ارائه ی معماری مدیریت داده‌های‌کلان درآموزش وپرورش

جواد افشاری
کارشناس ارشد مهندسی کامپیوتر گرایش نرم افزار، مدرس گروه کامپیوتر دانشگاه علمی کاربردی فرماندهی انتظامی خراسان جنوبی

چکیده

عبارت کلان‌داده مدت‌ها است که برای اشاره به حجم عظیمی از داده‌ها که توسط سازمان‌های بزرگی مانند گوگل یا ناسا ذخیره و تحلیل می‌شوند مورد استفاده قرار می‌گیرد. اما این اصطلاح بیشتر برای اشاره به مجموعه‌های بزرگی از داده‌ها استفاده می‌شود که به قدری بزرگ و حجیم هستند که با ابزارهای مدیریتی و پایگاه‌داده‌های رابطه‌ای و معمولی قابل مدیریت نیستند. مشکلات اصلی در کار با این نوع داده‌ها مربوط به توانایی گردآوری، ذخیره‌سازی، جست‌وجو، اشتراک‌گذاری، تحلیل و نمایش آن‌ها است. داده‌های ذخیره شده در آموزش‌وپرورش به‌قدری حجیم و متنوع هستند که انجام فرآیندهای مختلف روی این پایگاه‌داده‌ها با سرعت پایین انجام می‌گیرد. به‌نحوی که در پاره‌ای از مواقع پاسخ به کاربران در زمان مناسب ارائه نمی‌شود. ازطرفی قابلیت‌های محدود پایگاه‌داده‌های رابطه‌ای در ذخیره‌سازی داده‌ها و سرعت رشد داده‌های آموزش‌وپرورش، در آینده‌ی نزدیک با مشکلات عدیده‌ای در ذخیره‌سازی و ارائه‌ی گزارش و تحلیل از این داده‌ها روبرو خواهد شد.

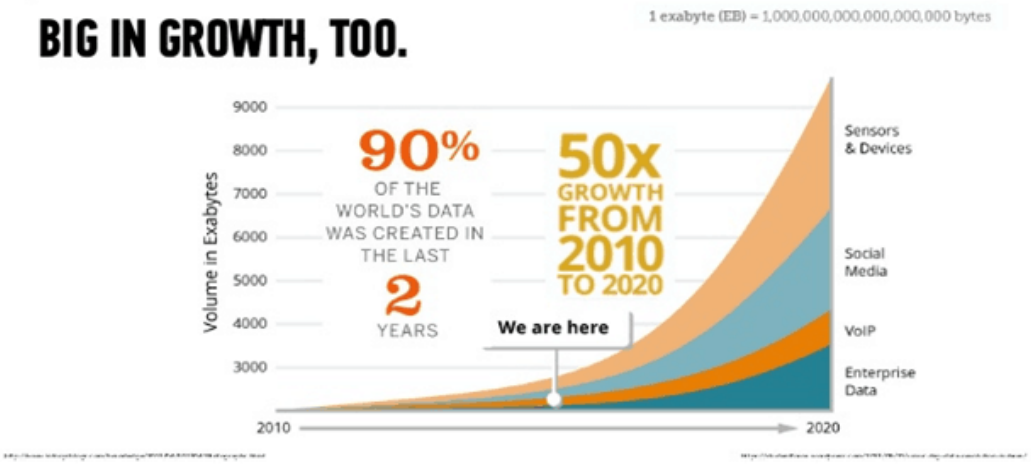
هدف این پژوهش، ارائه‌ی یک معماری مدیریت داده‌های کلان برای آموزش‌وپرورش به کمک ابزارهای موجود است، به‌نحوی که زیرساخت نرم‌افزاری مناسب برای مدیریت و تحلیل داده‌های کلان روی سخت‌افزارهای موجود پیاده‌سازی شود. با کمک این ابزارها، امکان تولید گزارش‌های درخواستی در بازه‌ی زمانی مناسب برای کاربران فراهم می‌شود.

واژگان کلیدی: داده‌کلان، حجم داده، پایگاه‌داده‌ی غیررابطه‌ای

مقدمه

امروزه بسیاری از متخصصان و حتی کاربرانی که در حوزه‌های مختلفی از فناوری اطلاعات فعالیت می‌کنند، از روند رشد چشمگیر داده‌ها و چگونگی پردازش آن‌ها شنیده‌اند و حداقل این نمودار رشد داده‌ها را مشاهده کرده‌اند. در مثال‌ها، بیشتر به شبکه‌های اجتماعی پرداخته می‌شود که کاربران آن‌ها در هر سال بیش از ۳ برابر سال قبل از آن، داده تولید می‌کنند.

عبارت کلان‌داده مدت‌ها است که برای اشاره به حجم عظیمی از داده‌ها که توسط سازمان‌های بزرگی مانند گوگل یا ناسا ذخیره و تحلیل می‌شوند مورد استفاده قرار می‌گیرد. اما به‌تازگی، این عبارت بیشتر برای اشاره به مجموعه‌های بزرگی از داده‌ها استفاده می‌شود که به‌قدری بزرگ و حجیم هستند که با ابزارهای مدیریتی و پایگاه‌داده‌های رابطه‌ای و معمولی قابل مدیریت نیستند. مشکلات اصلی در کار با این نوع داده‌ها مربوط به برداشت و جمع‌آوری، ذخیره‌سازی، جست‌وجو، اشتراک‌گذاری، تحلیل و نمایش آن‌ها است[۱]. نکته‌ی جالب توجه آن است که ۹۰ درصد داده‌هایی که اکنون در اختیار ما است، تنها در چند سال اخیر تولید شده است. شکل زیر رشد سریع داده‌ها در فاصله سال‌های ۲۰۱۰ تا ۲۰۲۰ را به خوبی نشان می‌دهد:



شکل(۱): حجم داده‌های تولید شده در فاصله‌ی سال‌های ۲۰۱۰ تا ۲۰۲۰ بر حسب اگزابایت[۲]

برخی از مهمترین اقدامات که بر روی کلان‌داده انجام گرفته عبارتند از[۳]:

ذخیره‌سازی داده‌های حجیم و بازیابی داده‌ها، توزیع داده‌ها، شبکه‌های پرسرعت کامپیوتری، عملکرد محاسباتی بالا، مدیریت موضوع و رویدادها، داده‌کاوی و تجزیه‌وتحلیل، یادگیری ماشین و تجسم سازی داده‌ها واضح است کلیه‌ی تکنیک‌های فوق در گذشته نیز مورد استفاده قرار گرفته‌اند، لیکن اهمیت آن‌ها را در موضوع کلان‌داده‌ها، گردآوری و ذخیره‌سازی داده‌های فراوان و نیز سخت‌افزارهای مناسب است. دیدگاه صنعتی و علمی کلان‌داده با سه مفهوم مشخص توصیف می‌شود[۳]:

۱) داده‌هایی که حجم آن‌ها بسیار زیاد است.

۲) داده‌هایی که در پایگاه‌داده‌های رابطه‌ای منظم طبقه‌بندی نمی‌شوند.

۳) داده‌هایی که با سرعت، ایجاد، دریافت و پردازش می‌شوند .

آن‌چه مسلم است، کلان‌داده درطول پنج سال آینده یکی از پنج موضوع حیاتی تحقیقاتی درزمینه‌ی تکنولوژی بوده و موجب تحول در بسیاری از زمینه‌ها از جمله تجارت، تحقیقات علمی و صنعتی، مدیریت اجتماعی، پزشکی و... خواهد گردید[۴].

مفهوم کلان‌داده دارای سه ویژگی اصلی است[۵]:

۱-حجم(Volume): داده‌ها به‌طورمیانگین سالیانه دو برابر می‌شوند. یک دلیل ساده برای آن این است که سازمان‌ها به صورت دوره‌ای اطلاعات خود را برای سال‌های متمادی نگه می‌دارند.

۲-سرعت(velocity): مفهوم آن این است که داده‌ها با چه سرعتی تولید می‌شوند و با چه سرعتی برای تحلیل و برآوردن آن‌چه مورد نیاز است پردازش می‌شوند. به‌عنوان مثال، سرعت تولید داده‌ی حسگرهای شناسایی فرکانس رادیوئی بسیار بالا

بوده که علاوه بر ذخیره‌سازی آن‌ها در لحظه، باید مورد تجزیه -وتحلیل نیز قرار بگیرند. واضح است سیستم‌های ذخیره‌سازی و تحلیل اطلاعات رابطه‌ای، به‌سادگی نمی‌توانند این جریان از اطلاعات را در لحظه مورد بررسی و نمایش قرار دهند. عکس‌العمل سریع در قبال سرعت فوق‌العاده تولید داده‌ها برای سازمان‌ها یک چالش بزرگ محسوب می‌شود.

۳-تنوع(Variety): امروزه اطلاعات درانواع ساختارهای متفاوت از سیستم‌های تولید داده‌ی استاندارد و یا تراکنش‌های پایگاه داده‌های تحلیلی آن‌لاین تولید می‌شوند. به‌عبارتی منابع تولید داده متفاوت بوده و موجب تولید داده‌ها با ساختارهای متفاوت می‌گردد. از یک‌نظر ساختارداده‌ها را می‌توان در سه دسته‌ی ساخت‌یافته، نیمه‌ساختار و بدون ساختار تقسیم‌بندی نمود.

علاوه بر موارد فوق، برخی شرکت‌های بزرگ ویژگی‌های دیگری را نیز برای کلان‌داده در نظر می‌گیرند:

۴-صحت(Veracity): شرکت IBM معتقد است نباید داده‌هایی که به‌آن اعتقاد نداریم را به‌کار بریم.

درواقع نمی‌توان انتظار داشت تمام داده‌ها، به‌طور کامل صحیح باشند، بلکه کیفیت داده‌ها به‌میزان زیادی وابسته به منابعی است که از آن‌ها گردآوری می‌شوند[۴].

