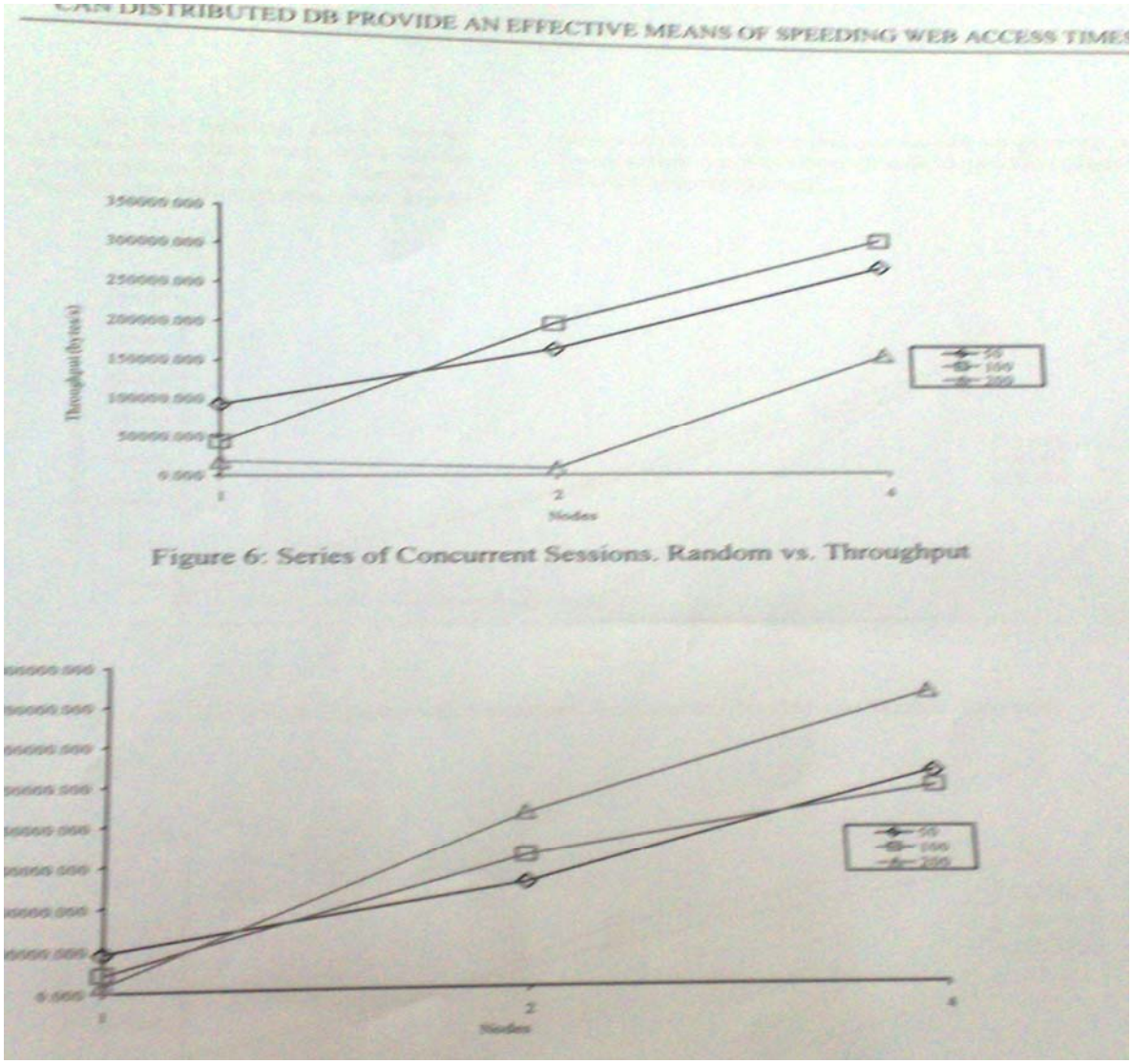


Figure 6: Series of Concurrent Sessions. Random vs. Throughput

با کاهش در زمان تأخیر (از طریق افزودن درتا بیس سرورهای اضافی) افزایش سطح توان عملیاتی مورد انتظار است. نتایج مربوط به توان عملیاتی به اندازه نتایج مربوط به تأخیر شگفت آور و عجیب نیست. با اضافه کردن دیتا بیس سرورها اضافی، یک روند خطی در افزایش توان عملیاتی در مامی مدل ها مشاهده می گردد. با استفاده از مدل تصادفی، نتایج حاصله از ۲ و ۴ دیتا بیس سرور به طور نزدیکی شبیه به هم اند و همدیگر را تأکید می کنند. این تناسب می تواند تا حد زیادی با اثر بار اضافی بر الگوریتم تصادفی، مرتبط باشد. آزمایشات دیگری مورد نیاز است تا مشخص گردد که در چه زمانی سطح آستانه توان عملیاتی بوسیله اضافه کردن دیتا بیس سرورهای اضافه تقویت می گردد و نتایج حاصله از مدل ترتیبی را با نتایج حاصله از مدل LB مقایسه نماید. مدل ترتیبی یک روند غیر خطی را نشان می دهد، که نشان دهنده سطح بازده بالاتری برای هر دیتا بیس سرور اضافی است. هر چند باید به این مسئله توجه شود که توان عملیاتی LB در سطح۴ دیتا بیس سرور۷۲۴/۶۸۳ bps است در حالی که در سطح ۴ دیتا بیس سرور توان عملیاتی ۳۳۰۹۸۷ bps رانشان می دهد.

