

تاثیر معماری سرویس‌گرا بر الگوریتم زمان‌بندی در محاسبات ابری

اعظم السادات نوربخش

هیات علمی گروه کامپیوتر دانشگاه آزاد اسلامی واحد لاهیجان
Nourbakhsh@liau.ac.ir

شاهد کاظمی شریعت پناهی

دانشجو کارشناسی ارشد مهندسی نرم افزار کامپیوتر دانشگاه آزاد واحد علوم تحقیقات گیلان
kazemi.shariatpanahi@gmail.com

ساسان برهلیا

دانشجو کارشناسی ارشد مهندسی نرم افزار کامپیوتر دانشگاه گیلان
Berehlia@msc.guilan.ac.ir

مسئول مکاتبات: شاهد کاظمی شریعت پناهی

چکیده

محاسبات ابری، طی سال‌های اخیر رو به محبوبیت گذارده و شرکتهای بزرگ فناوری اطلاعات سراسر دنیا شروع به طراحی زیرساخت آن نموده‌اند. در دنیای فناوری امروزی، برخی حوزه‌ها همانند SOA در محاسبات ابری نیز مطرح گردیده‌اند. مقاله حاضر، پژوهشی در مورد الگوریتم‌های زمان‌بندی محاسبات ابری با استفاده از SOA، مقوله‌هایی در زمینه پیاده‌سازی الگوریتمی جدید و ایده آل را مورد بررسی قرار داده و پیشنهاد می‌کند که این الگوریتم-ها با توجه به مقوله SOA با سایر گونه‌های دیگر آن به‌کار گرفته شود. افزون بر این، SOCCA طراحی سطح بالایی را به‌منظور حمایت بهتر از ویژگی‌های چندگانه عاریه‌ای از محاسبات ابری پیشنهاد می‌دهد. با این وجود، تفسیرهای گوناگونی از آن چه که محاسبات ابری می‌نامند، وجود دارد. حتی اگر برخی از ویژگی‌های اساسی محاسبات ابری از طریق تلاش‌های علمی و پژوهشی به مرحله تحقق رسانده شوند، این زمینه هنوز در ابتدای راه پیشرفت خود قرار دارد. همچنین همان گونه که گفته شد با در نظر گرفتن همپوشانی در میان SOA و محاسبات ابری در SOCCA می‌توان به آنتولوژی‌ها و پارامترهایی دست یافت که در آینده سبب بهبود در ساختار الگوریتم زمان‌بندی در محاسبات ابری شود و نوآوری جدیدی را پیش روی آورد. این مقاله به الگوریتم زمان‌بندی محاسبات ابری با توجه به‌تاثیر مقوله SOA پرداخته و روشی ایده‌آل را برای بهبود الگوریتم زمان‌بندی در محاسبات ابری بیان می‌کند.

واژگان کلیدی: معماری سرویس‌گرا، محاسبات ابری، الگوریتم زمان‌بندی، معماری لایه‌ای SOCCA

۱. مقدمه

می‌توان محاسبات ابری را به معنی به کارگیری قابلیت‌های کامپیوتری بر مبنای اینترنت در نظر گرفت. مزیت‌های محاسبات ابری در مقایسه با محاسبات سنتی شامل چابکی، کاهش هزینه، مستقل بودن از تجهیزات، عدم وابستگی به مکان و میزان پذیری است. معماری سرویس‌گرا، روشی برای طراحی و پیاده‌سازی نرم‌افزارهای گسترده سازمانی است تا بتوانند به وسیله ارتباط بین سرویس‌هایی که دارای خواص اتصال سست، دانه درشتی و قابل استفاده مجدد هستند، به هدف خود برسند. از این معماری، معمولاً برای ساخت سیستم‌های توزیع شده که کارکردهای نرم‌افزاری را در قالب سرویس ارائه می‌دهند، و نیز برای ساخت سرویس‌های جدید بهره جویی می‌شود. در این معماری، تلاش بر یکپارچه‌سازی فناوری‌ها در محیطی که انواع مختلف از سکوها نرم-افزاری و سخت‌افزاری وجود دارد، مدنظر است. با توجه به بررسی‌های گذشته در این حیطه، می‌توان محاسبات ابری و SOA را به طور مستقل و هم‌زمان مورد بررسی قرار داد، به‌گونه‌ای که محاسبات ابری و سرویس‌پیشنهادی بتوانند زیر بنای ارزشمندی برای دستاوردهای SOA را فراهم کنند. در ادامه این مقاله، به‌مراحل برای بهبود پیشرفت محاسبات ابری با کمک معماری سرویس‌گرا و دیگر موارد آن اشاره خواهد شد. با توجه به بحث‌های گفته شده، در این مقاله سعی شده است که نخست به معرفی و بررسی و سپس به ویژگی‌ها و لایه‌ها و پروتکل‌های استاندارد، که برای ترکیب با این دو مقوله سروکار دارند، پرداخته شود. در ادامه زمان بندی منابع ابرها در تئوری مصرف کننده و تهیه کننده مورد بررسی قرار گرفته‌اند. در این بخش نیازهای مشتری، منابع