۵-پیچیدگی(Complexity): کلان‌داده از منابع مختلف گردآوری شده‌اند و لازم است ارتباط میان آن‌ها به منظور دست‌یابی به داده‌های ساخت‌یافته و یا غیرساخت‌یافته، درنظر گرفته شود.

۶-ارزش(Value): باید درنظر داشت، مهم‌ترین ویژگی کلان‌داده، ارزش آن است. گرچه دسترسی به حجم بالای داده‌ها خوب است، اما تنها زمانی این دسترسی مفید است که داده‌ها، دارای ارزش واقعی باشند. دراین‌صورت، زیرساخت تکنولوژی برای ذخیره‌سازی و سرمایه‌گذاری تجاری کلان‌داده، بسیار ارزشمند خواهد بود[۴].

برخی دیگر از موارد کاربردی کلان‌داده‌ها را می‌توان در قالب زیر برشمرد:

۱-بهداشت و درمان: تحلیل کلان‌داده‌ها می‌تواند در صنعت بهداشت و درمان در قالب ارائه‌ی خدمات بهتر به عموم مردم کمک کند. شخصی‌سازی درمان می‌تواند منتج به افزایش سلامت جامعه و کاهش هزینه‌های دولت در بخش بهداشت و درمان شود.

۲-آموزش: کلان‌داده‌ها در صنعت آموزش به شخصی‌سازی فرایند یادگیری کمک می‌کند. این شخصی‌سازی به‌نوبه‌ی خود می‌تواند باعث شکوفایی استعدادهای دانش‌آموزان و دانشجویان شود و پویایی محیط یادگیری را افزایش دهد.

۳-تولید: در صنعت تولید استفاده از کلان‌داده‌ها می‌تواند به تولید طبق نیازهای مشتری کمک کند، زمان تولید محصول را کاهش دهد، همین‌طور با استفاده از شبیه‌سازی و بهینه‌سازی به‌کمک کلان‌داده‌ها می‌توان خط تولید را به‌صورت بهینه طراحی کرد و بسیاری از عیوب خط تولید کالاها را پیش از شروع به‌کار خط تولید شناسایی کرد.

۴-دولت: دولت می‌تواند از کلان‌داده‌ها برای ایجاد شفافیت، خدمت‌رسانی بهتر به مردم، استفاده‌ی بهینه از منابع محدود و تخصیص بودجه به فعالیتهای موجود استفاده کند. همین‌طور می‌تواند برای کمک و اطلاع‌رسانی در زمان بحران به مردم و مبارزه با فقر و جرم و جنایت کلان‌داده‌ها را بکار بگیرد.

۵-علوم اجتماعی: در مطالعات علوم اجتماعی کلان‌داده‌ها ابزاری برای بررسی پیچیدگی رفتار فردی و اجتماعی انسان‌ها بوده و دریچه‌ای جدید برای طرح سوال‌های جالب‌تر و الگوهای ناشناخته باشد.

۶-ورزش: در علوم ورزشی از کلان‌داده‌ها برای افزایش کارایی ورزشکاران در تمرین و مسابقه، پیشگیری از بروز مصدومیت و یافتن بهترین راهبرد برای مسابقات پیش‌رو استفاده می‌شود[۵].

در واقع رشد فزاینده‌ی کاربردهای کلان‌داده در زمینه‌های مختلف، از توانایی‌های نرم‌افزاری به‌کاررفته با هدف دریافت، مدیریت و پردازش این حجم داده‌ها بسیار فراتر است. مهمترین چالش در کاربرد کلان-داده کاوش در حجم انبوهی از داده‌ها و استخراج داده‌های مفید و قابل استفاده برای کاربردهای آینده است. از سویی این حجم بی‌سابقه از داده‌ها نیازمند سیستم‌های تحلیلی بسیار قدرتمندی است[۵].

رویکرد استفاده از صدها ابزار ذخیره‌سازی داده‌های کلان گرچه امری بی‌فایده تلقی می‌گردد، اما در کنار آن امکان ذخیره‌سازی صدها مجموعه داده با یک ترابایت داده فراهم می‌شود. بدین ترتیب می‌توان انتظار داشت امکان دسترسی مشترک به داده‌ها برای کاربران علی‌رغم سرعت پایین در پاسخ به عملیات تحلیلی فراهم می‌شود. با این‌حال این روش دو چالش اساسی دارد:

۱- اولین چالش زمانی است که سخت‌افزار دچار نقص شود. در این حالت امکان رخ دادن نقص سخت-افزار همزمان با افزایش تعداد آن‌ها به تناسب بیشتر می‌شود.
۲- دومین چالش زمانی است که بسیاری از عملیات‌های تحلیل، نیازمند گردآوری داده‌ها به‌روشهای مختلف است. به‌عنوان‌مثال، داده‌های خوانده شده‌ی یک دیسک نیازمند ترکیب باداده‌های ۹۹دیسک دیگرباشد[۵].

روش تحقیق

مفاهیم کلیدی مورد نیاز برای معماری پیشنهادی شامل پایگاه‌داده‌ی‌رابطه‌ای، مقیاس‌پذیری، دریاچه‌داده، هدوپ، ایمپالا می‌باشد. این مفاهیم را به‌طور خلاصه بررسی می‌نماییم.

۱.پایگاه‌داده‌ی رابطه‌ای

مفهوم سیستم‌های مدیریت پایگاه‌داده(DBMS) از قریب چهار دهه‌ی گذشته توسعه یافت. سیستم‌های مدیریت پایگاه‌داده‌ی رابطه‌ای، شامل نرم‌افزارهایی هستند که برای مدیریت مجموعه‌ای از داده‌های مرتبط و همه‌ی متعلقات آن‌ها به کار می‌روند. براین اساس، سیستم‌های مدیریت پایگاه‌داده یا از نوع متمرکز هستند که پس از نصب بر روی یک سکو، به‌صورت فیزیکی توسط کامپیوترهای متعدد که از طریق شبکه توزیع شده‌اند، پردازش می‌شوند و یا به‌صورت توزیع‌شده، که در واقع شامل چندین پایگاه‌داده‌ی توزیع‌شده بر روی یک شبکه است که مانند یک سیستم واحد عمل می‌کنند. از رایج‌ترین آن‌ها، به سیستم مدیریت پایگاه‌داده‌ی رابطه‌ای(RDBMS) می‌توان اشاره کرد که در انواع گوناگونی از پایگاه‌داده‌ها مانند SQLSERVER، DB۲، Oracle و ... توسعه یافته است. پایگاه‌داده‌های رابطه‌ای، شامل جداولی هستند که داده‌ها را در قالب ماتریسی از ردیفها که هر کدام شامل ستون‌هایی با یک کلید اصلی یا کلید خارجی می‌باشند، ذخیره می‌کنند. تکنولوژی پایگاه‌داده‌های رابطه‌ای، بسیار سازگار و ثابت است که داده‌ها را در قالب جداول کاملا ساخت‌یافته ذخیره می‌نمایند. گرچه این نوع پایگاه‌داده‌ها برای انجام پردازش تراکنش آن‌لاین یا به اختصار OLTP مناسبند، در عین حال قادر به انتقال پاسخ سریع در صورت افزودن پردازنده یا حافظه بیشتر نمی‌باشند[۵].

۲.مقیاس‌پذیری

یک سامانه که کارآیی آن پس از افزودن سخت‌افزار جدید به‌تناسب ارتقا یافته است، یک سیستم مقیاس‌پذیر نامیده می‌شود. این اصطلاح برای الگوریتم‌های مناسب برای اجرا در موقعیت‌های بزرگتر نیز به‌کار می‌رود. با دو رویکرد می‌توان یک سامانه را پس از افزودن منابع سخت‌افزاری مقیاس‌پذیر نمود:
اولین رویکرد، زمانی است که یک سیستم به‌صورت عمودی مقیاس‌پذیر می‌گردد. در این حالت منابع به یک گره تنها در یک سامانه افزوده می‌شود که در حالت رایج، شامل افزودن یک پردازنده یا حافظه به یک رایانه است. مقیاس‌پذیری عمودی سیستم‌های موجود، آن‌ها را در استفاده از تکنولوژی مجازی‌سازی توانمند می‌سازد و منابع بیشتری برای استفاده‌ی سیستم‌های عامل میزبانی شده و برنامه‌های کاربردی اشتراک‌گذاری شده، دراختیار قرار می‌دهد.
دومین رویکرد، زمانی است که یک سیستم به صورت افقی مقیاس‌پذیر می‌شود. در این حالت گره‌های جدیدی به سامانه افزوده می‌شود. افزودن کامپیوتر جدید به یک نرم‌افزارکاربردی توزیع شده، افزودن یک وب‌سرور به یک سامانه با تعداد ۳ عدد وب‌سرور مثال‌هایی از این نوع مقیاس‌پذیری است[۶].

۳.دریاچه‌داده

دریاچه‌داده یک راه حل برای مدیریت داده‌های سازمانی است که مقدار زیادی از داده‌ها در یک مخزن بزرگ برای تجزیه‌وتحلیل بیشتر ذخیره می‌شوند[۷]. در واقع دریاچه‌داده یک مخزن بسیار بزرگ از داده‌های خام در همان قالب اصلی خود است که داده‌ها را برای مدت زمان طولانی و تا زمان نیاز در خود نگهداری می‌کند.
رویکرد دریاچه‌داده در برخی جنبه‌ها دارای تفاوت‌هایی با سایر موارد مشابه است :

- داده‌ها کاملاً در دریاچه‌داده قرار می‌گیرند و از هیچ داده‌ای صرف‌نظر نمی‌شود.
- داده‌های پایین‌ترین سطوح(مثلاً توضیحات یک فرد در یک مقاله یا یک سایت) بدون تغییر یا با حداقل تغییرات به دریاچه‌داده منتقل می‌شوند. این رویکرد ذخیره‌سازی داده‌ها که در آن داده، بدون توجه به ساختار و منبع ذخیره می‌شود را به اصطلاح خواندن با ساختار می‌نامند. داده‌ها به‌صورت خام در فرمت‌های اولیه نگهداری می‌شوند و ابزارهایی در اختیار کاربران قرار داده می‌شود که بتوانند داده‌ها را با همین شکل مورد ارزیابی و تحلیل قرار دهند. گاهی این ارزیابی مشخص می‌کند چه‌بخشی از داده‌ها ارزش لازم برای تحلیل‌های کامل‌تر را دارند و ممکن است برای این بخش از داده‌ها، ساختار مورد نیاز را طراحی کنند.