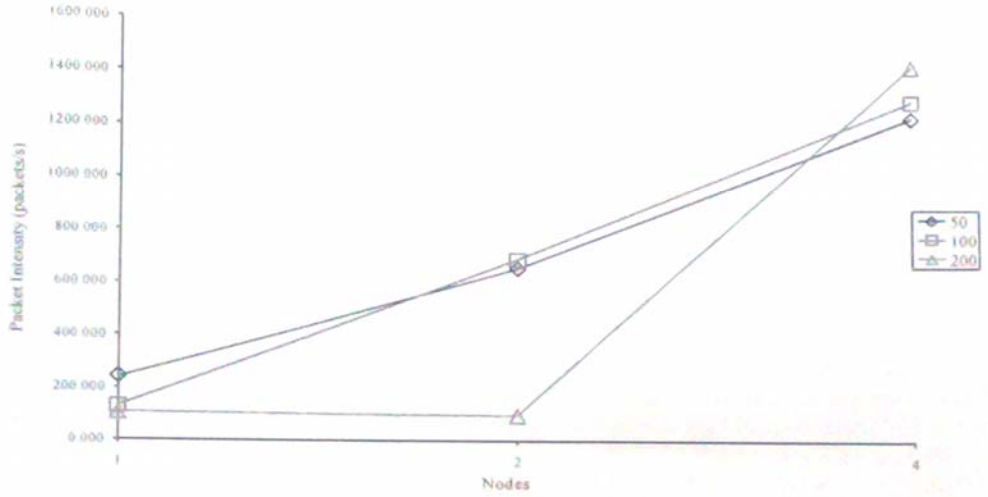


Figure 8: Series of Concurrent Sessions. Sequential Nodes vs. Packet Intensity

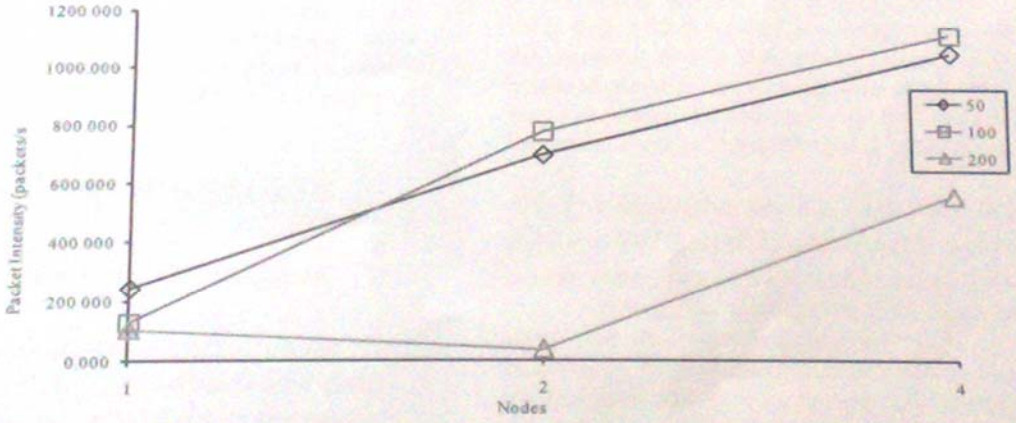
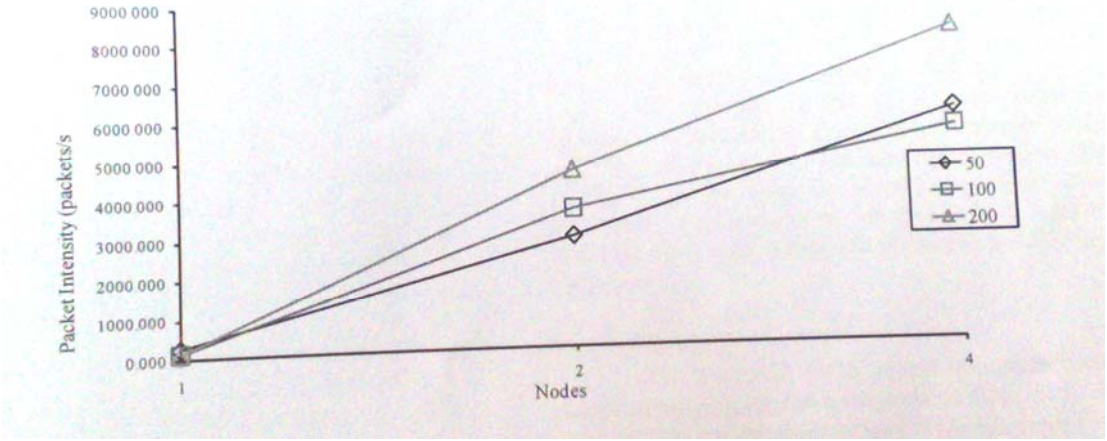


Figure 9: Series of Concurrent Sessions. Random Nodes vs. Packet Intensity



در تمامی مدل ها، ترافیک (شدت جریان) پکت ها عموما افزایش می یابد زمانی که تعداد دیتا بیس ها افزایش می یابد، هر چند در مواردی در مدل تصادفی، زمانی که بار کاری زیادی به مدل تحمیل می شود منجر به هزینه بر بودن آن می گردد. مدل LB کاراتر از مدل ترتیبی است. مدل LB با ۴ دیتا بیس سرور در سطح ۲۰۰ تماس و با ۴۶۵۴ بسته در ثانیه به نقطه اوج یا سر حد نهایی خود می رسد در حالی که مدل ترتیبی با ۱۴۷۰ بسته در ثانیه به این حد می رسد. نتایج حاصله از مدل تصادفی با ۲ و ۴ دیتا بیس سرور به طور نزدیکی به هم مرتبط