

بدست آوردن چارچوب تعیین منابع برای محاسبات ژنومیک مبتنی بر ابر

چکیده

در سال های اخیر محاسبات علمی با استفاده گسترده از محاسبات ابری به دلیل انعطاف پذیری آن در مدیریت منابع محاسباتی تکامل یافته است. در این مقاله، ما روی پردازش داده های ژنومی که به سرعت در حال پیشرفت در تحقیق و فعالیت های پزشکی است، تمرکز می کنیم. از ویژگی های اصلی این مجموعه داده، نه تنها تعداد فایل های ژنوم موجود در حال تبدیل شدن بسیار بزرگ است، بلکه هر یک از مجموعه داده ها به تنهایی به طور قابل توجهی بزرگ بوده، و بیش از ده گیگابایت می باشد. از این رو، پردازش بخش بزرگی از داده های ژنومی مبتنی بر ابر، تاثیر قابل توجهی در منابع شبکه دارند، زیرا هر پردازش نیاز به انتقال دهها گیگابایت به گره های محاسبه دارد. برای بررسی این موضوع، در این مقاله ما یک چارچوب تعیین منابع پیشنهاد می کنیم که عوامل تصمیم گیری با اطلاعات مورد نیاز برای انتخاب مناسب ترین گره های محاسبه را فراهم می کند. ما تابع تعیین منابع را به صورت توزیع شده اجرا کرده، و به طور گسترده آن را در یک بستر آزمایشی آزمایشگاه متشکل از حدود 70 گره تست کردیم. ما دریافتیم اورهد راه حل پیشنهادی در مقایسه با مقدار داده منتقل شده، ناچیز است.

کلمات کلیدی

محاسبات مبتنی بر ابر، پردازش ژنومیک، چارچوب توزیع شده، پروتکل های تعیین منابع، سبنگال دهی off-path.

NSIS

1. مقدمه

در سال های اخیر، اجرای سیستم عامل محاسبات علمی با استفاده از محاسبات مبتنی بر خوشه محلی و کاربرد محاسبات شبکه ای توزیع یافته، تکامل یافته، و آخرین دستاورد، استفاده از زیرساخت های محاسبات ابری می باشد [13]. این تکامل است به دلیل انعطاف پذیری بهبود یافته در مدیریت منابع محاسباتی می باشد که این آخرین پارادایم را نسبت به سایر راهها ارائه می دهد. با این حال، محاسبات علمی با برنامه های کاربردی معمولی مبتنی بر ابر، اعم از میزبانی سرور چند رسانه ای و استقرار تجهیزات ذخیره سازی، بسیار متفاوت است. تفاوت اصلی شامل حجم اطلاعات مدیریت شده توسط برنامه های علمی و یا CPU مورد نیاز، است که نسبت به سایر کاربردها، بسیار بیشتر می باشد. علاوه بر این، برنامه های کاربردی محاسبات علمی بسیار ناهمگن هستند. برای مثال، آنها ممکن است از سیستم عامل های مورد استفاده برای ذخیره و پردازش خروجیهای آزمایشهای فیزیکی دارای انرژی بالا، که نیاز به امکانات تجربی به منظور جمع آوری مقدار زیادی از داده ها به موقع، یا شبیه سازی تغییرات آب و هوا دارد، استفاده کنند. این کار نیاز به عملکرد محاسباتی بالا، و یا پردازش مجموعه داده های ژنومی دارد، که ممکن است مقادیر بیشتری نسبت به منابع محاسباتی لازم برای اجرای تک، با تعداد بسیار بیشتر اجراها، نیاز داشته باشد که این مسئله سبب می شود که اندازه ورودی فایل ها به راحتی به دهها گیگابایت برسد.

در این مقاله، ما روی پردازش داده های ژنومی کار می کنیم. این موضوع با توجه به کاهش در هزینه های تعیین توالی DNA، به سرعت در فعالیت های پژوهشی و پزشکی در حال پیشرفت و گسترش است. [1] از ویژگی های اصلی این مجموعه داده، نه تنها تعداد فایل های ژنوم موجود در حال تبدیل شدن بسیار بزرگ است، بلکه هر یک از مجموعه داده ها به تنهایی به طور قابل توجهی بزرگ بوده، و بیش از ده گیگابایت می باشد. به این مساله تحت عنوان Big² اشاره شده است، و اهمیت آن در طول زمان افزایش می یابد. در واقع، انتظار می رود که در چند سال آینده تمام نوزادان تعیین توالی خواهند شد و علوم پزشکی، بر نتایج پردازش ژنوم پردازش استوار خواهد شد. واضح است که این مساله واقع بینانه نیست که فرض کنیم که هر بیمارستان قادر به دست آوردن یک مرکز محاسبات بزرگ (ابر خصوصی) برای رفع نیاز به تقاضای پردازش داخلی خواهد بود. بنابراین، استفاده از خدمات پردازش

عمومی مبتنی بر ابر راه حل واضح خواهد بود [11]. از این رو، بدیهی است که مدیریت مناسب داده های ژنومی نه تنها برای خدمات ذخیره سازی، بلکه همچنین برای پردازش آنها و انتقال آنها ضروری خواهد بود. در واقع، پردازش داده های ژنومی مبتنی بر ابر تاثیر قابل توجهی در منابع شبکه دارند، زیرا هر درخواست پردازش سبب انتقال گیگابایت ها از داده ها به گره های محاسبه در مراکز داده ی ارائه دهندگان ابر خواهد شد.

در این مقاله یک پروتکل تعیین منابع ابر [22] [29]، با ارائه عوامل تصمیم گیری با تعدادی از اطلاعات زمینه پیشنهاد خواهد شد. این اطلاعات ممکن است شامل موقعیت و در دسترس بودن منابع پردازش (RAM, CPU, ذخیره سازی)، داده های ورودی به پردازش شامل فایل های کمکی (به عنوان مثال حاشیه نویسی [12])، و فایل های تصویری از ماشین های مجازی (VMS) میزبانی بسته های نرم افزاری ژنومی مورد استفاده برای پردازش ژنومیک داده، باشد. در واقع، نوع پردازش ژنومی (اجرا از طریق ترکیبی از برنامه های مختلف، مربوط به خط لوله ژنومی) است که می تواند در حین یک ژنوم تک بسیار بزرگ اجرا شود. فقط به یک ایده، خواننده علاقه مند می تواند یک لیست غیر جامع گزارش شده در جدول 1، از [12] را پیدا کند. بنابراین، واقع بینانه نیست که انتظار داشته باشیم هر مرکز داده از یک ارائه دهنده ابر، توانایی میزبانی همه ماشین های مجازی مورد نیاز برای هر درخواست ممکن را به طور همزمان داشته باشد.