از چالش‌های اصلی دریاچه‌داده، درک صحیح اهداف تحلیل‌های مورد نیاز و محدودیت‌های پیاده سازی به سبب در اختیار نبودن ابزارهای تکنولوژی را می‌توان برشمرد[۸]. رویکرد دریاچه‌داده در تضاد با رویکرد رابطه‌ای انبارداده است. در انبارداده، داده‌ها پیش از بارگذاری در یک شمای از پیش‌تعیین‌شده مورد تجزیه‌وتحلیل قرار می‌گیرند که باعث صرف زمان زیاد قبل از استفاده‌ی مفید داده‌ها و کاهش کارایی سیستم‌های مربوط می‌شود[۹]. ازدیدگاه علوم داده نگهداری همه‌ی داده‌ها به‌صورت کامل مفیدتر و کاربردی‌تر است. زیرا مشخص نیست کدام بخش از داده‌ها برای تحلیل‌های داده‌ای پیش‌رو دارای ارزش خواهند بود[۱۰]. انبارداده مبتنی بر عملیات استخراج، انتقال و بارگذاری داده‌ها یا به اختصار ETL می‌باشد که دیگر برای سرعت رشد تولید داده‌های امروزی مناسب نمی‌باشد[۱۱]. دریاچه‌داده بر خلاف رویکرد انبارداده، داده‌ها در فرم خام و اصلی خود ذخیره می‌شوند و لذا نیاز به فرآیند ETL نبوده و داده‌ها بلافاصله در اختیار کاربران قرار دارد. تفاوت بین دریاچه‌داده و انبارداده را می‌توان در حوزه‌های جدول زیر با یکدیگر مقایسه کرد[۱۲]:

جدول (۱): تفاوت دو رویکرد انبارداده و دریاچه‌داده در تحلیل داده‌ها

حوزه مقایسه	انبار داده	دریاچه داده
نوع داده ها	پردازش شده، ساختار یافته	ساختار یافته، نیمه ساختار یافته، غیر ساختار یافته، خام
نحوه پردازش	ساختار داده‌ها به‌طور دقیق تعریف شده و به صورت از پیش تعیین شده پردازش می‌شوند.(داده ها به صورت ساختار یافته و با هدف خاصی ذخیره می شوند.)	ساختار داده ها به طور دقیق تعریف نشده و به صورت خام ذخیره می شوند.(داده‌ها به صورت خام ذخیره می‌شوند و بعداً پردازش می‌شوند.)
هزینه ذخیره‌سازی	هزینه ذخیره‌سازی بالا برای حجم بسیار زیاد داده‌ها	هزینه ذخیره‌سازی پایین برای حجم بسیار زیاد داده‌ها
تغییر پذیری	تغییرات داده کم و به صورت مشخص	تغییرات داده زیاد و به صورت نامشخص
امنیت	بالا	در حال رشد
کاربران	تحلیل‌گران تجاری	دانشمندان داده و تحلیل‌گران فنی
روش تحلیل داده‌ها	تحلیل داده به صورت ساختار یافته و با هدف خاصی انجام می‌شود.	تحلیل داده به صورت خام و با هدف‌های مختلف انجام می‌شود.

۴.هدوپ

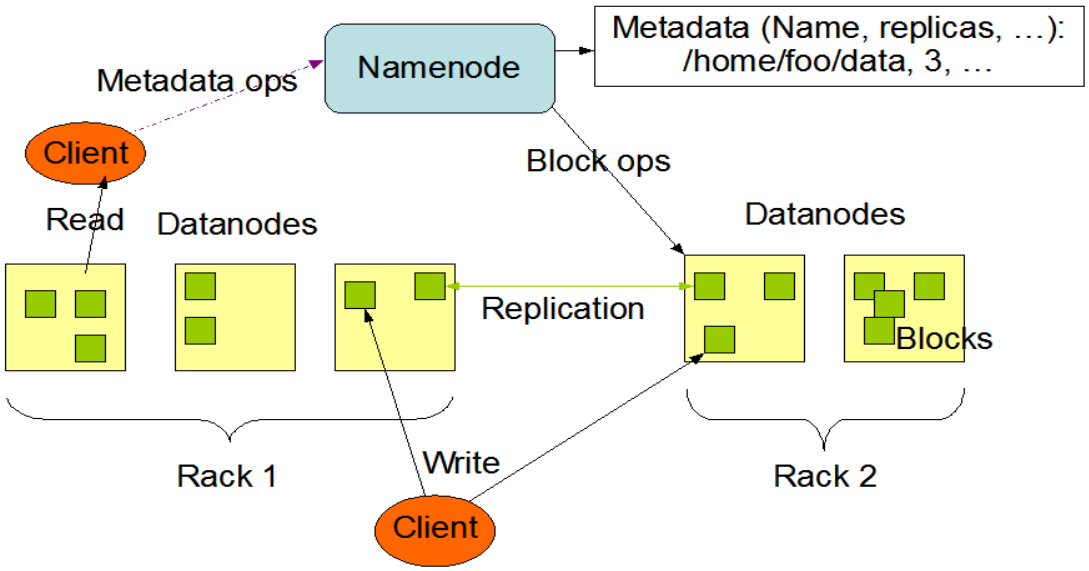
ساخت یک موتور جستجوی اینترنت ابتدا یک هدف جاه‌طلبانه ارزیابی می‌شد. انتقال حجم بالایی از صفحات وب برای عملیات کاووش یک چالش بزرگ محسوب می‌شد که توانایی مقیاس‌پذیری برای مواجهه با میلیون‌ها صفحه‌ی وب ذخیره شده را نداشت. با توسعه ابزار جدیدی به نام GFS امکان مواجهه با حجم بسیار زیادی از صفحات وب جمع‌آوری شده توسط ابزارهای جستجو فراهم گردید. این ابزار امکان مدیریت گره‌های ذخیره‌سازی را نیز داشت. همزمان توسعه ابزار متن‌باز جدید NDFS در

سال ۲۰۰۴ آغاز گردید و همان‌زمان، گوگل ابزار متن‌باز MapReduce را معرفی کرد. در سال ۲۰۰۸ ابزار هدوپ معرفی شد که به صورت متن‌باز امکان ذخیره‌سازی و اجرای برنامه‌های کاربردی روی تعداد زیادی از سخت‌افزارهای همگام را فراهم می‌نمود[۱۴][۳]. برخلاف پایگاه‌داده‌های رابطه‌ای، هدوپ توانایی کار تحلیلی روی حجم زیادی از داده‌ها را دارد[۱۴]. هدوپ درواقع یک چارچوب متن‌باز پردازش، ذخیره‌سازی و تحلیل حجم زیادی از داده‌های توزیع‌شده و بدون‌ساختار است. آغازکار هدوپ، به شرکت‌های بزرگ جستجوی اینترنتی بر می‌گرددکه دنبال ابزارها و مدل‌های پردازشی جدید برای استفاده در شاخص‌گذاری صفحات اینترنتی و جستجو روی آن‌ها بوده‌اند[۱۴]. مهمترین عامل گرایش سازمان‌ها در بکارگیری هدوپ، توانایی بالای آن در ذخیره‌سازی و پردازش حجم زیادی از انواع مختلف داده است.

اجزای اصلی سازنده‌ی هدوپ عبارتند از[۱۳]:

۱-Hadoop Common : این بسته کتابخانه‌ها و ابزارهای کاربردی برای راه‌اندازی سایر اجزای هدوپ است و به‌عنوان چارچوب اصلی ارائه‌دهنده‌ی خدمات و فرآیندهای اساسی مانند انتزاع از سیستم عامل و سیستم فایل عمل می‌کند.

۲-Hadoop Distributed File System(HDFS) : این ابزار حجم بسیار زیادی از داده‌ها در ماشین‌های مختلف که معمولا شامل صدها یا هزاران گره متصل به‌هم می‌باشد را ذخیره می‌کند. دو نسخه از آن‌ها در یک گروه و نسخه‌ی دیگر در گروهی دیگر قرار می‌گیرد. معماری HDFS شامل دسته‌بندی‌هایی است که از طریق یک ابزار نرم افزاری به‌نام NameNode که امکان مدیریت و رصد سیستم‌فایل‌های این دسته‌بندی‌ها و سازوکار دسترسی کاربران به آن‌ها را فراهم می‌کند و روی یک ماشین نصب می‌شود، در دسترس قرار می‌گیرد. روی سایر ماشینها یک نمونه از ابزارDataNode برای مدیریت ذخیره‌سازی دسته‌ها نصب می‌شود. این سیستم‌فایل دارای قابلیت‌هایی از قبیل ذخیره‌سازی توزیع شده، قابل حمل بودن، قابلیت اعتماد(حفظ و نگهداری از داده‌ها) می‌باشد.