اند و تقریبا با هم سازگارند. این تناسب تا حد زیادی به بررسی اثر با اضافی در الگوریتم تصادفی مربوط می گردد. به علاوه آزمایش مورد نیاز است تا پیش بینی نماید که در چه زمانی حد آستانه ترافیک پکت ها با اضافه کردن دیتا بیس سرورهای اضافی بهبود می یابد و نتایج حاصله در مدل ترتیبی را با نتایج مدل LB مقایسه می نماید.

بحث در مورد نتایج :

رد سه فرضیه صفر:

H_1 : شدت بارکاری در مدل دیتا بیس توزیع یافته بر زمان بازیابی اطلاعات و مدت زمان تأخیر مشتریان ندارد. زمانی که از ۵۰ درخواست همزمان به ۲۰۰ تا تغیر وضعیت می دهیم، زمان تأخیراز ۲/۰۷ به ۴/۸۱ میلی ثانیه افزایش می یابد. اضافه کردن گره های دیتا بیس سرور، بار کاری را با روش LB در میان ۴ گره توزیع می کند و افزایش تعداد درخواست ها از ۵۰ به ۲۰۰ زمان تأخیر را از ۲/۰۷ به ۰/۰۶ میلی ثانیه کاهش می دهد. بنابراین فرضیه H_1 باید رد شود.