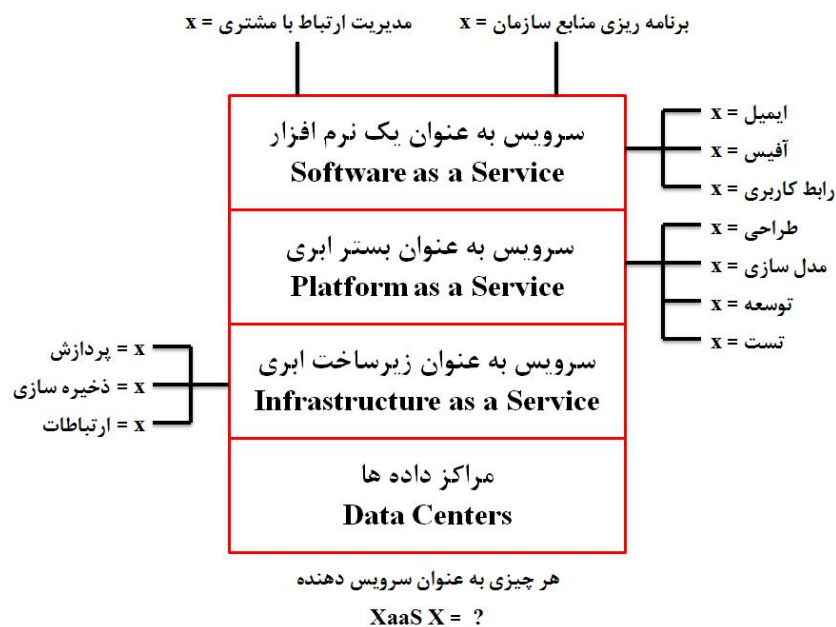
فروشنده‌ها، قیمت مورد توجه قرار گرفته و سپس هر کدام با توجه به مقوله SOA مدل شده‌اند. در بخشی دیگر الگوریتم مصرف انرژی که محاسبات ابری بیشتر به قسمت لایه واسطه‌ای در SOA مربوط می‌شود تحلیل شده و پس از این موارد، در قسمتی دیگر الگوریتمی جدید که سبب ترکیب SOCCA شده و با توجه به مقوله سازمانی SOA را با خود به همراه دارد تشریح شده و در نهایت روشی ایده‌آل را بیان می‌شود که با توجه به SOA، به بهبود الگوریتم زمان-بندی در حیطه محاسبات ابری کمک شایانی می‌کند.

۲. مروری بر محاسبات ابری و معماری سرویس‌گرا

معماری سرویس‌گرا و محاسبات ابری به یکدیگر مرتبط‌اند، SOA الگوی معماری است که راه‌حل‌های کسب و کاری برای ایجاد، سازماندهی و مصرف دوباره‌ی اجزای محاسباتی را هدایت می‌کند. به عبارت دیگر SOA و محاسبات ابری با هم همزیستی دارند و مکمل یکدیگرند و هم دیگر را پشتیبانی می‌کنند (Tsai, Sun, & Balasooriya, 2010), (Barry & Dick, 2013), (شمس و مهجوریان, ۱۳۸۹).

۱.۲. معرفی محاسبات ابری

محاسبات ابری مدلی است برای ایجاد دسترسی همگانی و به صورت آسان و بنا به سفارش شبکه. ابرها مجموعه‌ای از منابع محاسباتی پیکربندی پذیر مانند شبکه‌ها، سرورها، فضای ذخیره‌سازی، برنامه‌های کاربردی و سرویس‌ها اند که بتوانند با کم‌ترین نیاز به تلاش‌های مدیریتی یا تعامل با ارائه دهنده سرویس، به سرعت فراهم شده یا آزاد و رها شوند (National Institute of Standards and Technology, 2011). کاربرد محاسبات ابری برای کاربران و سایر سازمان‌ها سعی بر این داشته است که به جای نصب نرم‌افزار روی چند سیستم، تنها یک نرم‌افزار بر روی یک سرویس دهنده یک بار اجرا و بارگذاری کند که بقیه بتوانند به آن دسترسی داشته باشند. برخی از ویژگی‌ها مانند سرویس مبتنی بر تقاضا^۱، دسترسی وسیع شبکه^۲، اثتلاف منابع^۳ و انعطاف پذیری سریع^۴ سبب افزایش دسترسی به مقوله SOA در ترکیب با محاسبات ابری شده است (Dillon, Wu, & Chang, 2010), (Rittinghouse & Ransome, 2010). محاسبات ابری دارای لایه‌های سرویس مختلفی می‌باشد که در شکل ۱ مشاهده می‌شود. هر لایه زیر بنایی برای لایه بالایی خود است.



شکل ۱- سلسله مراتب محاسبات ابری (Tsai et al., 2010)