پروتکل تعیین ما از توابع ارائه شده توسط گام های بعدی در چارچوب انتشار سیگنال (NSIS) استفاده می کند [2] [3]. پس ما ابتدا با استفاده از توابع ارائه شده توسط گسترش *offpath* که به تازگی تعریف شده، مجموعه پروتکل های NSIS اجرا شده را ارائه [8] و در ادامه برای آن تعریف بهتری ارائه خواهیم داد [9]. این راه حل اجازه می دهد تا انتشار سیگنال را در مناطق شبکه ای به شکل دلخواهش نزدیک شود. با اعمال قابلیت رهگیری از لایه انتقال سیگنالینگ NSIS، این پروتکل انتشار سیگنالینگ، بسیار کارآمد خواهد شد و ممکن است اجازه ی پیدا کردن منابع نزدیک به یک مسیر شبکه داده را بدهد. در سناریوی در نظر گرفته، مسیر تحت بررسی می تواند، به عنوان مثال، یکی از اتصالات مخزن ذخیره سازی تصاویر VM مورد نیاز و مرکز داده های ذخیره سازی داده های ورودی باشد. از آنجایی که خوشه های محاسباتی در مراکز داده قادر به میزبانی ماشینهای مجازی هستند لذا نمی

توانند در مسیر IP اتصال دو سرور (فقط روتر در مسیر قرار دارد) قرار گیرند، پس از یک پروتکل سیگنالینگ با قابلیت تعیین مسیر برای تعیین مراکز داده با هر دو قابلیت محاسبات کافی و موقعیت مناسب به منظور به حداقل رساندن ترافیک شبکه استفاده می شود. هنگامی که این اطلاعات به مرکز تصمیم فرستاده می شود، می تواند یک الگوریتم بهینه سازی برای انتخاب مرکز داده را اجرا کند.

راه حل ما می تواند بعنوان راه تشخیص منابع حل نقطه به نقطه با قابلیت کوبله شدن با آدرس های شبکه مشخص بر پایه ان اس ای اس تقسیم بندی شود. همانگونه که در ۲۶ بیان شد روش های ملاسک نقطه به نقطه قابلیت ارائه راه حل های بر پایه نزدیکی ندارند و تنها سعی در محدود کردن اورهد شبکه دارند. روش های شاخه ای تشخیص مناسب منابع مربوط به یک سوپر نقطه را ممکن می کنند اما وقتی بیش از دو سوپر نقطه در میان باشند مشکل به این سادگی نخواهد بود. راه حل های دیگری که اخیراً ارائه شده اند از اونتولوژی مناسب برای تشخیص سرویس استفاده می کنند. بدون اینکه به مشکلات نزدیکی مسیر توجه کنند. در ۲۴ نویسندگان یک لایه خلاصه کننده را جهت تشخیص مناسب ترین منابع زیرساختی معرفی نمودند که بعداً در راه حل برپایه قیود که در فضای ابری چند منبعه استفاده می شود. با این وجود مشکلات نزدیکی به طرز دقیقی رسیدگی نمیشوند و تنها با ارضای نیازهای مکانی رسیدگی میشوند. راه حل های دیگری نیز به طور خاص برای پلتفرم های محاسبه شبکه ای مثل ۲۱ ۲۷ طراحی شده اند. در ۲۱ نیز تلاش به جهت به حداقل رساندن همپوشانی اطلاعات شبکه است و نه برای پیدا کردن گره های نزدیک به مکان ها یا مسیرهای خاص.

ما راه حل تعیین ارائه شده در یک بستر آزمایشی آزمایشگاه متشکل از حدود 70 گره را اجرا کردیم، و آزمایشات گسترده ای در این باره انجام دادیم. نتایج به دست آمده ثابت کرد که مقدار اورهد شبکه در راه حل پیشنهادی در مقایسه با اندازه فایل های داده رد و بدل شده ناچیز است.

مقاله بصورت زیر مرتب شده است. بخش دوم برخی از پیش زمینه های NSIS، و سناریوی مرجع ارائه شده توسط پروژه ARES [10] را نشان می دهد. بخش سوم اطلاعات الگوریتمی و پروتکل راه حل ما را فراهم می کند، و در

بخش چهارم نتایج آزمایشات آزمایشگاه ارائه شده است. در نهایت، در بخش پنجم ملاحظات و توضیحات ما ارائه شده است.

2. پیش زمینه و سناریوی مرجع

A. پیش زمینه پروتکل های NSIS و اجرای off-path

مجموعه پروتکل NSIS، توابع سیگنالینگ را به دو لایه تقسیم می کند [3]. لایه فوقانی، به نام پروتکل لایه سیگنالینگ NSIS (NSLP)، وظیفه ی پیاده سازی منطق برنامه سیگنالینگ را بر عهده دارد. لایه پایین تر، به نام پروتکل لایه انتقال NSIS (NTLP)، وظیفه ی ارائه پیام NSLP به گره NSIS بعدی در مسیر به سوی هدف مشخص را دارد. سیگنالینگ انتقال اینترنت عمومی (GIST) نسخه IETF تعریف شده ی [5] NTLP است. خدمات انتقال GIST تنها با استفاده از پروتکل های موجود در TCP / IP مقدر می باشد. GIST تنها پیام های NSLP نقطه به نقطه بین جفت همسایه های گره سیگنالینگ NTLP را می رساند، در حالی که توابع سیگنالینگ آنها به انتها، در صورت نیاز، توسط NSLP ارائه شده است.

قبل از شروع، GIST باید یک بخش پیام (MA) با استفاده از اطلاعات انتقال یافته در بسته GIST، مانند شناسه منحصر به فرد گره GIST (نظیر هویت) و آدرس های IP را ایجاد و انجام دهد.