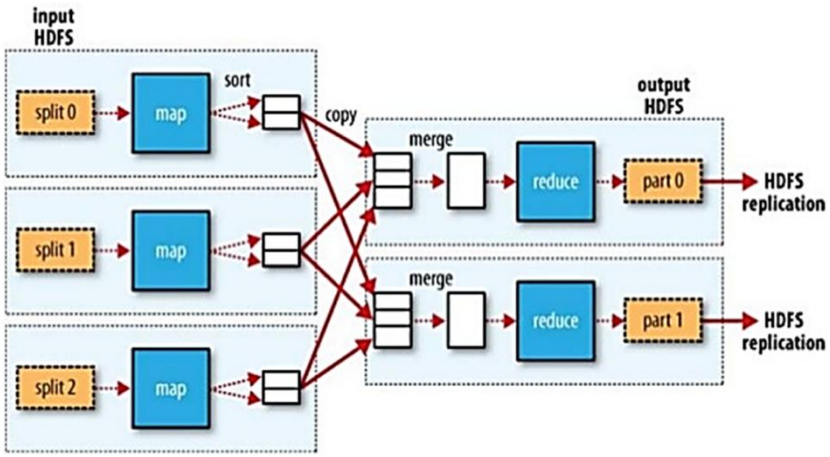
شکل(۲) : معماری سیستم فایل توزیع‌شده‌ی هدوپ

۳- MapReduce: در واقع یک مدل برنامه‌نویسی است که توسط شرکت گوگل برای پردازش و تولید مجموعه‌داده‌های بزرگ روی دسته‌هایی از کامپیوترها با استفاده از روش تقسیم‌وغلبه برای درهم شکستن مسائل داده‌های حجیم به بخشهای کاری کوچک و پردازش موازی آن‌ها معرفی شده است. در واقع MapReduce روی دسته‌های بزرگی از کامپیوترهای همگام اجرا می‌شود و بالقوه مقیاس‌پذیر است. چارچوب MapReduce از دو بخش اصلی تشکیل شده است :

- تابع Map : این تابع اجازه می‌دهد نقاط مختلف دسته‌ی توزیع، فعالیت خود را توزیع نمایند. دراین مرحله داده‌ی گره اصلی به تعدادی زیر مسئله‌ی کوچکتر خرد شده و به‌دنبال آن یک گره کارگر تعدادی زیر مجموعه از

مسئله‌های کوچکتر را زیر نظر مدیردرخواست پردازش کرده و نتایج را در سیستم‌فایل محلی ذخیره میکند، به‌نحوی که یک تابع کاهنده قادر به دسترسی به‌آن باشد.

- تابع Reduce : این تابع برای کاهش شکل نهایی نتایج دسته‌ها به‌یک خروجی طراحی شده و داده‌های ورودی از مراحل نگاشت را تحلیل و ادغام می‌کند. چندین وظیفه‌ی کاهش برای موازی‌سازی اجتماع می‌تواند همزمان اجرا شده و این وظایف روی گره‌های کارگر تحت نظر مدیردرخواست اجرا می‌شوند.



شکل(۳): معماری مدل MapReduce در هدوپ

۵.کلاودراایمپالا :

کلاودراایمپالا نرم‌افزار متن‌بازی است که شرکت کلاودرا برای پردازش موازی سنگین ارائه نموده است. ایمپالا فناوری پایگاه‌داده‌ی مقیاس‌وسیع را به هدوپ آورده و کاربران را قادر به ارائه‌ی درخواست و دریافت پاسخ در کمترین زمان ممکن می‌نماید و دسته‌ای از نرم‌افزارهای هدوپ را برای طیف گسترده‌ای از تحلیلگران پایگاه‌داده‌ها، کاربران و توسعه‌دهندگان در اختیار قرار می‌دهد. این پروژه در اکتبر ۲۰۱۲ به‌صورت عمومی معرفی شد. این محصول حاوی یک موتور SQL برای ذخیره‌سازی داده‌ها درکلاسترهای رایانه‌ایی مبتنی بر هدوپ است. موتور پردازش‌های موازی ایمپالا، درخواست‌های SQL هدوپ را به‌اندازه‌ی کافی برای تحلیل‌گرانی که با SQL آشنایی دارند و کاربران ابزارهای هوش کسب‌وکار، آسان نموده و برای انجام اکتشافات تعاملی و آزمایشات مختلف به-اندازه‌ی کافی سریع است. این درخواست‌ها، در قالب SQL روی داده‌های مستقر شده در HDFS و HBase اجرا می‌شود. در هنگام درخواست، داده‌ها روی HDFS جابجا یا منتقل نمی‌شوند. ایمپالا با هدوپ تجمیع شده تا از فایل‌ها، ابرداده‌ها، حفاظت و مدیریت منابع به‌صورت مشترک بهره برد[۱۵].

ایمپالا همچنین امکان کار با چارچوب‌های MapReduce، Hive و Pig را فراهم می‌کند و در واقع کلاودراایمپالا برای ترکیب اعتمادپذیری و مقیاس‌پذیری که از هدوپ مورد انتظار است با عملکرد و پشتیبانی از SQL که توسط پایگاه‌داده‌های با موتور پردازش موازی تجاری پیشنهاد شده طراحی گردیده است. در حال حاضر کلاودراایمپالا درخواست‌ها را با سرعت بین ۱۰ تا ۱۰۰ برابر سریعتر از راه‌حل‌های موجود هدوپ اجرا می‌کند که قابل مقایسه با پایگاه‌داده‌هایی با قابلیت پردازش موازی بوده و اجازه‌ی تحلیل‌های اکتشافی و تعاملی روی کلان‌داده‌ها را می‌دهد.[۱۶]. کلاودراایمپالا دارای یک لایه‌ی موثر ورودی، خروجی برای داده‌های ذخیره شده روی HDFS است که به‌ازای هر دیسک روی گره‌ها یک نخ ورودی و خروجی تولید می‌نماید[۱۲]. این ابزار جایگزین چارچوب‌های کاری روی MapReduce مانند Hive قرار نمی‌گیرد، بلکه درخواست‌های سریع و تعاملی به‌سبک SQL را روی داده‌های ذخیره‌شده‌ی HDFS فراهم می‌آورد. همچنین برای فراهم‌نمودن یک سکوی ذخیره‌سازی یکپارچه، ایمپالا از فراداده‌ها و دستورات مشابه SQL استفاده می‌کند.

پژوهش‌های مرتبط

سازمان‌های بزرگ به‌طور ویژه دنبال معماری و یا سیستم‌های مقیاس‌پذیری هستند که توانایی مواجهه با داده‌ها را داشته باشند. چالش‌های مدیریتی مرتبط با ماهیت انفجاری داده‌ها به‌ندرت در گذشته وجود داشته است. درحال حاضر این چالش‌ها با دسترس‌بودن جریان داده‌ها در سرعت بالا از منابع مختلف ایجاد شده است[۱۷]. تجزیه‌وتحلیل داده‌های کلان به بهبود رویکرد سازمان‌ها کمک شایسته‌ای می‌کند. اما لازم است داده‌ها از منابع جداگانه برای تجزیه‌وتحلیل و ارزیابی‌های مورد نیاز به پایگاه و مخزن اصلی آن به‌صورت مداوم و مستمر در جریان باشند. وجود یک دریاچه‌داده برای ذخیره‌سازی مقادیر زیاد داده‌ها به‌صورتی که همه‌ی آن‌ها در یک منبع اصلی گردآوری شوند و تجزیه‌وتحلیل نهایی روی آن‌ها صورت گیرد یک راه‌کار مناسب بدون نیاز به ساختار خاص امنیتی و صلاحیت‌سنجی می‌باشد[۱۸]. دریاچه‌داده در واقع یک انبار عظیم از داده‌های ذخیره‌سازی شده مورد نیاز در فرم و قالب اصلی خود است که با رویکرد انبارداده رابطه‌ای به‌گونه‌ای در تضاد است. به‌این صورت که در دریاچه‌داده به فرضیات و طراحی‌های بیشتری در مورد نحوه به‌کارگیری و استفاده از داده‌ها نیاز است در حالی که در انبارداده فرآیند پیش‌پردازش تعیین-شده قبل از بارگذاری داده‌ها روی آن انجام می‌گیرد. دراین صورت تلاش و وقت فراوانی را قبل از استفاده‌ی مفید از داده‌ها برای برنامه‌های کاربردی همراه دارد[۱۹]. دریاچه‌داده جایگزین و یا تکمیل‌کننده‌ی انبارداده نیست. در هسته‌ی دریاچه‌داده مجموعه‌ای از مولفه‌ها و ابزارها مانند پایگاه‌داده‌های رابطه‌ای ، مراکز داده‌عملیاتی و خوشه‌های سیستم‌فایل توزیع‌شده وجود دارند[۲۰]. در سال ۲۰۰۳ شرکت Google معماری سیستم فایل توزیع‌شده‌ی خود را با نام GFS منتشر کرد. این معماری مسئله‌ی کمبود فضای فایل‌های حجیم تولیدشده توسط موتورهای جستجو را مدیریت و برطرف می‌نمود. در سال ۲۰۰۴ آن‌ها تصمیم گرفتند یک نسخه‌ی پیاده‌سازی شده‌ی متن‌باز از آن معماری را ایجاد نموده و آن‌را سیستم‌فایل توزیع‌شده Nutch یا به اصطلاح NDFS نامیدند. در سال ۲۰۰۴ شرکت Google مقاله‌ای را با عنوان MapReduce ارائه نمود. خیلی زود در سال ۲۰۰۵ توسعه-دهندگان Nutch شروع به پیاده‌سازی یک نسخه از آن نمودند و طولی نکشید در اواسط همان سال تمامی الگوریتم‌های Nutch برای استفاده از MapReduce و NDFS تغییرساختاردادند. مدل برنامه‌نویسی نگاشت‌کاهش برای پردازش مجموعه‌های عظیمی از داده‌های رایانه‌ها(گره‌ها) که روی موضوعی خاص فعالیت می‌کنند طراحی شده است. این مجموعه روی‌هم رفته درصورتی‌که از سخت‌افزار یکسان بهره برند به‌عنوان خوشه شناخته می‌شوند. پردازش محاسباتی روی داده‌های ذخیره شده درون سامانه‌ی فایل بدون ساختار یا بر روی پایگاه‌داده‌ی ساختاریافته قابل اجرا است.این مدل در واقع شامل دو گام اساسی است:

- گام نگاشت: گره اصلی ورودی را به قطعات کوچک‌تر تقسیم می‌کند. دراین‌حالت یک مسئله‌ی بزرگ به‌چند مسئله‌ی کوچک تقسیم می‌شود. پس از آن این مسائل کوچک بین گره‌های کارگرتوزیع می‌شوند. یک گره کارگر نیز ممکن است این عملیات را به‌نوبه‌ی خود تکرار نماید، که ایجادکننده‌ی ساختار درختی و چند مرحله‌ای است. هرگره کارگر زیرمسئله‌ی خود را حل نموده و نتیجه را به گره اصلی خود برمی‌گرداند.
- گام کاهش: در مرحله بعد گره اصلی جواب زیرمسائل را از گره‌های کارگر گرفته و خروجی را می‌سازد تا خروجی، که حل مسئله ورودی است، را ایجاد نماید.