¹ On demand self service

² Broad network access

³ Resource pooling

⁴ Rapid elasticity

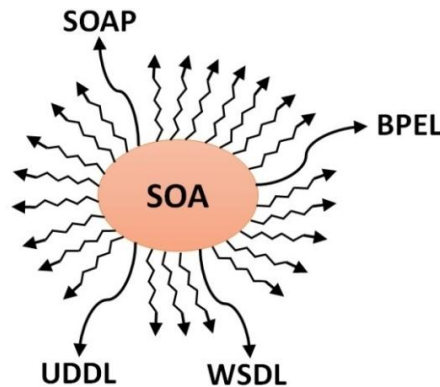
سرویس نرم‌افزار ابری سرویس‌های برنامه‌های کاربردی ابری یا نرم‌افزار به عنوان نرم‌افزار به صورت سرویس SaaS روی اینترنت قرار می‌گیرند. لایه سرویس به عنوان بستر ابری، به مصرف‌کننده این اجازه را می‌دهد تا بر روی محیطی که برنامه بر روی آن است تنظیماتی انجام داده و با کنترل و نظارت داشته باشد. در لایه سرویس زیرساخت ابری، مصرف‌کننده می‌تواند از منابع محاسبات پایه همانند قدرت پردازش، فضای ذخیره سازی و اجزای شبکه، میان-افزار استفاده کند. زیر بنای پردازش ابری، مراکز داده است. مراکز داده پایه‌ای برای ذخیره ابرها است (Tsai et al., 2010)، (Rittinghouse & Ransome, 2010)، (Barry & Dick, 2013)، (Magoulès, Pan, & Teng, 2012).

۲.۲. معماری سرویس‌گرا (SOA)^۵

معماری سرویس‌گرا بخشی نوظهور از معماری نرم‌افزار است. این بخش شامل روش‌های مختلف نامطمئن بوده و تاکید بیشتر بر آزمون و خطا ضروری است. SOA یک پدیده جدید بوده و برای پیاده سازی روش‌های آن نیاز به شناخت و تشریح بیشتر می‌باشد. خدمات از طریق ESB کشف و ارائه شده است. بنابراین ESP^۶ به عنوان یک کانال ارتباطی بین خدمات عمل می‌کند. این توابع عواملی از قبیل مسیریابی، امنیت، قابلیت اطمینان، خدمات مدیریت و ورود به سیستم را فراهم می‌کند. خدمات به دو بخش خدمات عمومی و خدمات فرآیند طبقه بندی می‌شوند (Waris, Khan, & Fakhar, 2013). معماری سرویس-گرا دارای چهارچوبی استاندارد است، همچنین صفات و خصوصیتی را در ذات خود دارد که سبب ایجاد قابلیت‌هایی از قبیل ساخته شدن، استقرار و یا مدیریت دیگر سرویس‌هاست. برای نمونه عدم امکان تعامل پذیری و یکپارچگی تمامی سیستم‌های اطلاعاتی سازمانی، در جهت استفاده از مؤلفه‌های نرم-افزاری مشترک، یکی از دلایل منطقی در معماری سرویس‌گرا است و همچنین می‌توان تفاوت میان دیدگاه‌های کارشناسان IT با کارکنان کسب و کار را از جمله آن موارد ذکر کرد (شمس و مهجوریان، ۱۳۸۹)، (Williams, 2014). شکل ۲ نمونه‌ای از پروتکل‌ها و استانداردها را در این زمینه نشان می‌دهد.

مهم‌ترین اهداف ارائه معماری سرویس‌گرا شامل موارد زیر اند: (شمس و مهجوریان، ۱۳۸۹)، (Williams, 2014):

- ایده‌آل کردن قابلیت استفاده مجدد و انعطاف پذیری
- انعطاف‌پذیری فناوری اطلاعات در پاسخ به تغییرات مداوم کسب و کار
- ایجاد استاندارد سازی و یکپارچگی سکوها و زیرساخت‌های فناوری اطلاعات
- هم‌راستایی کسب و کار با IT
- ایجاد تعامل پذیری بهتر در بین سازمانی