H_2 : تعداد گره های یک دیتا بیس بر زمان پاسخگویی به مشتریان ثری ندارد. با استفاده از روش LB و حرکت از ۱ دیتا بیس به ۴ تا، تحت بار کاری ۵۰ درخواست همزمان، میانگین زمان تأخیر از ۲/۰۷ به ۰/۰۸ میلی ثانیه کاهش می یابد. تعیین بار کاری ۲۰۰ د خواست همزمان، استفاده از روش LB و حرکت از۱ به ۴ دیتا بیس سرور زمان تأخیر را از ۴/۸۶ به ۰/۰۶ میلی ثانیه کاهش می دهد. بنابراین فرضیه H_2 باید رد گردد.

H_3 : الگوی مورد استفاده در توزیع درخواست ها به گره مورد نظر در سیستم دیتا بیس های توزیع یافته اثری بر زمان تأخیر مشتریان ندارد.

با بار کاری ۳۰ درخواست همزمان، استفاده از ۴ دیتا بیس سرور و سپس تغییر از مدل LB به مدل ترتیبی، میانگین زمانی تأخیر از ۰/۰۸ به ۰/۳۹ ملی ثانیه افزایش می یابد و وقتی مدل LB را به مدل تصادفی تبدیل می کنیم زمان تأخیر از ۰/۰۸ به ۰/۴۸ میلی ثانیه افزایش می یابد. زمانی که بار کاری به ۲۰۰ افزایش می یابد، با استفاده از ۴ دیتا بیس سرور و تغییر از روش LB به روش ترتیبی میانگین زمان تأخیر از ۰/۰۶ به ۰/۳۴ میلی ثانیه افزایش می یابد و با تغییر به روش تصادفی این زمان به ۰/۸۹ میلی ثانیه افزایش می یابد بنابراین فرضیه H_3 باید رد شود.

عملکرد حاصله در ارتباط با اضافه کردن دیتا بیس سرور:

در مدل ترتیبی با حرکت از ۱ دیتا بیس سرور به ۴ دیتا بیس سرور(تحت بار کاری ۵۰ در خواست همزمان) زمان تأخیر از ۲/۰۷ به ۰/۳۹ کاهش می یابد. این اثر زمانی که بار کاری به ۲۰۰ درخواست همزمان می رسد بیشتر است و منجر به ایجاد تغییراز ۴/۸ به ۰/۳۴ میلی ثانیه می گردد.

این بسیار مشکل است که میزان(توازن، درجه) مدل LB را بسنجیم، زمانی که این مدل حتی در سطح بار کاری ۵۰ و با تغییر به ۴ دیتا بیس سرور با همان شرایط قبلی صادق است. نتیجه کاهش زمان تأخیر از ۴/۰۸ به ۰/۰۶ می باشد. مدل تصادفی کمترین اثر را بر زمان تأخیر دارد، در مقایسه با کاهش از ۲/۰۷ به ۰/۲۹ میلی ثانیه د رمدل ترتیبی و کاهش از ۲/۰۷ به ۰/۰۸ میلی ثانیه در مدل LB، مدل تصادفی تنها کاهش ۲۲/۰۷ به ۰/۷ میلی ثانیه را در میانگین زمان تأخیر برای بار ی ۵۰ تماس همزمان ارائه می دهد، زمانی که از یک دیتا بیس سرور به ۴ دیتا بیس سرور تغییر می کند. اضافه کردن تعداد دیتا بیس سرورها از ۱ به ۴ با بار کاری ۲۰۰ تماس همزمان در مدل تصادفی باز هم باعث کاهش میانگین زمان تأخیر از ۴/۸۱ به ۰/۸۹ میلی ثانیه می گردد. این از تغییرات به ۴/۸۱ الی ۱۳/۶۱ افزایش می یابد زمانی که در بارکاری ۲۰۰ تماس همزمان ۲ دیتا بیس سرور به کار می بریم.