اگر چه IETF تنها دو پروتکل NSLP را به صورت استاندارد ارائه کرده است (کیفیت سرویس سیگنالینگ و NAT یا فایروال پیمایش)، اما دیگر پروتکل ها مانند NetServ NSLP [7] نیز اجرا شده است.

NSIS در ابتدا برای مدیریت ناحیه های در طول گره و مبتنی بر مسیر های داده طراحی شده بود. برای این منظور، پیام های NTLP ممکن است در گره های دارای قابلیت NSIS در مسیر قطع شوند. به طور خاص، پروتکل GIST اجازه ی، تعیین مسیریابی پیام از طریق روش مسیریابی پیام (MRMs) را می دهد. دو MRM در حال حاضر در RFC GIST مشخص شده اند: (1) کوپله شدن آدرس MRM، که مسیرهای سیگنالینگ پیام از طریق مسیر داده

تعیین می کند، و (2) MRM دارای انتهای آزاد، برای پیش شرط گذاری مناطق در جعبه های میانه مقصدهای داده که در پشت سر آنها قرار دارد، مورد استفاده قرار میگیرد.

پیام GIST شامل یک پیام مسیریابی اطلاعات (MRI) است، که اجازه می دهد تا گره NSIS برای شناسایی MRM مورد استفاده قرار گیرد. برای مثال، در مورد یک MRM کوپله شده ی آدرس، بسته GIST توسط گره NSIS در مسیر قطع و پس از پردازش توسط NSLP دوباره به سمت مقصد ارسال خواهد شد.

معماری مدولار NSIS و GIST اجازه ی گسترش و همچنین تعریف MRMs جدید، را می دهد [4]. در [8] [9] ما قابلیت های سیگنالینگ NSIS با ویژگی های off-path با تعریف یک MRM جدید را گسترش داده ایم. این فرمت نیاز به تعریف یک پروتکل تعیین GIST شده برای ساخت هر گره GIST در همتای GIST دارد. با استفاده از اطلاعات جمع آوری شده توسط این پروتکل، ما یک استراتژی تحویل جدید پیام های سیگنالینگ، که از قابلیت های سیگنالینگ رهگیری GIST استفاده می کند، تعریف کردیم.

در این کار، ما از یک استراتژی تحویل off-path خاص تعریف شده در [8] [9]، به اصطلاح حالت شلنگ استفاده کردیم. این کار شامل ارائه پیام NSIS به تمام گره های NSIS که در یک فاصله هستند و در IP گره ای بیان شده است می شود، به شرطی که این فاصله، کمتر یا مساوی فاصله شعاعی گره از مسیر اتصال فرستنده داده با مقصد سیگنالینگ باشد. در عمل، هر گره NSIS در مسیر داده مسئول انتشار سیگنال در فاصله کمتر از شعاع گره (در حالت حباب) است. از این رو، شلنگ شامل دنباله ای از حبابهای مجاور است. همانطور که در الگوریتم مورد استفاده برای پیاده سازی توابع ارائه شده، در این مقاله ما از پروتکل تعیین GIST به نام برگ، استفاده می کنیم که ثبات و اورهد کم ایجاد می کند، و این مساله زمانی اتفاق می افتد که انتشار سیگنالینگ است با استراتژی جاری شدن GIST انجام گیرد. خواننده علاقه مند می توانید جزئیات بیشتری را در [9] پیدا کند.

B. سناریوی مرجع: پروژه ی ARES

سناریوی مرجع این مقاله پروژه تحقیقاتی ARES (تحقیقات شبکه پیشرفته ژنومی برای اتحادیه اروپا [10])، که هدف آن پیاده سازی امکانات پردازش مبتنی بر ابر در نقطه ی هدف (PoPs) از شبکه Géant [17] است. بنابراین، چارچوب ARES می تواند به عنوان یک شبکه تحویل (ADN, [14])، و شبکه Géant به عنوان ارائه دهنده ابری با مراکز داده های متعدد در نظر گرفته شود.

ما فرض می کنیم که هر پاپ دارای یک مقدار مشخصی از فضای ذخیره سازی و قابلیت های محاسباتی میزبانی شده در یک یا چند سرور بوده و مدیریت آن از طریق از OpenStack انجام می شود [16]. در این زیرساخت مجازی، ما همچنین فرض می کنیم که یک گره خاص از خوشه محاسبات در پاپ، یک VM را اجرا می کند که میزبان فضای NetServ [7] است. این راه حل ویژگی خوب پلاگین و اجرا دارد، بعلاوه اضافه کردن یک VM ساده و آسان بوده و نیاز به انجام هر گونه تغییر در مدیریت زیرساخت های محاسبات ندارد. علاوه بر این، اجازه ی استقرار معماری سرویس با سیستم های مختلف مدیریت ابرگونه را می دهد، و برای این کار تنها نیاز به انطباق با سرویس NetServ است.

NetServ یک راه حل برای مدولار سازی و مجازی سازی در خدمات شبکه است. معماری گره، برای هر نوع از گره ها، مانند روترها، سرورها، جعبه تنظیم، و موارد مورد نیاز کاربران، مناسب است. معماری آن، برای مجازی سازی شبکه های خدمات رسانی و یک محیط اجرای مشترک برای خدمات شبکه، هدف گذاری شده اند. در ARES، خدمات NetServ به عنوان انبارهای محتوا [15] استفاده می شود، که هدف آن محدود کردن اورهد انتقال فایل های مربوط به ژنوم های بزرگ، و پیاده سازی توابع مدیریت خدمات، که در تعامل با OpenStack هستند، است. همچنین خدمات ذخیره NetServ میتواند در روتر میزبانی شود (تلاش برای پورت NetServ به روترهای Jupiter در حال حاضر در حال پیشرفت است). برنامه و روش کنترل NetServ از سیگنالینگ NSIS برای هماهنگ کردن اجرای خدمات و مازول های خدمات در اطراف گره NetServ استفاده می کند.