شکل ۲- پروتکل‌ها و استانداردها در SOA

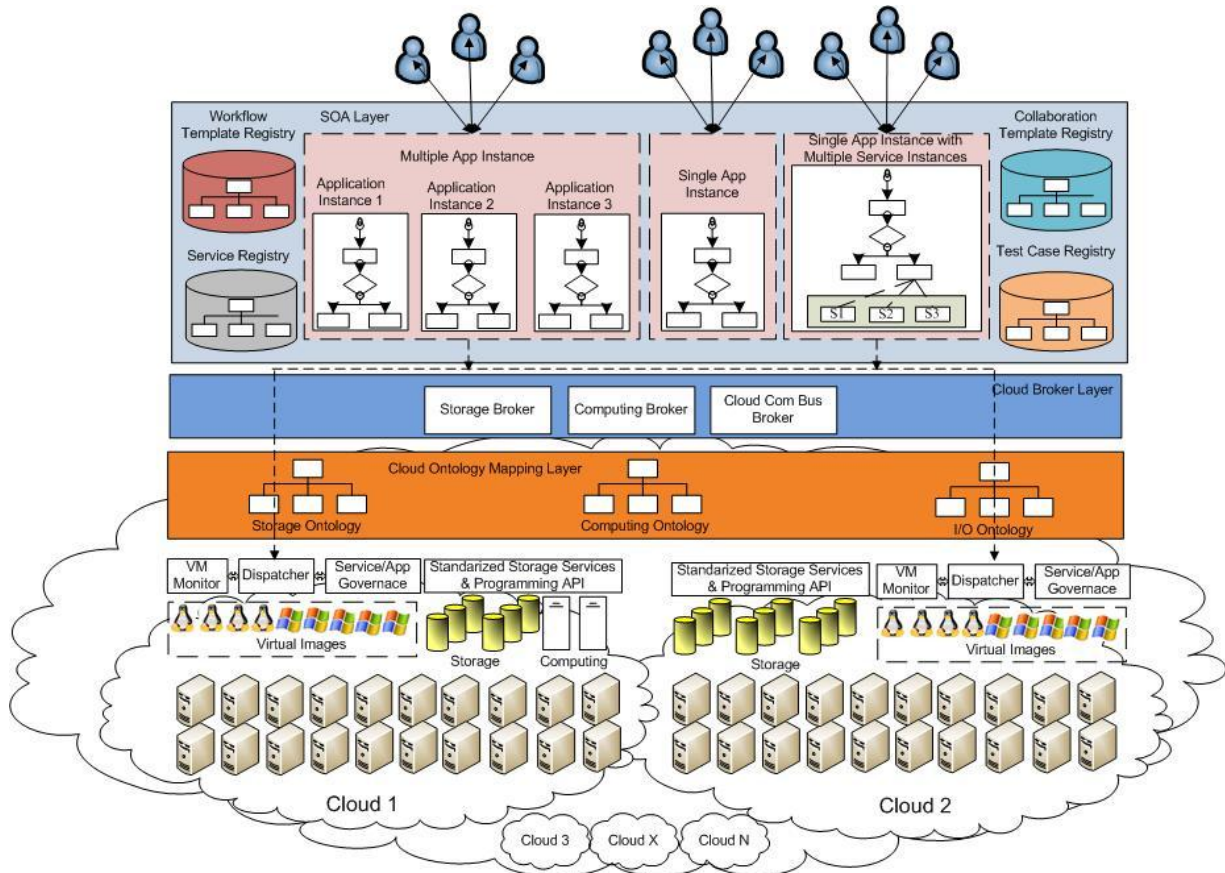
پروتکل‌ها و استانداردها در SOA بسیار منحصر به فرد اند. SOAP ساختاری برای تبادل پیام‌ها در قالب XML دارا است. WSDL زبانی مبتنی بر XML است که هدف آن توصیف ویژگی عملیاتی سرویس‌های وب می‌باشد. UDDL واسطی برای انتشار و شناسایی سرویس‌های وب است و شامل یک مخزن می‌شود که ارائه دهندگان به انتشار و تبلیغ سرویس خود می‌پردازند تا دیگران بتوانند آن را شناسایی کنند. BPEL زبانی برای مشخص کردن فرآیند-های اجرایی کسب و کار است (شمس و مهجوریان، ۱۳۸۹)، (Williams, 2014)، (Barry & Dick, 2013).

⁵ Service oriented architecture

⁶ Enterprise Service Bus

۳. رهیافتی برای ترکیب میان محاسبات ابری با SOA

پردازش ابری و SOA دارای نقاطی مهم و مشترک هستند. بیشترین اهمیت، در نزدیکی بالای پشته مربوط به پردازش ابری دیده می شود که اجزای کاربردی قابل دسترسی شبکه و سرویس های نرم افزاری هستند و مانند سرویس های همزمان شبکه پردازش ابری و معماری سرویس گرا، مفاهیم کلی سرویس گرا را به اشتراک می گذارند. SOA از منظر حیطه، عملکرد، اجرا و پیاده سازی سرویس های نرم افزاری، حائز اهمیت است. پردازش ابری بر روی جنبه هایی از پشته پردازش IT تمرکز می کند و می تواند آن را به محصول مصرفی تبدیل نمایند، که دریافت آن، به طور گام به گام از ایجادگرهای ابری ممکن باشد. به همین نحو، یک سکو برای اجرای کاربردهای شبکه گرا می تواند از مراکز مازاد بر احتیاج داده در ابر به اجاره گرفته شود. در شکل ۳ وابستگی این دو مقوله را می توان به وضوح مشاهده کرد.



شکل ۳- معماری سرویس گرای محاسبات ابری (Tsai et al., 2010)

۱.۳. معماری لایه ای SOCCA

۱.۱.۳. لایه ایجادگر ابری تکی

این لایه شبیه محاسبات ابری جاری است. هر ایجادگر ابری، مراکز اطلاعاتی خودش را می سازد و قدرت این را دارد که ابر به مراکز اطلاعاتی، خدمات (سرویس) ارائه کند. هر ابر ممکن است تکنولوژی های مجازی تحت تملک خودش را داشته باشد و یا تکنولوژی های مجازی Open source مانند Eucalyptus⁷ را بهینه سازی نماید. در این لایه ایجادگرهای ابری منابع و مجازی سازی را مراقبت می کنند (Tsai et al., 2010).

⁷ Eucalyptus System. <http://www.eucalyptus.com>

۲.۱.۳. آنتولوژی^۸ در مسیر دهی لایه‌ای ابری

ایجادگرهای ابری ممکن است با استانداردهای ذکر شده دقیقاً مطابقت نداشته باشد: همچنین این امکان وجود دارد که آنها ویژگی‌های اضافی را که شامل این استانداردها هستند، پیاده‌سازی نکنند. سیستم‌های مهم هستی شناسی که مورد نیازند به شرح زیر هستند (Tsai et al., 2010):