توان عملیاتی : در مدل ترتیبی زمانی که از ۱ دیتا بیس سرور به ۴ تا تغییر وضعیت می دهیم بهبود در عملکرد را مشاهده می کنیم. تحت بار کاری ۵۰ تماس همزمان، استفاده از ۴ دیتا بیس سرور و با به کارگیری مدل ترتیبی توان عملیاتی از ۹۲۷۵۸ به ۳۱۲۲۳۳ BPS افزایش می یابد. تغییر وضعیت از ۱ دیتا بیس سرور به ۴ دیتا بیس سرور و

تحت بار کاری ۲۰۰ باعث افزایش توان عملیاتی از ۱۷۸۷۹ BPS به ۳۳۰۹۸۷ bps می گردد. مدل LB توازن زیاد توان عملیاتی را منجر می گردد. در بار کاری ۵۰، زمانی که از ۱ دیتا بیس سرور به ۴ تا تغییر وضعیت می دهیم، توازنعملیاتی از ۹۲۷۵۸ به ۵۲۲۵۲۷ bps افزایش می یابد. در همین وضعیت و در بار کاری ۲۰۰ توان عملیاتی از ۱۷۸۷۸ به ۷۲۴۶۸۴ افزایش می یابد.

در مدل تصادفی زمانی که از ۱ دیتا بیس سرور به ۲ دیتا بیس سرور تغییر وضعیت می دهیم و تحت بار کاری ۲۰۰، توان عملیاتی از ۱۷۸۷۸ به ۱۵۷۸۹۴ bps افزایش می یابد. هر چند زمانی که از ۱ دیتا بیس سرور به ۲ دیتا بیس سرور تغییروضعیت می دهیم، در بار کاری ۲۰۰ توان عملیاتی به ۸۵۲۹ bps کاهش می یابد.

ترافیک یا شدت پکت ها: در مدل ترتیبی با افزایش میزان بار کاری تا سطح ۲۰۰، با تغییر وضعیت از ۱ به ۴ دیتا بیس سرور سطح شلوغی و ترافیک از ۱۰۴/۰۴ به ۱۴۷۰/۳۷ bps افزایش می یابد. تحت بار کاری ۲۰۰ و ۱ دیتا بیس سرور در مدل ترتیبی وقتی به ۲ دیتا بیس سرور تغییر وضعیت می دهیم، ترافیک به طور نسبی از ۱۰۴/۰۳ به ۹۶/۲۵ bps کاهش می یابد. در مدل تصادفی میزان ترافیک بسته زمانی به حد نهایی خود می رسد که در حالت بار کاری ۱۰۰ تماس همزمان از ۱ دیتا بیس سرور به ۴ تا تغییر وضعیت می دهیم و مقدار ترافیک از ۱۲۸/۷۰ به ۱۱۱۳/۵۰ bps افزایش می یابد. با یک دیتا بیس سرور و تحت بار کاری ۲۰۰ زمانی که به ۲ دیتا بیس سرور تغییر وضعیت می دهیم ترافیک از ۱۰۴/۰۴ به ۳۶/۷۳ کاهش می یابد. هر چند زمانی که تحت شرایط مشابه از ۱ دیتا بیس سرور به ۴ تا تغییر وضعیت می دهیم سطح ترافیک از ۱۰۴/۰۴ به ۵۵۸/۵۷ bps افزایش می یابد.

در مدل LB و با بار کاری ۲۰۰ سطح ترافیک زمانی به حد نهایی خود می رسد که از ۱ دیتا بیس سرور به ۴ تا تغییر وضعیت می دهیم و سطح ترافیک از ۱۰۴/۰۴ به ۸۳۷۵/۹۶ bps افزایش می یابد که بالاتری رکورد افزایش در میان ۳ مدل است. مدل LB در سطح با ر کاری ۲۰۰ زمانی که از ۲ دیتا بیس سرور به ۴ تا تغییر وضعیت می دهیم افزایش نسبتا جزئی (از ۴۶۵۳/۹۶ به ۸۳۷۵/۹۶) را در سطح ترافیک نشان می دهد. این کاملا واضح است که با اضافه کرن سرورهای اضافی در مدل تصادفی و LB عملکرد بهبود می یابد. در سطح بار کاری بالاتر، افزایش عملکرد به ترتیب در مدل های ترتیبی و LB قابل توجه تر است.