3. تعیین چارچوب

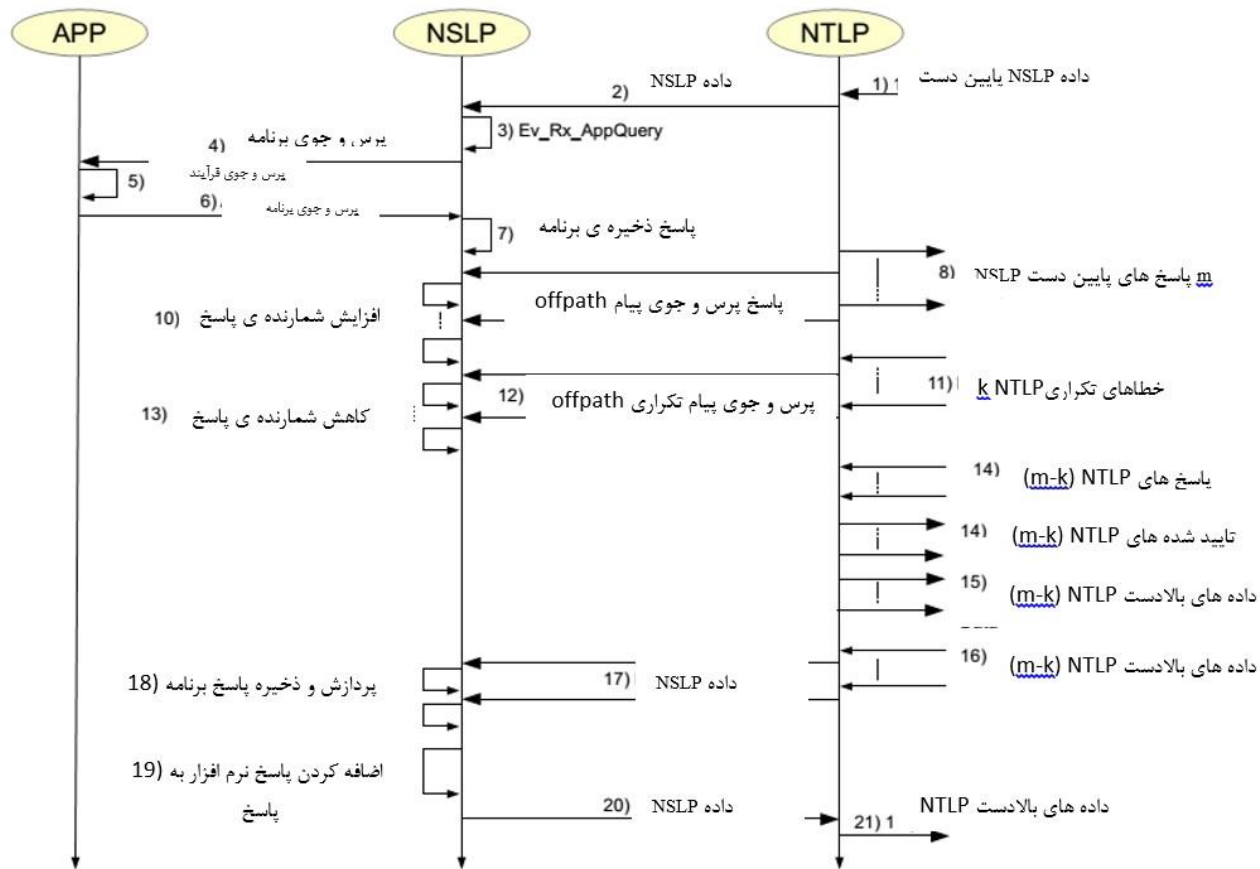
تعیین چارچوب ارائه شده مبتنی بر فرمت های off-path از مدل NSIS و به ویژه GIST می باشد. در جزئیات بیشتر، ما یک NSLP با حداقل یک نوع پیام، به نام پرس و جو، و پاسخ مربوطه (پاسخ پرس و جو) طراحی کرده ایم. و از آن برای جمع آوری اطلاعات باینری (درست یا غلط) در مورد وضعیت دریافت کنندگان استفاده می شود. البته در این شرایط تمام گره های فعال NSIS در شلنگ اطراف IP در مسیر بین فرستنده و مقصد می باشند. واضح است، این رفتار می تواند برای فعال کردن انواع پیچیده تر از پرس و جو استفاده شود. ساختار این پیام ها در طرح شکل 1 به صورت خلاصه وار ارائه شده است .



شکل 1. فرمت بسته های NSLP

از شکل 1، واضح است که این نوع از پرس و جو می تواند برای جستجوی هر دو گره ذخیره کننده یک محتوای مورد نظر (انبارهای یا مخازن اصلی) استفاده شود، و دیتاسنترها برای اجرای محاسبات مناسب هستند، در نتیجه برای مواد خاص رضایت بخش بوده و به حداقل موارد نیازمند هستند.

شکل 2 ارتباط بین فرآیند بین بخش های درگیر، مثل چارچوب یک گره فورواردر، زمانی که یک پیام پرس و جو را از طریق یک شلنگ GIST دریافت می کند، را نشان می دهد. بخش های درگیر (GIST) NTLP هستند که به تازگی تحت عنوان NSLP طراحی شده، و نرم افزار مدیریت داده ها هستند.



شکل 2. نمودار توصیفی توالی تعاملات بین نرم افزار، NSLP و NTLP درون یک گره

پس از دست دهی NTLP همانگونه که در [5] شرح داده شده، داده NSLP به GIST تحویل داده شده، که شرایط لازم برای ارسال آنها به NSLP را فراهم می کند. نوع پیام را شناسایی کرده، ظرفیت نرم افزار استخراج کرده و آن را به عنوان نمونه به نرم افزار (مراحل 1-4 در شکل 2) انتقال می دهد. نرم افزار پرس و جو را پردازش کرده و کد پاسخ را به NSLP برمی گرداند، و شرایط لازم برای ذخیره سازی آن را فراهم می کند (مراحل 5-7 در شکل 2). در همین حال، NTLP مسئول انتقال پیام پرس و جو از طریق شلنگ است. همچنین مقصدهای بعدی را شناسایی کرده و یک دست دهی می شود GIST با آنها آغاز می کند. هر دست دهی باعث تحویل یک وضعیت پیام از NTLP به NSLP [5] می شود. این پیام به اطلاع NSLP می رساند که پرس و جو به سمت یک مقصد جدید فرستاده شده و یک پیام پاسخ از مقصد آن انتظار می رود. به این معنا که، در NSLP می توانید یک

شمارنده بگذارید، که تعداد پیام پاسخ های نرم افزار به NSLP را قبل ارسال پاسخ به بالادست (مراحل 10/8 در شکل 2) ذخیره کند.