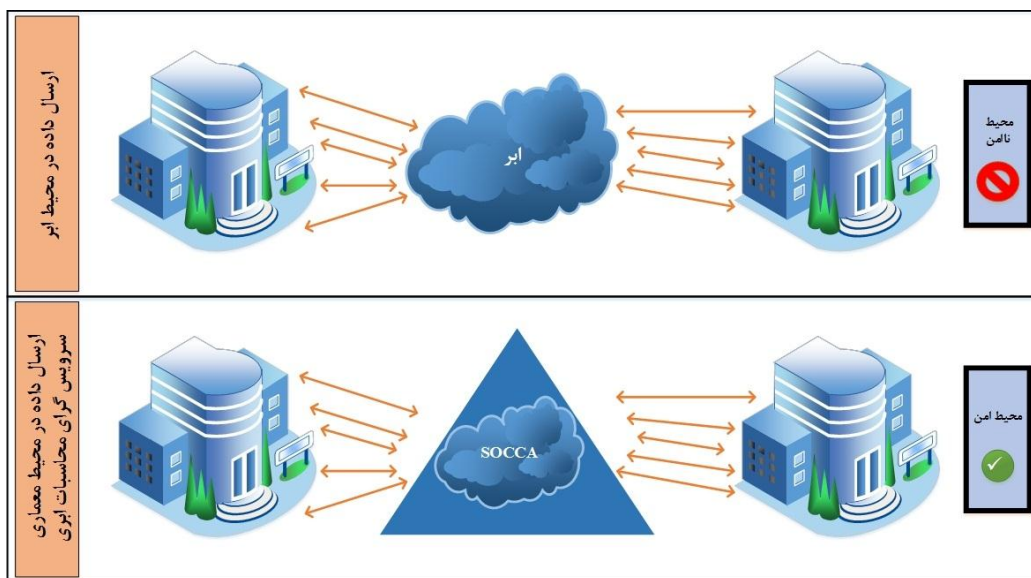
- ۱- آنتولوژی ذخیره‌ای: به مفاهیم و واژه‌هایی اطلاق می‌شود که با دستکاری داده‌ها در ابرها مرتبط اند همانند داده‌های به روز، افزودن داده‌ها، حذف داده‌ها، انتخاب داده‌ها و غیره.
- ۲- آنتولوژی محاسباتی: به مفاهیم و واژه‌های اطلاق می‌شود که با محاسبات توزیع شده درون ابرها مرتبط اند، همانند چارچوب‌های Map/Reduce.
- ۳- آنتولوژی ارتباطی: شامل تعریف مفاهیم و گرایش مربوط به الگوی ارتباطات از جمله ابرها، همانند کد گذاری الگو و مسیر پیغام‌ها هستند.

آنتولوژی‌ها باعث می‌شوند که همه ایجادگرها، با یک چارچوب^۹ به منظور تشکیل ابر، موجودیت پیدا کنند، اما در عوض هر ایجادگر ابری در ابرها چارچوب‌هایی را می‌سازد- مانند آن چه که در CCOA وجود دارد - و هر ایجادگر ابری اطلاعاتی را به روش یکسان به اشتراک می‌گذارد (Tsai et al., 2010).

۳.۱.۳. لایه کارگزار (واسطه‌ای) ابری^{۱۰}

این لایه، با اطلاعاتی سروکار دارد که به ایجادگر^{۱۱} (تامین کننده) ابری مربوط است. این لایه شامل اطلاعات ایجادگر ابری است که اطلاعاتی از قبیل قیمت گذار، سخت افزار، نرم افزار و سرویس‌های ایجاد شده عناصر هسته‌ای را شامل می‌شود. هر ایجادگر، رتبه بندی نحوه درجه بندی مشتریان مختص به خود را داراست. این مورد با مقایسه‌ی قیمت‌ها، قابلیت اطمینان و مرور دیدگاه افرادی که از آن ابر استفاده کرده اند، در ارتباط است. واسطه‌های ابری به عنوان عامل‌های مابین ایجادگر فردی ابری و لایه SOA عمل می‌کنند. هر سرویس عمده‌ی ابری، نوعی سرویس واسطه‌ای وابسته به خود را دارد و به طور کلی واسطه‌های ابری باید وظایف خاصی را انجام دهند (Tsai et al., 2010).

۲.۳. همپوشانی در محاسبات ابری و SOA



شکل ۴- مقایسه‌ای از ارسال داده در محیط ابر و SOCCA

محاسبات ابری به نسبت SOA مفهوم وسیع‌تری را پوشش می‌دهد که از طریق نرم‌افزار لایه‌ای ارائه‌ی پشته، کلیه پشته، حتی سخت‌افزار، را در بر می‌گیرد. با توجه به بحث ترکیب محاسبات ابری با SOA باید این مسئله را در نظر گرفت که SOA در بعضی از حوزه‌ها قوی‌تر از محاسبات ابری عمل می‌-

⁸ Ontology

⁹ Framework

¹⁰ Broker Layer

¹¹ Provider

کند و همینطور بالعکس؛ یعنی محاسبات ابری مواردی را در بر می گیرد که در آن قوی تر از SOA است. SOA و پردازش ابری می توانند به طور مستقل یا همگام، از روش دیدگاه مکملی، مورد توجه IT قرار گیرند. مساله مشخص شدن سرویس ها در ابعاد ابرها و SOA بسیار مهم است. که البته با توجه به پروتکل هایی نظیر UDDI، REST، SOAP، WSDL این امر ساده تر انجام می شود. بطور کلی مشخص شدن سرویس ها در داخل ابرها بحثی بسیار ضروری است.