در اوایل ۲۰۰۸ هدوپ به‌عنوان سریع‌ترین سیستم مرتب‌سازی یک ترابایت داده رکورددار شد. با استفاده از یک کلاستر ۹۱۰ تایی هدوپ یک ترابایت داده را در ۲۰۹ ثانیه مرتب سازی نمود. در اواخر همان سال Google ادعا کرد می‌تواند همان حجم داده را در ۶۸ ثانیه مرتب سازی نماید. در اواسط ۲۰۰۹ اعلان شد تیمی از Yahoo! توانسته همان حجم داده را در ۶۲ ثانیه مرتب سازی نماید.

سامانه‌ی ثبت داده‌های محصلین آموزش‌وپرورش

سامانه‌ی ثبت داده‌های محصلین آموزش‌وپرورش به‌منظور ثبت و نگهداری داده‌های تحصیلی دانش‌آموزان طراحی و پیاده‌سازی شده است. این سامانه در سال ۸۹ برای استفاده در کلیه‌ی مراکز آموزشی کشور راه‌اندازی و مورد بهره-برداری قرارگرفت. در طراحی این سامانه از پایگاه‌داده‌ی رابطه‌ای SQLSERVER۲۰۰۵ استفاده شده و کلیه‌ی داده‌های مرتبط با وضعیت تحصیلی

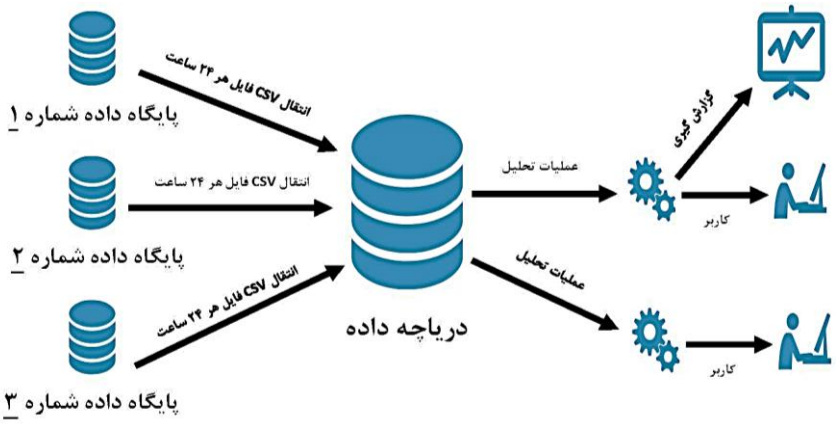
افراد برای مدت زمان نامحدود در این پایگاه‌داده ذخیره‌سازی و مورد استفاده قرار می‌گیرد. اطلاعات فردی، مشخصات والدین، نمرات، کارنامه‌های سالیانه، وضعیت انضباطی و فارغ‌التحصیلی مقاطع تحصیلی مختلف، داده‌های مربوط به محل سکونت و تماس، مشخصات فردی دبیران و آموزشگاه و... در این پایگاه‌داده ذخیره می‌شوند و در تمامی سال‌ها برای افراد قابل ارائه و استناد می‌باشند. در سامانه‌ی ثبت اطلاعات تحصیلی دانش‌آموزان از پایگاه‌داده‌ی رابطه‌ای SQLSERVER۲۰۰۵ استفاده شده که به‌منظور کاهش سربار و افزونگی می‌بایست نرمال‌سازی، اصول مربوط به تراکنش‌ها، سازگاری داده‌ها، استقلال فرآیندها و جامعیت پایگاه‌داده را رعایت کرد. بااین‌حساب در پایگاه‌داده‌ی رابطه‌ای فوق ۶۱ جدول مورد نیاز است تا اطلاعات مربوط به وضعیت تحصیلی دانش‌آموزان در ادوار مختلف را ذخیره‌سازی نماید. این تعداد جدول نرمال‌سازی شده فضای ذخیره‌سازی را افزایش می‌دهند و از طرفی لزوم وجود ارتباط زیاد بین آن‌ها برای استخراج اطلاعات کامل مورد درخواست از پایگاه‌داده سبب کاهش سرعت گزارش‌گیری و طولانی شدن انجام درخواست‌ها روی این نوع پایگاه‌داده‌ها می‌شود. از طرفی به‌علت محلی بودن پایگاه‌داده‌های فوق ارائه‌ی گزارش‌های جامع و کامل به‌منظور تحلیل دقیق‌تر از وضعیت موجود و اتخاذ رویکردهای درست و منطقی کار دشواری است.

کلیات رویکرد پیشنهادی

با توسعه‌ی ابزارهایی مانند هدوپ که شرایط مطلوب برای افزایش بهره‌وری، سرعت و کارآیی عملیات روی داده‌ها را فراهم نموده و نیز لزوم ارائه‌ی گزارش‌های تحلیلی جامع وکامل در زمان مطلوب برای آموزش‌وپرورش، در ابتدا می‌توان روی یک سخت‌افزار مناسب هدوپ و سرویس‌های لازم آن‌را فعال کرده و کلاودراایمپالا را برای ایجاد جداول و پایگاه‌داده‌ی دلخواه وگردآوری داده‌ها و اجرای درخواست‌ها فعال نمود. با توجه به قابلیت‌های ایمپالا که امکان بارگذاری از پایگاه‌داده‌ی رابطه‌ای را فراهم می‌کند، می‌توان شمای لازم از پایگاه‌داده را متناسب با پایگاه‌داده‌ی رابطه‌ای ایجاد نمود و سپس بارگذاری آن‌ها را از پایگاه‌داده‌ی رابطه‌ای به پایگاه‌داده‌ی جدید انجام داد. سپس انجام درخواست‌ها و تحلیل‌ها در زمان مطلوب‌تری انجام می‌شود. لذا با توجه به استفاده از ساختارفایل هدوپ برای ذخیره‌سازی داده‌ها به هیچ سخت‌افزار اضافی برای ذخیره‌سازی و یا عملیات پیش‌پردازش نیاز نمی‌باشد.دراین رویکرد به سخت‌افزار جدید و قدرتمند نیازی نبوده و در هرگام امکان اجرای درخواست‌های جامع و کامل بر مبنای داده‌های ذخیره شده در زمان مطلوب‌تر نسبت به پایگاه داده‌های رابطه‌ای فراهم است.

از آنجا که درگام اول داده‌ها و جداول لازم برای انجام درخواست‌های تعیین شده در یک پایگاه‌داده‌ی رابطه‌ای ذخیره شده‌اند، لذا باید تمامی این داده‌ها یا بخشی از آن‌ها که مورد نیاز تحلیل‌گران و کاربران نهایی است به دریاچه‌داده در یک پایگاه‌داده‌ی غیررابطه‌ای منتقل و درخواست‌ها روی آن‌ها اجرا شود. دراین‌صورت اجرای درخواست‌ها روی داده‌های موجود در دریاچه‌داده‌ی پایگاه‌داده‌ی غیررابطه‌ای به‌مراتب سریعتر از پایگاه‌داده‌ی رابطه‌ای خواهد بود.

معماری ارائه شده در قالب کلی به‌صورت زیر است :



شکل(۴): معماری پیشنهادی

ابزارهای پیاده‌سازی

در این معماری پیکربندی و ایجاد دریاچه‌داده برای ذخیره‌سازی داده‌های مورد نیاز درنظر گرفته شده است. در اولین قدم باید پایگاه‌داده‌ی منبع که حاوی داده‌های خام هستند، برای استخراج و انتقال داده‌ها به دریاچه‌داده مورد تجزیه‌وتحلیل قرار گیرند. داده‌های پایگاه‌داده‌ی منبع با تبدیل به فرمت استاندارد CSV قابلیت بارگذاری و انتقال به پایگاه‌داده‌ی غیررابطه‌ای را دارند. در واقع تمامی این مراحل را به‌شکل ساده‌تر شامل عملیات استخراج، تغییرشکل و بارگذاری می‌توان در نظر گرفت. داده‌های اصلی ما در این تحقیق روی نرم‌افزار مدیریت پایگاه‌داده‌ی SQLSERVER قرار گرفته است و از سرویس-های این نرم افزار برای تغییر شکل داده‌ها به فرم استاندارد CSV استفاده می‌شود.

همانطورکه گفته شد پایگاه‌داده‌ی غیررابطه‌ای که به‌صورت توزیع‌شده راه‌اندازی شده است، باید سه ویژگی سازگاری، دردسترس بودن و تحمل تقسیم‌بندی را دارا باشد. برای مقیاس‌پذیری افقی باید تحمل تقسیم‌بندی آن قوی بوده و از ویژگی‌های سازگاری و دردرسترس بودن تنها یکی را انتخاب نمود. زیرا:

- اگر سیستمی تحمل تقسیم‌بندی را دارا نباشد دیگر آن را به‌عنوان یک سیستم توزیع‌شده و مقیاس‌پذیر افقی نمی‌توان درنظر گرفت.
- این ویژگی‌ها در تضاد با یکدیگرند و در یک‌زمان بیشتر از دو موردآن‌ها امکان کنار هم‌بودن را ندارند. لذا با توصیه‌ی متخصصان تحمل تقسیم‌بندی انتخاب قطعی بوده و انتخاب گزینه‌ی بعدی از بین سازگاری و دردسترس‌بودن کامل به ماهیت نرم‌افزار و اولویت‌های سازمانی بستگی دارد.