در طول دست دهی GIST، اگر قسمت بعدی همان پیام پرس و جو ی برنامه را از گره دیگر دریافت کرده باشد، آن به پیام پرس و جو ی GIST با خطا تکراری offpath پاسخ خواهد داد، که سبب از بین رفتن دست دهی می شود [9]. این خطا با استفاده از یکی دیگر از وضعیتهای پیام به NSLP اطلاع داده میشود، که باعث کاهش تعداد پاسخ های (مراحل 11-13 در شکل 2) گزارش شده می شود. اگر NTLP با تعداد m از دست دهی GIST شروع شود و تعداد k از خطای تکراری offpath دریافت کند، NSLP برای دریافت m-k پاسخ قبل از ارسال آن به بالادست صبر می کند (شکل 2، مراحل 1-16). هر بار که NSLP یک پاسخ از برنامه دریافت می کند، از تعداد شمار می کاهد (مراحل 17-19 در شکل 2). پس از دریافت آخرین پاسخ، عددشمار پاسخ به صفر کاهش می یابد (یعنی همه پاسخ های مورد انتظار دریافت شده است). NSLP پاسخ خود را به دسته ی پاسخ ها افزوده و به پاسخ بالادست ارسال می کند (شکل 2، مراحل 20-22).

در این مجموعه پاسخ، NSLP همچنین مسئول حفظ درست عمق موضوع هر پاسخ دریافتی است. این مساله در شکل 1 نشان داده شده است. این موضوع، همراه با گره-Id و کد پاسخ، اجازه می دهد تا آغازگر سیگنالینگ وظیفه ی محاسبه یک ساختار داده که شامل هر دو اطلاعات نرم افزار و توپولوژی شلنگ است، را انجام دهد. در طول مراحل 7 در شکل 2، NSLP مقدار دهی اولیه بسته بندی پاسخ را با داده های ارائه شده توسط کاربردهای محلی (مرحله 6 در شکل 2) را انجام داده، همچنین NSLP دارای گره-Id مختص خود بوده و یک عمق میدان حاوی مقدار 0 است. در مرحله 19 شکل 2، عمق میدان تمام بسته ی پاسخ های دریافتی، قبل از اضافه کردن آن را به بسته ی محلی، به اندازه ی 1 افزایش یافته است. شکل 3 نشان می دهد بسته ی موجود در هر پیام، پاسخ زمانی که یک پرس و جو از طریق یک شلنگ با شعاع 2 از R1 به R5 ارسال می شود، تغییر می کند. زمینه کد پاسخ برای خوانایی حذف شده است. این روشن است، که با استفاده از این قانون ساده بر روی درخت شکل 3، میتوان استنباط کرد که روتر R1 می تواند ساختار شلنگ با دندان صریح به تصویر کشیده باشد. در این شکل، برچسب

DC_i نشان دهنده ی مرکز داده (به این ترتیب یک گره خارج از مسیر)، در حالی که برچسب با R_j یک روتر را نشان می دهد.

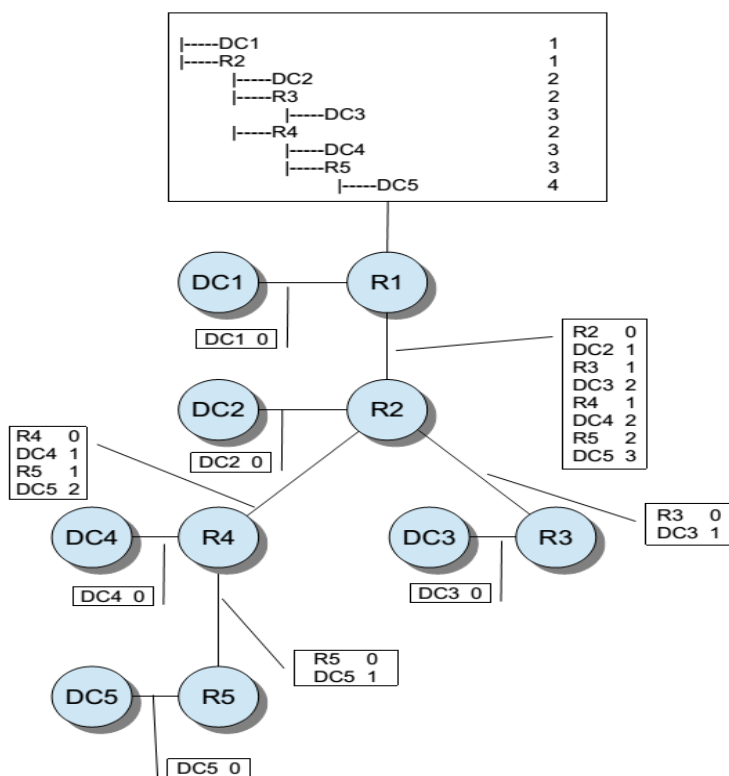
با اشاره به سناریوی پروژه ARES، شرح داده شده در بخش 2، مایک فرایند جستجو کاملاً جدید تعریف کردیم، که متشکل از سه مرحله بود. دو مرحله اول از سیگنالینگ شلنگ نشان داده شده در بالا استفاده میکرد.

در مرحله اول، عامل تصمیم گیری ARES، شناخته شده به عنوان GCM (مدیر CDN ژنومی [10])، باعث یک سیگنالینگ شلنگ، در ذخیره ی نرم افزار VM، به سمت گره ذخیره سازی داده های ژنومی مجموعه شد که دلیل آن انتقال آنها با هدف پردازش بود. علاوه بر این، سیگنالینگ های بیشتری نیز در محل ذخیره سازی فایل های کمکی انجام شد که برای محاسبه خط لوله [10] مورد نیاز بود. همچنین هر دو VM مجموعه ذخیره سازی و ژنومیک ذخیره سازی داده ها برای محاسبه خطوط لوله مورد نیاز بود. اولین مجموعه از سیگنالینگ شلنگ برای تعریف مراکز داده بالقوه نزدیک به هر یک از سه مسیر درگیر، و تامین حداقل الزامات مشخص شده در حمل بار پیام پرس و جو (در شکل 1 را ببینید میدان فهرست برنامه-نیاز) استفاده شد. هنگامی که برنامه ای که سیگنالینگ را ارسال کرده، شامل پیام پاسخ پرس و جو باشد، آن را به GCM انتقال داده که در نتیجه می تواند که تمام هویت دیتاسترها قادر به پشتیبانی از پردازش درخواستی را شناسایی کند.