عملکرد حاصله از مدل های تخصیص متفاوت:

بیشترین کاهش در میانگین زمان تأخیر در مدل LB حاصل می شود، با بار کاری ۲۰۰ زمانی که از ۱ دیتا بیس سرور به ۴ تا تغییر وضعیت می دهیم کاهش زمانی معادل ما به تفاوت ۴/۸۱ و ۰/۰۶ میلی ثانیه حاصل می شود. زمانی که به بررسی میانگین زمانی تأخیر می پردازیم مدل LB کاهش اساسی را نشان می دهد. (زمانی که از ۱ دیتا بیس سرور به ۲ تا و سپس به ۴ تا تغییر وضعیت می دهد، تحت بار کاری ۵۰ تا ۴۰۰ تماس همزمان)

در روش تصادفی، زمانی که از ۱ دیتا بیس سرور به ۲ تا تغییر وضعیت می دهیم با بار کاری ۲۰۰، به بالاترین میانگین زمان تأخیر ثبت شده مواجه می شویم که در واقع افزایشی از ۴/۸۱ به ۱۳/۶۱ میلی ثانیه را منجر می گردد. مدل تصادفی در سطح بار کاری ۱۰۰ تماس همزمان کاهشی در میانگین زمان تأخیر را ارائه می دهد که با تغییر از ۳/۸۸ به ۰/۴۵ میلی ثانیه به نقطه اوج خود می رسد.(زمانی که از ۱ دیتا بیس سرور به ۴ تا تغییر وضعیت می دهیم.)

در مدل ترتیبی میانگین زمان تأخیر، با سطح ۵۰ تماس همزمان، تغییر از ۱ دیتا بیس سرور به ۲ تا، از ۲/۰۷ به ۰/۷۵ کاهش می یابد و در همین شرایط با تغییر از وضعیت ۱ دیتا بیس سرور به ۴ تا میانگین زمان تأخیر از ۲/۰۷ به ۰/۳۹ میلی ثانیه کاهش می یابد. در مدل ترتیبی زمانی که از وضعیت ۱ دیتا بیس سرور به ۲ یا ۴ تغییر وضعیت می دهیم، میانگین زمان تأخیر کاهش می یابد چه بار کاری ۵۰ باشد چه ۲۰۰۰. تنها استثنا زمانی اتفاق می افتد که ما از ۱ دیتا بیس سرور به ۲ تا تغییر وضعیت می دهیم که منجر به افزایش زمان تأخیر از ۴/۸۱ به ۵/۱۹ میلی ثانیه می گردد.

اثر ترافیک و میزان شلوغی مشتریان بر متدولوژی طراحی:

بار کار بالاتر به ورت غیر قطعی منجر به سطوح اشباع بالاتری از حد بهینه سرور می شود. مشکل زمانی مشاهذه می گردد که سطح بار کاری ۸۰۰ درخواست همزمان از یک siege client دریافت می شود. Siege clientهای اضافی بوسیله توزیع تعداد درخواست های همزمان به صورت برابر در میان ۲ تا siege client، بهینه می شود. زمانی که siege client اضافی، اضافه می کنیم، این کاملا واضح است که ۴ دیتا بیس سرور به منظور اداره ۸۰۰ تعداد درخواست کافی نیست و زمانی که فعالیت های فعال همزمان ااز سطح ۲۸۵ بالاتر می رود عملیات قطع می گردد. siege در دوره های زمانی تعریف شده، زمانی که تعداد درخواست ها ناکافی دریافت شود، متوقف می گردد.(عملیات خود را قطع می کند) این اثر متوقف کننده در مدل های تصادفی و ترتیبی منجر می گردد زمانی که سرورها به علت تقاضاهای پردازشی ارسال شده به حد اشباع رسیده اند، تقاضای جدید قبول نکنند.

زمان بازیابی اطلاعات در سرورها هم مفهوم با ارزشی است. در بسیاری از موارد شروع به کار دوباره دیتا سرور ها در بین فواصل زمانی بررسی ها ضروری نیست. هر چند یک دوره زمانی خاموشی ۴-۲ دقیقه ای وجود دارد – زمانی که صلاح در این است که پیش از انجام بررسی ها و درخواست های قبلی درخواست جدیدی قبول نکنند. سروربابد تا زمان استراحت بعدی درخواست های ورودی را قبول کند. موارد کمی وجود دارد که در آن siege به جای پافشاری برای انجام یک درخواست، error بدهد. که این موارد شامل زمان هایی است بررسی تقاضاها قبل از دریافت ۱۰۰۰۰۰ پکت ها کامل شده اند. این موضوع نشان دهند این است که یکی از سرورها دارای یک سیاست امنیتی بوده است که فرآیند HTTP را غیر فعال کرده بوده است.