وضعیت انتخاب یکی از دو گزینه‌ی سازگاری و یا دردسترس‌بودن به‌شرح زیر است :

- انتخاب گزینه‌ی دردسترس‌بودن و تحمل تقسیم‌بندی(AP): انتخاب این دو گزینه به‌معنی این است که هر درخواستی حتما پاسخی دریافت خواهد کرد و این پاسخ تا حد ممکن شامل جدیدترین اطلاعات خواهد بود. اما شاید این اطلاعات قدیمی باشند. از طرفی شاید نوشتن اطلاعات جدید در لحظه روی گره‌ها ممکن نباشد. اما سیستم همچنان به فعالیت خود ادامه داده و در نهایت به ثبات اطلاعات خواهد رسید. درنهایت انتخاب این دو گزینه بیانگر این است برای سیستم دردسترس‌بودن و سرعت بیشتر آن اولویت بالاتری نسبت به ثبات و دوام اطلاعات دارد.
- انتخاب گزینه‌ی سازگاری و تحمل‌تقسیم‌بندی(CP): انتخاب این دو گزینه به‌معنی این است که هر درخواستی برای دریافت اطلاعات به‌طور قطعی و دقیق باید آخرین نسخه‌ی موجود را به-عنوان پاسخ ارسال نماید. از طرفی این آخرین نسخه‌ی موجود از هر اطلاعاتی روی سیستم، به‌طور کامل مشخص و واضح است. یعنی اگر دو درخواست یکسان از دو مکان مختلف به سیستم ارسال شود، سیستم جواب یکسان و مشخصی را به هر دو درخواست خواهد داد. نکته‌ی دیگر این است که افزایش سازگاری زمان رکود سیستم را افزایش می‌دهد. هر میزان که سازگاری سیستم افزایش داده شود سیستم دچار رکود شده و سرعت پاسخگویی کاهش می‌یابد.

از آن‌جا که برای ارائه‌ی تحلیل‌های دقیق و جامع از وضعیت آموزشی و تحصیلی، همواره باید آخرین و بروزترین اطلاعات و داده‌های موجود در دسترس باشند، لذا برای انتخاب پایگاه‌داده‌ی غیررابطه‌ای مناسب ویژگی‌های سازگاری و تحمل تقسیم‌بندی(CP) را می‌بایست در نظر داشت. از این‌رو درمیان پایگاه‌داده‌های غیررابطه‌ای که از این ویژگی برخوردار هستند با در نظرگرفتن رده‌بندی شکل زیر پایگاه داده‌ی‌غیررابطه‌ای HBase برای پیاده‌سازی انتخاب شد.

include secondary database models — 11 systems in ranking, September 2020

Rank			DBMS	Database Model	Score		
Sep 2020	Aug 2020	Sep 2019			Sep 2020	Aug 2020	Sep 2019
1.	1.	1.	Cassandra	Wide column	119.18	-0.66	-4.22
2.	2.	2.	HBase	Wide column	48.35	-0.76	-7.37
3.	3.	3.	Microsoft Azure Cosmos DB	Multi-model	31.67	+0.94	+0.80
4.	4.	4.	Datastax Enterprise	Wide column, Multi-model	8.51	-0.13	-0.12
5.	5.	5.	Microsoft Azure Table Storage	Wide column	5.77	+0.19	+1.61
6.	6.	6.	Accumulo	Wide column	4.11	+0.09	+0.07
7.	7.	7.	Google Cloud Bigtable	Wide column	3.67	+0.29	+1.82
8.	8.	8.	ScyllaDB	Multi-model	2.59	+0.10	+0.96
9.	9.	9.	MapR-DB	Multi-model	0.79	+0.05	+0.17
10.	10.	↑ 11.	Alibaba Cloud Table Store	Wide column	0.43	+0.05	+0.20
11.	11.		Elassandra	Wide column, Multi-model	0.39	+0.06	

شکل(۵): معماری پیشنهادی

نحوه‌ی پیاده‌سازی

برای ارزیابی معماری پیشنهادی پایگاه‌داده‌های آموزش‌وپرورش استفاده شد. این داده‌ها در پایگاه‌داده‌ی رابطه‌ای SQLSERVER ذخیره شده است. در پایگاه‌داده‌ی غیررابطه‌ای میان جداول رابطه وجود ندارد. لذا همه‌ی اطلاعات جداول مرتبط در پایگاه‌داده‌ی رابطه‌ای باید در یک جدول پایگاه‌داده‌ی غیررابطه‌ای ذخیره‌سازی شوند. از این‌رو جدول پایگاه‌داده‌ی غیررابطه‌ای دارای ستون‌های زیر خواهد بود :

جدول(۲): ستونهای سند پایگاه‌داده‌ی غیررابطه‌ای

نام فیلد	نام فیلد	طول فیلد
کد دانش آموز	StudentCode	۱۲
نام دانش آموز	FirstName	۱۰۰
نام خانوادگی دانش آموز	LastName	۱۰۰
کد ملی	NationalCode	۱۰

۱۰۰	FatherName	نام پدر
۱۰	SchoolCode	کد مدرسه
۱۲	TimeYear	سال تحصیلی
۵	CourseCode	کد درس
۱۰۰	CourseName	نام درس
۵	Val	نمره

این داده‌ها در پایگاه‌داده‌ی رابطه‌ای در جداول جداگانه‌ی نرمال ذخیره‌سازی شده‌اند. با الحاق جداول به‌شکل مناسب در محیط SQLSERVER اطلاعات موردنیاز از پایگاه‌داده استخراج شده و در قالب CSV ذخیره می‌شوند. پس از راه‌اندازی و پیکربندی پایگاه‌داده‌ی غیررابطه‌ای و ایجاد بانک اطلاعاتی و جداول مورد نیاز، داده‌های فوق به‌این پایگاه بارگذاری شده و درخواست‌ها روی آن‌ها اجرا شد. برای اجرای درخواست‌ها و مقایسه زمان اجرا در این تحقیق از داده‌های آزمایشی استفاده شد.

محیط اجرا

دراین تحقیق برای پیاده‌سازی و ارزیابی معماری پیشنهادی، پایگاه‌داده‌ی غیررابطه‌ای HBase استفاده شد که از سیستم فایل هدوپ(HDFS) برای ذخیره‌سازی داده‌ها استفاده می‌کند. برای شبیه‌سازی هدوپ و راه‌اندازی پایگاه‌داده‌ی HBase از محیط مجازی کلاودرا نسخه ۵.۵ استفاده شد.

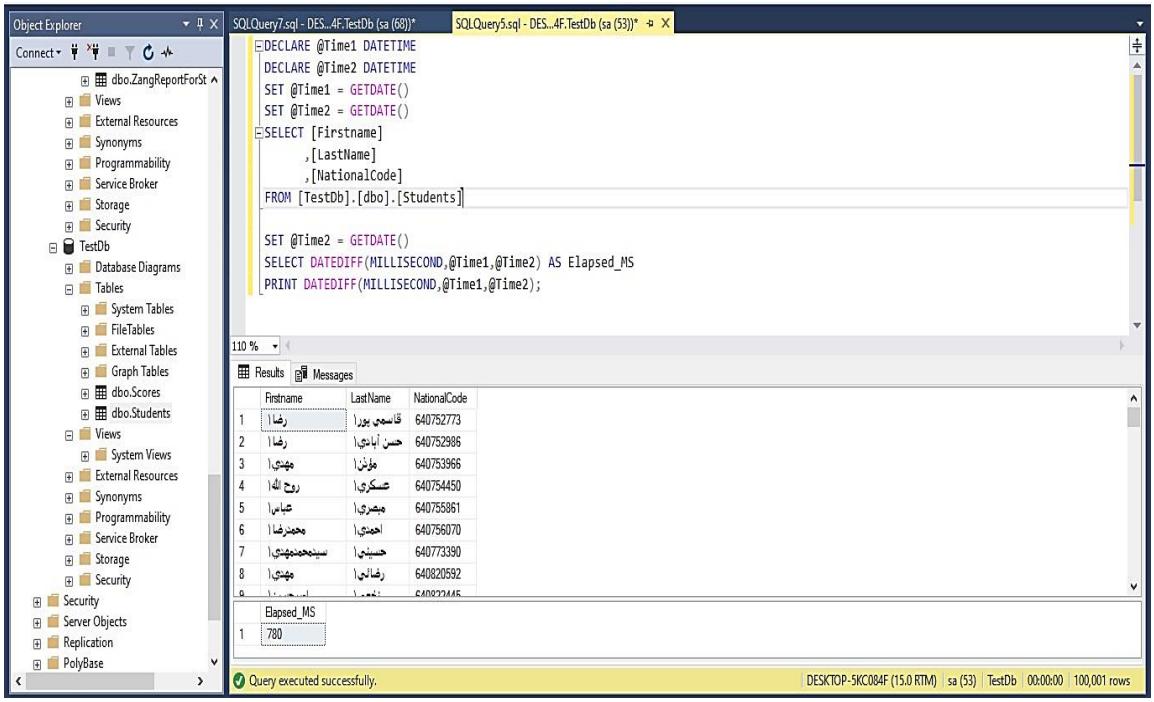
شکل(۶): ماشین مجازی کلاودرا

درخواست‌هایی که برای مقایسه‌ی زمان اجرا در پایگاه‌داده‌ی رابطه‌ای SQLSERVER و پایگاه‌داده‌ی غیررابطه‌ای HBase درنظر گرفته شده‌اند عبارتند از :