گام بعدی، رها کردن از GCM به گره همان مرحله قبلاست، بعلاوه سیگنالینگ شلنگ برای شناسایی مکان های بالقوه (مخزن اصلی یا انبارهای) که در آن محتویات حافظه نهان (فایل های تصویری VM و یا فایل های کمکی) قرار دارند، باید ذخیره شود. در این مورد، محتوای جستجو در لیست محتوا-ID از پیام پرس و جو (به شکل 1. دوباره نگاه کنید) حفظ شده است. باز هم، زمانی که برنامه ای که سیگنالینگ حاوی پیام پاسخ پرس و جو را ارسال می کند، آن را به GCM انتقال داده، که در نتیجه می تواند تمام مکان های که در آن اطلاعات مورد نیاز برای محاسبه قرارداد را شناسایی کند.

در مرحله سوم، GCM ارتباط بین هویت گره ذخیره سازی داده ها به هر یک از مراکز کنترل داده (رابط محلی گسترش بسته نرم افزاری، LOIB، در معماری ARES، و به [10]مراجعه کنید) را ایجاد می کند، که فاصله از هر

مکان داده را اندازه گیری می کند (از نظر IP و یا زمان تاخیر شبکه). توجه داشته باشید که در بسیاری از موارد این اندازه گیری در گره موجود در لایه ی GIST است، چون در طول فاز اولیه کشف از پروتکل خود GIST جمع آوری شده است (به بخش III.A از [9] برای جزئیات بیشتر مراجعه کنید). بنابراین، بسته نرم افزاری (LOIB) به سادگی استفاده از GIST API به صورت محلی برای بازیابی این اطلاعات استفاده می کند. پس از آن، LOIB، برای هر نوع از داده ها (مجموعه داده های ژنومیکس، فایل های کمکی، فایل های تصویری VM) محل در کمترین فاصله را انتخاب کرده، و آن را به GCM برمی گرداند. وقتی GCM تمام پاسخها را بازیابی کرد، پس حاوی یک عکس کاملاً کامل از شبکه، از جمله منابع محاسباتی، فاصله شبکه، و در دسترس بودن محتوا، خواهد بود و میتواند یک تابع بهینه سازی برای انتخاب مرکز داده برای اجرای پردازش مورد نیاز اجرا کند



شکل 3 - توپولوژی بستر آزمایشی. گره های خاکستری یک مرکز داده واقع با PoP در 1 هاپ IP را نشان می دهد. هر پاپ با یک روتر شناسایی شده و، NSIS فعال است.

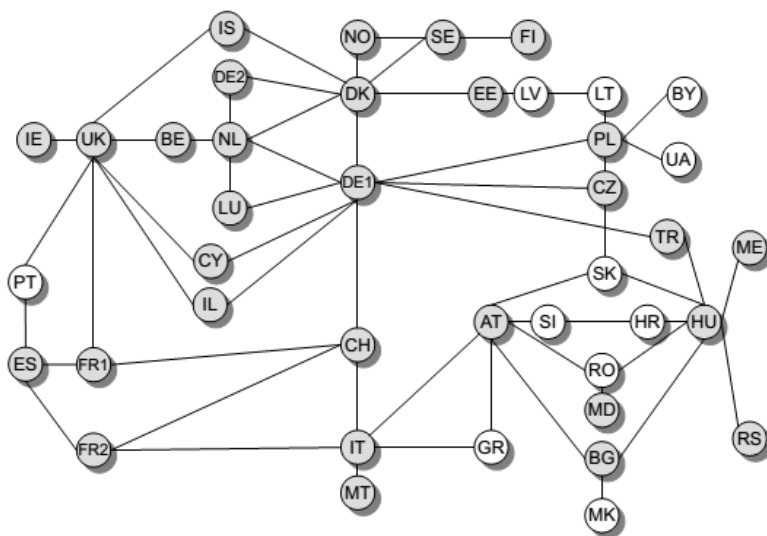
4. تخمین عملکرد

در این بخش، ما آورده یک پرس و جوی تک شلنگه، ، اندازه گیری شده و دارای توپولوژی 71 گره واقعی را ارائه می دهیم. ما از 71 ماشین مجازی در حال اجرای سیستم عامل اوونتو 12.04 استفاده کردیم.

شکل. 4 توپولوژی بستر آزمایشی را نشان می دهد. شبکه نشان دهنده ی سطح PoP شبکه Géant [17] می باشد. شبکه شامل 41 PoP می باشد، که شامل یک روتر واحد، و 30 مرکز داده تشکیل شده است. هر گره خاکستری در شکل به یک مرکز داده واحد (در شکل برای بهبود خوانایی نشان داده نشده) متصل بوده، و یک IP هاپ از آن خارج می شود. هر گره یک NSIS-KA [6] اجرا می کند، که همراه با offpath و NetServ است. ما از پیام NetServ NSLP پروب یا پروب پاسخ برای ارزیابی آورده سیگنالینگ پرس و جو یا پاسخ پرس و جو ، استفاده می کنیم. در واقع، آنها از همین منطق (نگاه کنید به [7]) استفاده می کنند. در واقع، پیام NetServ پروب از گره NetServ برای استفاده در اطلاعات وابسته استفاده کرده و پاسخ پروب مربوطه را به محل مورد نظر انتقال می دهد.