ترکیب پیشنهادی سرورها و روش توزیع درخواست ها:

به طور واضح کارایی روش LB از روش های تصادفی و ترتیبی بیشتر است. بهبود های پیشنهادی مربوط به محیط آزمایش شامل دو مورد زیر است:

۱) دو برابر کردن دیتا بیس ها از ۴ تا به ۸، به کارگیری ۲ وب سرور که هر سرور برنامه متفاوتی را ارائه می دهد، تخصیص اتوماتیک هر دیتا بیس سرور به برنامه وب بیس مورد نیاز و آزاد کردن دیتا بیس سرورها و تخصی درخواست ها به سرورهای دیگر زمانی که بار کاری به علت افزایش تقاضای مشتریان تحت وب افزایش می یابد.

۲) افزایش تعداد دیتا بیس سرورها به ۳۲، استفاده از مدل ها LB، و تست کردن قدرت هر کدام از ۲ وب سرور با استفاده از ۲،۴،۸،۱۶ و ۳۲ دیتا بیس سرور و تحت بارهای کاری ۴۰۰،۸۰۰ و ۱۶۰۰ درخواست همزمان. این ها همه مربوط به موردی است که ما ۴ تا ۸ siege clientرا به کار می بریم و درخواست های همزمان را به صورت برابر در میان siege client ها توزیع کنیم.

پیشنهاداتی برای تحقیقات مجدد به منظور اصلاح ابزارهای آزمایش و متدولوژی:

هر کدام از سرورها باید قبل از آزمایش اصلی ابتدا آزمایش شوند تا تعیین شود که سطح ظرفیت و آستانه آنها چقدر است. این حد آستانه می تواند ۵۰ تماس همزمان باشد. تعیین انحراف آماری بین هر کدام از بار کاری ها. تعیین علت عملکرد ضعیف مدل LB در سطح ۲۰۰ درخواست همزمان و سپس بر طرف شدن نسبی آن زمانی که به ۴۰۰ تماس همزمان تغییر وضعیت می دهیم.

تعیین توازن بوسیله افزایش دیتا بیس سرورها:

این کاملا واضح است زمانی که تعداد دیتا بیس سرورها افزوده می گردد افزایش عملکرد حاصل می شود. تعیین بار کاری که نشان دهنده حداکثر حد مجاز یک دیتا بیس سرور و نیاز به اضافه کردن یک دیتا بیس جدید است.

سؤال : در چه نقطه ای توصیه می گردد که وب سرورهای اضافی با درخواست هایی که به طور اتوماتیک تخصیص داده می شود اضافه گردد؟

افزایش شدت ترافیک مشتریان :

در دهه های اخیر، یک siege client می تواند تا حد ۱۶۰۰ مشتری را پاسخگو باشد از طریق ۸ مجموعه از ۲۰۰ تماس همزمان در فاصله زمانی ۴ دقیقه اضافه کردن siege client و توزیع بار به صورت برابر در بین آن دو مسئولیت مشتری بیشتری را حاصل کند.

داده ها می تواند با استفاده از TCPDUMP در هر siege client جمع گردد. داده ها تفسیر شده تجزیه و تحلیل توان عملکردی و زمان را امکان پذیر سازد.

دیتا بیس های بیشتر منجر به افزوده شدن عملکرد می گردد. این گفته خصوصا زمانی که از الگوریتم LB استفاده مـی کنیم درست است. هر چند انتظار داریم که در نقطه ای به ح بازده نزولی برسیم. داده های جمع آوری شده این حـد را نشان نمی دهند.

تحقیقات بیشتری مورد نیاز است تا به این سؤال جواب دهد. بنابراین به علت این که فقط تعداد محـدودی از گـره هـا مورد استفاده قرار گرفته است این مطالعه از نظر فرآیند فرضیه سازی بیشتر حائز اهمیت است تا به دسـت آوردن نتـایج پیمایشی.