- محاسبه‌ی میانگین نمرات یک درس دانش‌آموزان به‌تفکیک سال تحصیلی
- محاسبه‌ی میانگین نمرات یک درس دانش‌آموزان به‌تفکیک آموزشگاه
- محاسبه‌ی میانگین نمرات یک درس دانش‌آموزان به‌تفکیک سال تحصیلی و آموزشگاه
- محاسبه‌ی میانگین نمرات یک درس دانش‌آموزان تاکنون
- محاسبه‌ی معدل دانش‌آموزان به‌تفکیک سال تحصیلی
- محاسبه‌ی معدل دانش‌آموزان به‌تفکیک آموزشگاه محل تحصیل

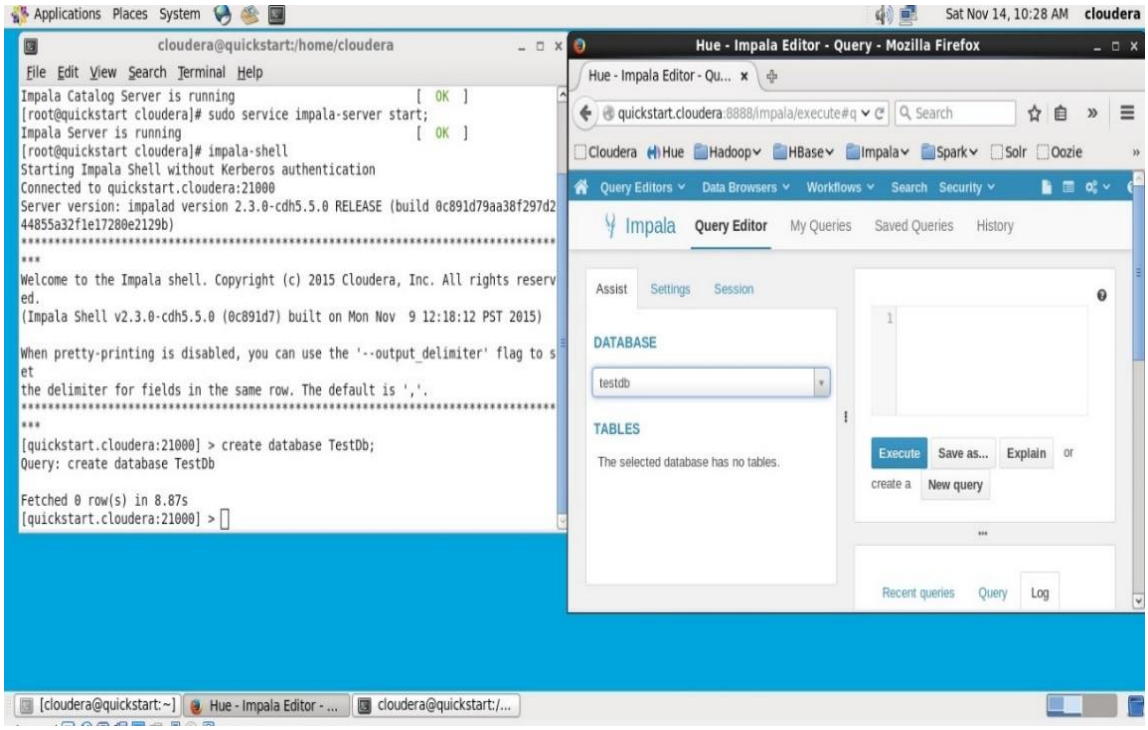
- محاسبه‌ی معدل دانش‌آموزان به‌تفکیک سال تحصیلی و آموزشگاه
- محاسبه‌ی معدل دانش‌آموزان تاکنون

داده‌های آزمایشی این تحقیق روی پایگاه‌داده‌ی رابطه‌ای SQLSERVER در جداول مجزا به‌شکل نرمال ذخیره‌سازی شده و حاوی یک‌صد هزار رکورد از اطلاعات دانش‌آموزان می‌باشد. در ابتدای راه‌اندازی ماشین مجازی، حافظه‌ی رم اختصاص یافته به‌آن دراین پژوهش ۴۰۹۶ مگابایت است. از طرفی برای ایجاد شرایط یکسان مقایسه، پایگاه‌داده‌ی رابطه‌ای SQLSERVER نیز روی یک ماشین مجازی با تنظیمات مشابه نصب و راه‌اندازی شده و داده‌ها به‌آن منتقل شدند.کد محاسبه‌ی زمان اجرای درخواست‌های پایگاه‌داده‌ی رابطه‌ای در شکل زیر نشان داده شده است :



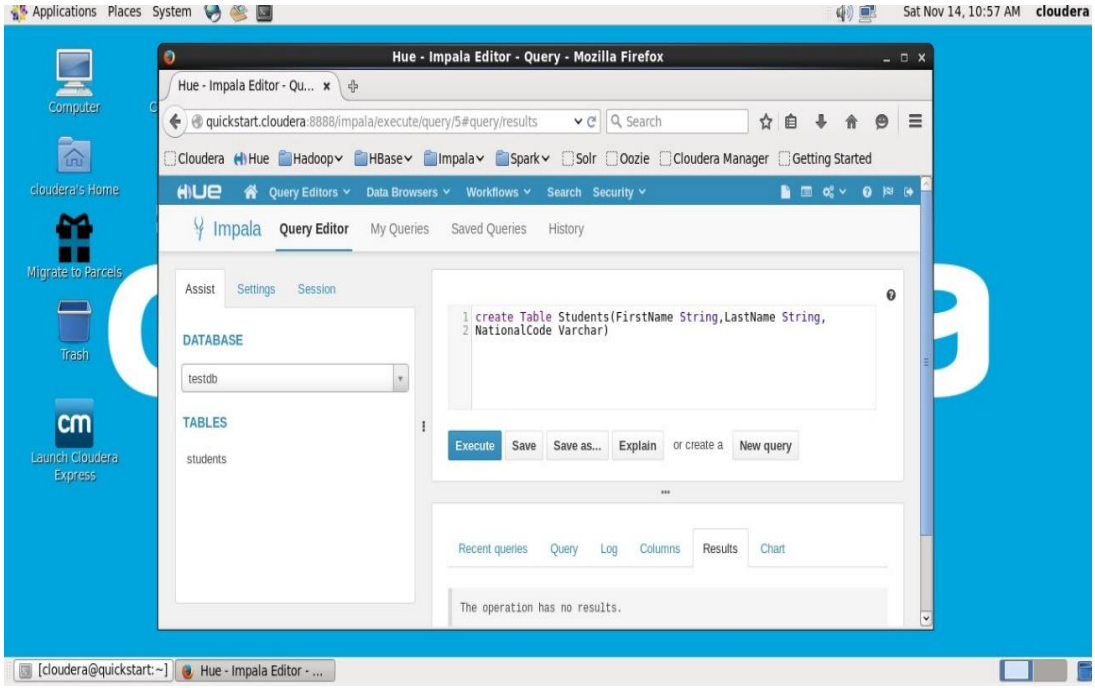
شکل(۷): اجرای درخواست‌ها در محیط پایگاه‌داده‌ی رابطه‌ای

راه‌اندازی سرویس‌های ایمپالا، پایگاه‌داده و جداول در پوسته‌ی آن یا مرورگر Hue به‌صورت زیر است:



شکل(۸): راه اندازی پایگاه‌داده در ایمپالا

اکنون می‌توان داده‌ها را از پایگاه‌داده‌ی رابطه‌ای در قالب فایل CSV استخراج و در جداول پایگاه-داده‌ی غیررابطه‌ای با اجرای دستور انتقال، بارگذاری نمود.



شکل(۹): ایجاد جدول در پایگاه‌داده‌ی ایمپالا

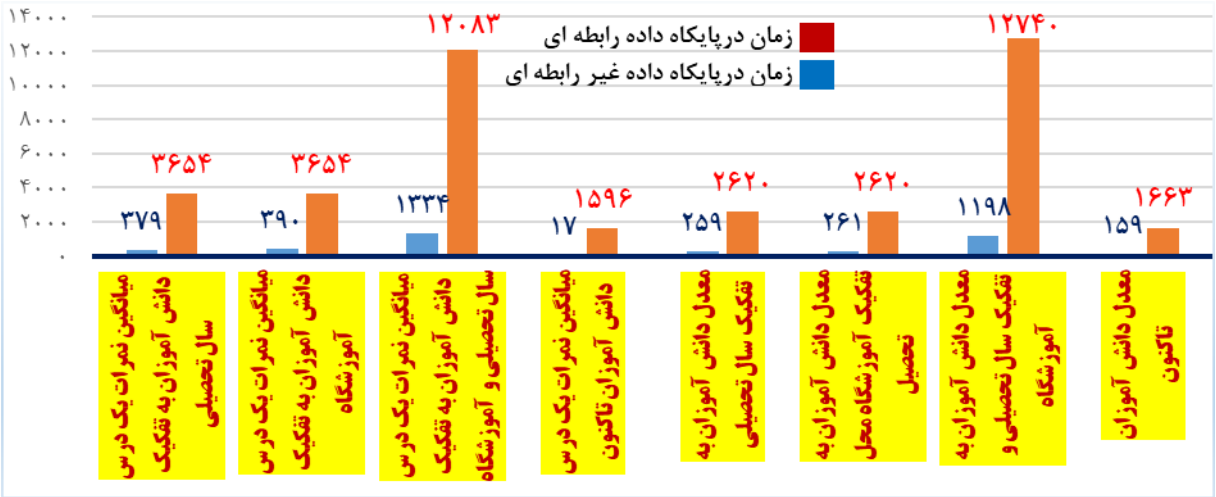
ارزیابی

برای ارزیابی معماری پیشنهادی، تعدادی از درخواست‌های پرکاربرد برای نمونه انتخاب شده و روی پایگاه‌داده‌ی رابطه‌ای اجرا شد و سپس همان درخواست‌ها روی پایگاه‌داده‌ی غیررابطه‌ای نیز اجرا گردید و زمان اجرای آنها با یکدیگر مقایسه شد.
زمان اجرای این درخواست‌ها در جدول زیر با یکدیگر مقایسه شده است :