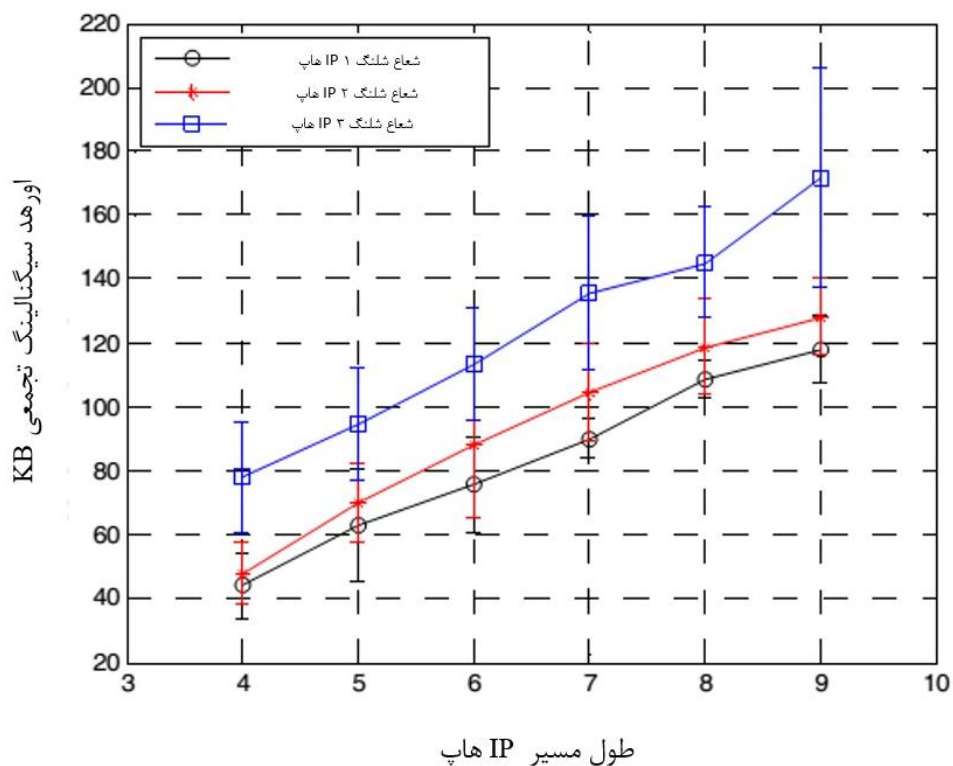
ما پیام پرس و جو را از یک گروه از منابع به مقصد با همان فاصله از نظر IP گره ارسال کردیم، و پس از آن ما را نتایج را با بازه ی اطمینان 95٪ میانگین گیری کردیم. پیام از طریق شلنگ های مختلف در حالی که شعاع در محدوده 1 تا 3 IP هاپ بود، روت شد. ما مسیرهای با طول 4 تا 9 هاپ IP را انتخاب کردیم. ما سربار شبکه های جمع آوری شده را اندازه گیری کرده، و توسط شمارش اندازه هر یک از پیام ها اندازه گیری (پروب یا پروب پاسخ و یا پیام خطا) کردیم که از هر لینک توپولوژی در لایه IP ، با استفاده از امکانات ورود به سیستم ارائه شده توسط iptables عبور کرد.

در این کار، ما از آورده ناشی از پیام های بی اساس GIST صرفنظر کردیم، چرا که تاثیر ناچیزی بر ظرفیت شبکه نداشت. این مساله قبلا نشان داده شده است [9].



شکل 4 - توپولوژی بستر آزمایشی. گره خاکستری نشان می دهد که مرکز داده با پاپ، در فاصله 1 IP هاپ قرارداد. هر پاپ با یک روتر در حال اجرا NSIS تحقق می یابد. هر مرکز داده توسط یک گره منفرد مدل شده، و در حال اجرای NSIS است.

شکل 5 نشان می دهد که مجموع اورهد تابعی از طول مسیر، با شعاع حباب شلنگ به عنوان یک پارامتر شکل استهمانطور که انتظار می رفت، اورهد با طول مسیر و شعاع حباب افزایش می دهد. نظر کلی این است که مجموع اورهد قطعا قابل اغماض است (کمتر از 200 KB در بدترین حالت، یعنی طول مسیر از 9 هاپ IP و شعاع شلنگ 3 IP هاپ). علاوه بر این، مقدار اورهد مطلق نیست بلکه در مقایسه با اندازه فایل نیاز به تغییر دارد. در واقع، همانطور که در گزارش [10] آورده شده، اندازه متوسط از یک ژنوم تنها حدود 3.2 GB است، در حالی که تصاویر VM فشرده حدود 3 گیگابایت می باشد. این به این معنی است که کل ترافیک به دلیل یک گام سیگنالینگ تک کمتر از 0.1% از اندازه از فایل کوچکتر منتقل شده، است. علاوه بر این، از آنجا که هر محاسبه برای چند ساعت [10] طول می کشد، تاثیر نرخ بیت سیگنالینگ مرتبط در عملکرد شبکه ناچیز می باشد.



شکل 5 - اورهد سیگنالینگ سر بار در شبکه به عنوان تابعی از طول مسیر، برای مقادیر مختلف از شعاع شیلنگ. لازم به ذکر است که با استفاده از اطلاعات مربوط به شبکه و وضعیت گره وضعیت، ارائه شده توسط سرویس تعیین پیشنهادی، میزان متوسط ترافیکی که به درون شبکه حرکت می کند را می توان با ضریب 6 کاهش داد. نتایج شبیه سازی [10] ارائه شده است. بنابراین، نه تنها اورهد سرویس با توجه به میزان ترافیک مربوط به خدمات پردازش ناچیز است، بلکه سود آن قطعا ارزش استقرارش را دارد.

5. نتیجه گیری

در این مقاله، ما یک چارچوب سیگنالینگ برای کشف منابع از منابع ابرها برای پردازش برنامه های کاربردی ژنومی ارائه دادیم. چارچوب پیشنهادی را می توان نه تنها برای جستجو مراکز داده قادر به میزبانی پردازش استفاده کرد، بلکه ذخیره نهان قادر به ارائه با اورهد کمتر است. ویژگی های خاص راه حل پیشنهادی، توانایی ارائه نتایج با درجه کنترل شده با توجه به مسیر داده مرتبط با بخشهای مربوطه است.

این جنبه بسیار مهم است که تصمیماتی گرفته شود که توانایی به حداقل رساندن تاثیر پردازش ژنومیک در شبکه های زیربنایی را داشته باشد. نتایج، از نظر اورهد شبکه، تایید می کنند که اورهد سیگنالینگ ناچیز و در نتیجه مقرون به صرفه برای هر سرعت شبکه ای است.