در بحث ترکیب معماری SOA و محاسبات ابری می توان، دید سازمانی به مقوله محاسبات ابری داشت. لایه های کسب و کار در سیستم های IT دربرگیرنده این مقوله اند. با نگرش در سلسله مراتب محاسبات ابری و لایه SaaS می توان به این نتیجه رسید که معماری سرویس گرا به انعطاف پذیری بین مقوله های مدیریت ارتباط با مشتری^{۱۲}، برنامه ریزی منابع سازمان^{۱۳} و غیره به این فناوری کمک می کند. به طوری که با تغییر چالش های بین سازمانی و با کمک پروتکل های SOA تعامل پذیری در بین سازمان ها به سهولت انجام شود (Zhang & Zhou, 2009). همانطور که در حیطه مجازی سازی بحث شد، مجازی شدن، خلق نسخه مجازی از سیستم عامل است که وسیله ذخیره بر روی سکوی سخت افزاری است (He & Liang, 2012).

SOA دارای نظارت عمیقی در سیستم های خود است که این امر در محاسبات ابری به خوبی انجام نشده است و این خود یکی از مواردی است که امکان پسرقت فناوری را در محاسبات ابری به همراه دارد (شمس و مهجوریان، ۱۳۸۹)، (Williams, 2014)، (Marks, 2008). ولی این امر در SOA به صورت بسیار کامل و کارا رعایت می شود. توانایی تعیین خط مشی ها، در رابطه با خدمات و مدیریت تغییرات اعمال شده بر روی این سرویس ها، نشان دهنده قدرت در SOA است. SOA به خوبی می تواند تغییرات را در سیستم مدیریت کند. در این میان محاسبات ابری با ارائه منابع مورد نیاز به SOA کمک می کند تا قدمی سریع در پیشرفت فناوری بردارد. SOA استانداردهای خود را به همراه سرویس هایی برای برنامه های کاربردی تحت وب قرار می دهد. ولی محاسبات ابری سعی در سرویس دهی به کاربرانی دارد که از طریق ابر به نرم افزار دسترسی دارند. بوسیله SOA و محاسبات ابری می توان، یک همکاری سازمانی موثر و بسیار پویا را امکان پذیر ساخت.

وقتی در یک محیط ابری داده ها بین چندین سازمان ارسال می شوند، ممکن است یک محیط ناامن و سرشار از خطا داشته باشیم که باعث از بین رفتن درصدی از اطلاعات و داده ها شود. اما وقتی داده ها در محیط ابری مبتنی بر معماری سرویس گرا در بین سازمان ها رد و بدل می شوند، محیطی امن خواهد بود. معماری سرویس گرا، با توجه به فاکتورهایش نظیر استفاده از پروتکل ها، مفهوم سراسری، مفهوم برون سپاری^{۱۴} و ... محیط امنی را برای ما در پی خواهد داشت که داده ها در آن بدون چالش خاصی بین سازمان ها در ارتباط هستند. این مقوله در شکل شماره ۴، با توجه به دو محیط ابری و SOCCA به نمایش درآمده است.

۴. بررسی الگوریتم زمان بندی با توجه به مساله مشتری خدمتگزار

در این بخش، الگوریتم های زمان بندی ای تشریح خواهند شد که نیازهای کاربر را برطرف نموده و همچنین محدودیت های محاسبات ابری را از بین می برند. ابتدا استفاده از تئوری فروشگاه (مصرف کننده و سرویس دهنده) در زمان بندی محاسبات ابری مطرح می شود تا نیازهای کاربر معرفی گردد. در ابعاد دیگری از این موضوع، یک الگوریتم ارزش پایه زمان بندی ابری ارائه شده است.

۱.۴. برنامه زمان بندی منابع ابری

از آنجا که موجودیت منابع در خوشه های^{۱۵} ابری تغییر می یابد، دو فاز الگویی برای زمان بندی منابع ابری^{۱۶} ارائه شده است: بر طبق کاربرد، انتشار منابع ابری، موقعیت منابع و غیر قابل پیش بینی بودن. برای ذخیره منابع قبل از اجرای یک ماشین مجازی^{۱۷}، به منابع ابری نیاز است. الگوی زمان بندی ها در پروسه های زمان بندی باید اتمیک باشد، یعنی یا کامل اجرا شود یا اصلا اجرا نشود و لغو شود. برنامه زمان بندی منابع ابری می تواند مانند مشتری (مصرف کننده)^{۱۸} و فروشنده (تهیه کننده)^{۱۹} ارائه شود که به آن تئوری فروشگاه^{۲۰} می گوئیم. ابر، یک مکان فروش برای مشتری ها و فروشنده هاست. ما نیاز و پیشنهاد مشتری ها را ارائه می کنیم. با توجه به ۳ فاکتور نیاز مشتری، پیشنهاد ارائه شده و هدف فرمول نهایی، به نتیجه بررسی الگوریتم زمان بندی با توجه به مساله مشتری خدمتگزار در شکل ۵ دست می یابیم (Yang, Yin, & Liu, 2011).