جدول(۳): مقایسه‌ی زمان اجرا

عنوان گزارش	زمان (میلی‌ثانیه)	
	پایگاه‌داده‌ی غیررابطه‌ای	پایگاه‌داده‌ی رابطه‌ای
محاسبه‌ی میانگین نمرات یک درس دانش‌آموزان به‌تفکیک سال تحصیلی	۳۷۹	۳۶۵۴
محاسبه‌ی میانگین نمرات یک درس دانش‌آموزان به‌تفکیک آموزشگاه	۳۹۰	۳۶۵۴
محاسبه‌ی میانگین نمرات یک درس دانش‌آموزان به‌تفکیک سال تحصیلی و آموزشگاه	۱۳۳۴	۱۲۰۸۳
محاسبه‌ی میانگین نمرات یک درس دانش‌آموزان تاکنون	۱۷	۱۵۹۶
محاسبه‌ی معدل دانش‌آموزان به‌تفکیک سال تحصیلی	۲۵۹	۲۶۲۰
محاسبه‌ی معدل دانش‌آموزان به‌تفکیک آموزشگاه محل تحصیل	۲۶۱	۲۶۲۰
محاسبه‌ی معدل دانش‌آموزان به‌تفکیک سال تحصیلی و آموزشگاه	۱۱۹۸	۱۲۷۴۰
محاسبه‌ی معدل دانش آموزان تاکنون	۱۵۹	۱۶۶۳

نمودار مقایسه‌ی زمان اجرا در پایگاه‌داده‌ی رابطه‌ای و غیررابطه‌ای این تحقیق در زیر ارائه شده است :



نمودار(۱): نمودار مقایسه‌ی زمان درخواست‌ها در پایگاه‌داده‌ی رابطه‌ای و غیررابطه‌ای

نتیجه‌گیری

همانطور که از نتایج اجرای درخواست‌ها روی پایگاه‌داده‌ی رابطه‌ای و غیررابطه‌ای و مقایسه‌ی زمان اجرا در آن‌ها استنباط می‌شود، اجرای درخواست‌ها روی پایگاه‌داده‌ی غیررابطه‌ای سریعتر از پایگاه‌داده‌ی رابطه‌ای انجام می‌شود. همچنین به‌دلیل اجرای درخواست‌های همزمان روی گره‌های مختلف در فرآیند نگاشت-کاهش، زمان اجرا به‌نسبت یک سرور با حجم داده‌ی بالا

بسیار کمتر بوده و به‌دلیل ویژگی مقیاس‌پذیری پایگاه‌داده‌های غیررابطه‌ای نگرانی در زمینه‌ی افزایش حجم داده‌ها و کاهش سرعت اجرای درخواست‌ها آن‌گونه که در پایگاه‌داده‌های رابطه‌ای مدنظر است وجود نخواهد داشت. راه‌اندازی پایگاه‌داده‌ی غیررابطه‌ای به‌دلیل سرعت بالای اجرای درخواست‌ها و مقیاس‌پذیری که امکان ذخیره‌سازی حجم بیشتری از داده‌ها را فراهم می‌کند، درمقایسه با تجهیز و ارتقای سرورهای پایگاه-داده‌های رابطه‌ای به‌مراتب کاربردی‌تر و در بازه زمانی طولانی‌تر مقرون به‌صرفه‌تر خواهد بود. گرچه درخواست‌های اجرا شده در این پژوهش شامل داده‌های عددی و اجرای عملیات محاسباتی بود، لیکن در مورد داده‌های آماری نیز تا حدودی این‌جنبه از تفاوت قابل توجه است. همان‌گونه در فصل‌های قبل این پژوهش مشاهده گردید، پایگاه‌داده‌های غیررابطه‌ای برای پردازش حجم بالای داده‌ها و ذخیره‌سازی اقسام مختلف داده توسعه یافته‌اند. به‌کارگیری این پایگاه‌داده‌ها برای ایجاد منبع عظیم داده که از آن به‌عنوان دریاچه‌داده یاد می‌شود علاوه‌بر افزایش سرعت در پردازش و اجرای درخواست‌ها، به‌دلیل خاصیت مقیاس‌پذیری می‌توانند به‌نسبت پایگاه‌داده‌های رابطه‌ای حجم بیشتری از داده‌ها را در خود ذخیره‌سازی نمایند. در آموزش‌وپرورش حجم بالای داده‌ها و سرعت بالای تولید داده، نیاز به دسترسی به داده‌های بیشتر به‌منظور تحلیل جامع و برنامه‌ریزی های مدیریتی می‌توان از پایگاه‌داده‌ی غیررابطه‌ای برای ذخیره‌سازی حجم زیادی از داده‌ها و اجرای درخواست‌ها در زمان مطلوب‌تر استفاده نمود. وجود معماری دریاچه‌داده افق‌های تازه‌ای را در مورد مباحث مربوط به ذخیره‌سازی، امنیت و پردازش موازی، تکنیک‌های استخراج و بروزرسانی داده‌ها و همچنین تکنیک‌ها و ابزارهای داده‌کاوی نوین فراهم می‌نماید. از ایده‌های پژوهشی بیشتر در همین زمینه می‌توان به‌موارد زیر اشاره نمود :

- استفاده از پایگاه‌داده‌های غیررابطه‌ای مقیاس‌پذیر برای دسترسی و ذخیره‌سازی همه‌ی داده-های آموزش‌وپرورش و اجرای درخواست‌ها در زمان مطلوب‌تر
- به‌کارگیری هوش مصنوعی در فرآیند داده‌کاوی و تحیل و ارزیابی داده‌ها
- استفاده از ابزارهای پردازش ابری
- استفاده از ابزارها و سیتم‌های امنیت و کنترل جامعیت داده‌ها و حفظ پیوستگی و صحت داده-ها ترمیم و حذف داده‌های تخریب شده

منابع

[۱] A. Mohamed, M.K. Najafabadi, B.W. Yap, E.A. Kamaru-Zaman, and R. Maskat, “The state of the art and taxonomy of big data analytics: view from new big data framework”. AI Review, ۲۰۱۹, ۱-٤۹.
[۲] https://avora.com/blog/rise-of-the-data-warehouse/
[۳] S. Singh and N. Singh, "Big Data Analytics", ۲۰۱۲ International Conference on Communication, Information & Computing Technology Mumbai India, IEEE, October ۲۰۱۱
[٤] B.Gerhardt, K. Griffin and R. Klemann, "Unlocking Value in the Fragmented World of Big Data Analytics", Cisco Internet Business Solutions Group, June ۲۰۱۲,
[٥] C. Tankard, "Big Data Security", Network Security Newsletter, Elsevier, ISSN ۱۳٥۳-٤۸٥۸, July ۲۰۱۲
[٦] http://inside-bigdata.com/۲۰۱۳/۰۹/۱۲/beyond-volume-variety-velocityissue-big-data-veracity
[۷] H. Fang, "Managing data lakes in big data era: What's a data lake and why has it became popular in data management ecosystem," in ۲۰۱٥ IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), ۲۰۱٥: IEEE, pp. ۸۲۰-۸۲٤.
[۸] P. Kortvelyesi, "Top ۷ challenges of building a data lake," ed, ۲۰۱۷.
[۹] R. Hai, S. Geisler, and C. Quix, "Constance: An intelligent data lake system," in Proceedings of the ۲۰۱٦ International Conference on

Management of Data, ٢٠١٦: ACM, pp. ٢٠٩٧-٢١٠٠.
[١٠] IBM, "IBM Industry Model support for a data lake architecture," ٢٠١٦.
[١١] O. Mendelevitch, C. Stella, and D. Eadline, Practical Data Science with Hadoop and Spark: Designing and Building Effective Analytics at Scale. Addison-Wesley Professional, ٢٠١٦.
[١٢] https://www.grazitti.com/blog/data-lake-vs-data-warehouse-whichone-should-you-go-for/
[١٣] Ibrahim Abaker Targio Hashem, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, Samee Ullah Khan, The rise of "big data" on cloud computing: Review and open research issues, Information Systems, Volume ٤٧, January ٢٠١٥, Pages ٩٨-١١٥, ISSN ٠٣٠٦-٤٣٧٩
[١٤] C. Eaton, D. Deroos, T. Deutsch, G. Lapis and P.C Zikopoulos, Understanding Big Data Analytics for Enterprise Class Hadoop and Streaming Data, Mc Graw-Hill Companies ٩٧٨-٠-٠٧-١٧٩٠٥٣-٦,٢٠١٢
[١٥] S. Madden, "From Databases to Big Data", IEEE Internet Computing, June ٢٠١٢, v.١٦, pp.٤-٦
[١٦] Ishwarappa,Anurradha J,'A Brief Introduction on Big Data ٥Vs Characteristics and Hadoop Technology',Procedia Computer Science ٤٨(٢٠١٥)٣١٩-٣٢٤
[١٧] Ruoran Liu, Haruna Isah, Farhana Zulkernine, A Big Data Lake for Multilevel Streaming Analytics
[١٨] H. Fang, "Managing data lakes in big data era: What's a data lake and why has it became popular in data management ecosystem," in ٢٠١٥ IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), ٢٠١٥: IEEE, pp. ٨٢٠-٨٢٤.
[١٩] R. Hai, S. Geisler, and C. Quix, "Constance: An intelligent data lake system," in Proceedings of the ٢٠١٦ International Conference on Management of Data, ٢٠١٦: ACM, pp. ٢٠٩٧-٢١٠٠.
[٢٠] IBM, "IBM Industry Model support for a data lake architecture," ٢٠١٦