واضح است، که حتی اگر ما به صراحت به چارچوب ARES چارچوب اشاره کرده باشیم، این نوع از سیگنالینگ برای تمام سناریوهای شبکه که در آن مقادیر زیادی از داده ها به سمت سایت های ابری برای پردازش ثانویه منتقل می شوند، مناسب است. بنابراین، این راه حل حتی در خارج از سناریوهای ژنومی نیز قابل اجرا است. کارهای آینده باید شرح کامل سیستم و استقرار، از جمله تعریف یک تابع بهینه سازی مناسب در عامل تصمیم گیری و اندازه گیری، نه تنها از نظر اورهد سیگنالینگ سربار، بلکه شبکه مصرف منابع در طول انتقال اطلاعات، را در نظر بگیرد.

REFERENCES

- [1] Wetterstrand KA. DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP) Available at: www.genome.gov/sequencingcosts. Accessed on December 11, 2013.
- [2] R. Hancock, G. Karagiannis, J. Loughney and S. Van den Bosch, "Next Steps in Signaling (NSIS): Framework" IETF, RFC 4080, June 2005.
- [3] X. Fu et al., "NSIS: a new extensible IP signaling protocol suite", IEEE Communications Magazine, 43(10), 2005, pp. 133- 141.
- [4] J. Manner et al., "Using and Extending the NSIS Protocol Family", IETF, RFC 5978, October 2010.
- [5] H. Schulzrinne, R. Hancock, "GIST: General Internet Signalling Transport", IETF RFC 5971, October 2010.
- [6] NSIS-ka, open source NSIS implementation by Karlsruhe University, available at: <https://projekte.tm.uka.de/trac/NSIS/wiki/>.
- [7] M. Femminella, R. Francescangeli, G. Reali, J.W. Lee, H. Schulzrinne, "An Enabling Platform for Autonomic Management of the Future Internet", IEEE Network, Nov./Dic. 2011, pp. 24-32.
- [8] M. Femminella, R. Francescangeli, G. Reali, H. Schulzrinne, "Gossip based signaling dissemination extension for next steps in signaling," IEEE/IFIP Network Operations and Management Symposium (NOMS 2012), April 2012, Maui, HW, USA.
- [9] M. Femminella, R. Francescangeli, G. Reali, H. Schulzrinne, D. Valocchi, "Off-path Signaling Extension for General Internet Signaling Transport Protocol", submitted for journal publication. Available at <http://arxiv.org/abs/1406.7650>.
- [10] Mauro Femminella, Emilia Nunzi, Gianluca Reali, Dario Valocchi, "Networking issues related to delivering and processing genomic big data", International Journal of Parallel, Emergent and Distributed Systems, 2014, DOI: 10.1080/17445760.2014.929685
- [11] A. O'Driscoll a, J. Daugelaite, R. D. Sleator, "'Big data', Hadoop and cloud computing in genomics", Journal of Biomedical Informatics, vol. 46, 2013, pp. 774-781.
- [12] M. Yandell and D. Ence, "A beginner's guide to eukaryotic genome annotation", Nature Reviews Genetics, vol. 13, May 2012.

- [13] Eric E. Schadt, Michael D. Linderman, Jon Sorenson, Lawrence Lee, Garry P. Nolan, "Computational solutions to large-scale data management and analysis", *Nature Reviews Genetics*, vol. 11, September 2010.
- [14] P. Romano, F. Quaglia, "Design and Evaluation of a Parallel Invocation Protocol for Transactional Applications over the Web", *IEEE Transactions on Computers*, 63(2), 2014, pp. 317-334.
- [15] Femminella M., Reali G., D. Valocchi, Francescangeli R., Schulzrinne H, "Advanced caching for distributing sensor data through programmable nodes", *IEEE LANMAN 2013*, Bruxelles, Belgium, 2013.
- [16] OpenStack, <http://www.openstack.org/>, accessed on 30 April 2014.
- [17] The GÉANT pan-European research and education network. Available at <http://www.geant.net>
- [18] Satyen Abrol, Latifur Khan and Bhavani Thuraisingham, "An Ontology based System for Cloud Service", *Collaboratecom 2012*, October 14–17, 2012 Pittsburgh, United States.
- [19] Taekgyeong Han, Kwang Mong Sim, "An Ontology-enhanced Cloud Service Discovery System", *Proc. of International MultiConference of Engineers and Computer Scientists (IMECS 2010)*, March 17-19, 2010, Hong Kong.
- [20] Jaeyong Kang, Kwang Mong Sim, "Towards Agents and Ontology for Cloud Service Discovery", *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC 2011)*, October 2011, Beijing, China.
- [21] Rajiv Ranjan, Lipo Chan, Aaron Harwood, Rajkumar Buyya, Shanika Karunasekera, "Decentralised Resource Discovery Service for Large Scale Federated Grids", *E-SCIENCE '07*, Bangalore, India, December 2007.
- [22] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose, Rajkumar Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", *Softw. Pract. Exper.* 2011; 41:23–50.
- [23] Wu-Chun Chung, Chin-Jung Hsu, Kuan-Chou Lai, Kuan-Ching Li, Yeh-Ching Chung, "Direction-Aware Resource Discovery Service in Large-Scale Grid and Cloud Computing", *IEEE International Conference on Service-Oriented Computing and Applications (SOCA 2011)*, December 2011, Irvine, United States.
- [24] Wright et al., "A constraints-based resource discovery model for multi provider cloud environments", *Journal of Cloud Computing: Advances, systems and Applications 2012*, 1:6.
- [25] Ranjan R., Harwood A., Buyya R., "Peer-to-peer-based resource discovery in global grids: a tutorial", *IEEE Communications Surveys & Tutorials*, vol. 10, No. 2, 2008.
- [26] Gregor Pipan, "Use of the TRIPOD overlay network for resource discovery", *Future Generation Computer Systems*, Vol. 26, No. 8, October 2010, Pages 1257–1270.
- [27] Leyli Mohammad Khanlia, Saeed Kargarb, "FRDT: Footprint Resource Discovery Tree for grids", *Future Generation Computer Systems*, Vol. 27, No. 2, February 2011, Pages 148–156.
- [28] Guan Le, Ke Xu, Junde Song, "Gossip-based Hybrid Multi-attribute Overlay for Resource Discovery in Federated Clouds", *Ninth IEEE International Conference on e-Business Engineering (ICEBE 2012)*.
- [29] Alhamazani K., Mitra K., Lizhe Wang, Rabhi F., Ranjan R., "Cloud monitoring for optimizing the QoS of hosted applications", *IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom)*, 2012, pp. 765-770, 2012.