¹² Customer Relationship Management (CRM)

¹³ Enterprise resource planning (ERP)

¹⁴ Form of outsourcing

¹⁵ Cluster

¹⁶ Cloud Resource Scheduling

¹⁷ Virtual Machine

¹⁸ Consumer

¹⁹ Supplier

²⁰ Market

فرمول نهایی:	پیشنهاد:	نیاز مشتری:
$\sum_{i=1}^n \delta_{ji} * \alpha_i \leq K_{j1},$ $\sum_{i=1}^n \delta_{ji} * \beta_i \leq K_{j2},$ $\sum_{i=1}^n \delta_{ji} * \gamma_i \leq K_{j3},$ $\sum_{i=1}^n \delta_{ji} = N_i,$ $P_{\min} \leq P \leq P_{\max}$	$P_{\min} \leq P \leq P_{\max}$	$\{N_i * \{\alpha_i, \beta_i, \gamma_i\}\}, i = 1..n$

شکل ۵- فرمول های برنامه زمان بندی (Yang et al., 2011)

سه شرط برای رسیدن به بهترین هدف در این تئوری وجود دارد (Yang et al., 2011):

۱. P کمتر از Pmin

۲. P مشتری را راضی میکند

۳. P بیشتر از Pmax

در شکل ۶، این سه شرط به همراه کل برنامه زمان بندی ابری به نمایش درآمده است.

بررسی شرط اول: به این معنی است که P از خواسته های کاربر بهتر است، ولی کاربر می خواهد به بالاترین قیمت منابع برسد که بالاترین اجرا را دارد. بنابراین می توانیم فرایند زمان بندی را با ارائه فروشنده با کمترین قیمت تطبیق دهیم. از آنجایی که ما اول N1 بعد N2 و ... را چک می کنیم، این یک تطبیق جزئی است. در جریان تطبیق، ما تفاوت بین جدیدترین موارد Ni و قدیمی ترین آنها را محاسبه می کنیم. اگر قیمت نهایی P به اضافه تفاوتش بیشتر از Pmin باشد حلقه شکسته می شود. توزیع باقی مانده از Ni+1 به Nn بدون تغییر باقی می ماند. این تطبیق می تواند زمان زیادی را ذخیره کند.

بررسی شرط دوم: برای شرط دوم ما به راه حل درست می رسیم. بنابراین زمان بندی کامل است.

بررسی شرط سوم: شرط سوم بدین معنی است که اگر منابع پایین همچنان موجود نباشد، سعی می کنیم بهترین محاسبه را داشته باشیم.

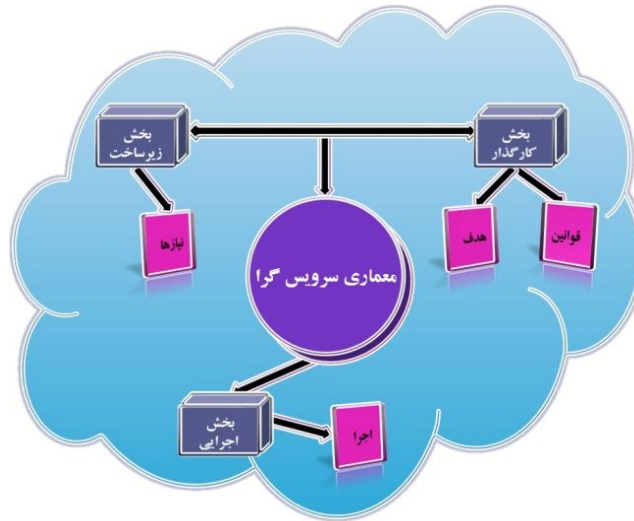
بنابراین قیمت تمام موارد نمی تواند راضی کننده باشد؛ حتی اگر فروشنده های معمول فروشگاه را راضی می کند.

- یک حلقه برای: مقدار ۱ تا n مشتری
- یک حلقه برای: مقدار ۱ تا n فروشنده
- قیمت فروشنده برای آیتم هایی که داشته است (ریشه ها، حافظه ها، دیسک ها)، با توجه به INSTJ
- اتمام حلقه / ایجاد حلقه جدید برای: مرتب کردن قیمت واحد
- اگر فروشنده راضی باشد، در INSTJ قرار بگیرد
- پایان دو حلقه.
- محاسبه قیمت نهایی P. اگر $P < P_{\max}$ بود، به کاربر گزارش میدهد قیمت نهایی کمتر از P باشد.
- اگر $P > P_{\min}$ بود، در INSTJ قرار بگیرد
- ایجاد یک حلقه جدید: جهت رضایت فروشنده
- اگر فروشنده راضی باشد، در INSTJ قرار بگیرد
- پایان حلقه
- در نهایت اگر $P < P_{\min}$ باشد، گزارش می دهد به کاربر که حداقل قیمت خیلی زیاد است و پیشنهاد می دهد که کاربر حداقل قیمت را کم کند.
- پایان برنامه: بهترین حالت این است که P مشتری را راضی کند.

شکل ۶- برنامه پایه زمان بندی ابری (Yang et al., 2011)

۵. ارائه یک روش ترکیبی برای بهبود الگوریتم زمان بندی در محاسبات ابری با توجه به حیطه SOA

ارائه خدمات محاسبات ابری در مورد روش جامع برای هر مدل تخصیص منبع، از اهمیت خاصی برخوردار است. لذا باید هر دو مورد منابع محاسبه و منابع شبکه را برای ارائه دقیق در الگوریتم زمان بندی در نظر بگیریم. محاسبات ابری در ابتدا برای شروع کارهای محاسبه ای تحت شبکه ایجاد شده اند. در این بحث به حیطه منابع مشتریان از لحاظ دسترسی و از لحاظ چگونگی منابع فیزیکی با توجه به لایه بندی در معماری SOCCA، اشاره می شود. تاخیرهای انتقال داده یکی از موانع اصلی است که با توجه رشد انبوه مشتری با آن مواجه می شویم. به عبارتی دیگر می توان گفت که وقتی مقدار زیادی داده به محیط تخصیص می یابد، کارایی خدمات شبکه مربوط به سیاست های کلی در قبال فرآیندهای دیگر در رابطه با زمان بندی خواهد بود. در نتیجه گفتنی است که مشتریان خصوصیات SOA را در ابر به عنوان عامل اصلی در تصمیم شان برای انتقال در نظر خواهند گرفت. مدل پیشنهاد شده در این بحث، با توجه به پارامترهای ارائه شده، همراه با چالش های تخصیص منابع مشتری/خدمتگزار و براساس دیدگاه لایه ای SOCCA و لایه کارگزار آن است.



شکل ۷- ساختار عمومی گره، الگوریتم زمان بندی محاسبات ابری مبتنی بر SOA

تئوری مشتری/خدمتگزار با توجه به الگوریتم های زمان بندی محاسبات ابری که مبتنی بر معماری ای برگرفته از معماری سرویس گرا است - همان گونه که در شکل ۷ به نمایش درآمده است- به صورت زیر بررسی می شود:

در ابتدا باید نیاز مشتری بررسی شود. یکسری فاکتورها برای اجاره کردن درخواست می شود. $\{N_1, N_2, \dots, N_n\}$. N_i تعداد ورود شمارنده i است. که منابع مورد نیاز شامل: ریشه ها a_i ، حافظه ها B_i و دیسک ها Y_i است. نیاز مشتری با استفاده از فرمول ۱ بررسی می شود:

$$\{N_i * \{\alpha_i, \beta_i, \gamma_i\}\}, i = 1..n \quad (1)$$

سپس باید یک هدف کلی از کار ارائه شود. که این هدف می تواند از سوی مشتری پیشنهاد شود که در محدوده زیر است:

$$P_{\min} \leq P \leq P_{\max} \quad (2)$$

مکان های مشتری به قیمت نهایی فاکتورهایی که در نظر دارد، محدود می شود. یعنی قیمت نهایی باید بیشتر یا برابر با P_{\min} و یا کمتر یا مساوی

با P_{\max} باشد.



شکل ۸- تطبیق دهنده SOA برای هماهنگی n واحدی سازمانی در محیط ابر

پس از تعیین اهداف، در این تئوری باید به یک قانون جامع رسید، تا مفهوم SOA در قالب n واحد سازمانی در محیط ابری همانند شکل ۸ مشخص گردد (این قانون برای پیدا کردن n محیط سازمانی است).

بر اساس اصل استقرا ریاضی $(\forall n \in N)[P(n)] \Rightarrow (\forall k \in N)[P(k) \Rightarrow P(k+1)]$ ، فرمول ۳ برقرار است:

$$(n * (n + 1) * (2n + 1) / s) \tag{3}$$

با توجه به نیاز، هدف و قانون گفته شده، فروشنده در نهایت با فرمول ۴ به هدف نهایی خود می‌رسد.

$$\sum_{i=1}^n \delta_{ji} * \alpha_i \leq K_{j1},$$

$$\sum_{i=1}^n \delta_{ji} * \beta_i \leq K_{j2},$$

$$\sum_{i=1}^n \delta_{ji} * \gamma_i \leq K_{j3},$$

$$\sum_{i=1}^n \delta_{ji} = N_i,$$

$$P_{\min} \leq P \leq P_{\max} \tag{4}$$

نتیجه، رسیدن به حداقل قیمت P است که مشتری را راضی نگه می‌دارد.

کار در سه بخش انجام شده است: در بخش زیرساخت SOA نیاز مشتری گفته شده است، سپس در لایه Broker یا همان بخش Broker، هدف و قانون کار آماده می‌گردد. در نهایت با توجه به مسائل مطرح شده در اجرای کار به نتیجه اصلی دست می‌یابیم. مطالب بررسی شده در قالب شکل ۹ به نمایش درآمده است.

```
بخش : Infrastructure SOA
R_Start (SOA_Name)
E_Start (N_Client)
Chek (Layer_SOA)
ListOK (OK)

بخش : Broker Layer
Resource (R_Client)
Roll (Rolls_n)

بخش : Run SOA
Do {
  If ( (N_Client) == (Layer_SOA) ) {
    OK
  }
  // N_Client = {N_i * {alpha_i, beta_i, gamma_i}, i = 1..n
  If ( (R_Client) == (Layer_SOA) ) {
    OK
  }
  // R_Client = P_min <= P <= P_max
  If ( (Roll_n) == (Layer_SOA) ) {
    OK
  }
  // Roll_n = (n * (n + 1) * (2n + 1) / s)
  If ( (N_Client) && (R_Client) && (Roll_n) ) {
    OK
  }
  // Run:
  sum_{i=1}^n delta_{ji} * alpha_i <= K_{j1},
  sum_{i=1}^n delta_{ji} * beta_i <= K_{j2},
  sum_{i=1}^n delta_{ji} * gamma_i <= K_{j3},
  sum_{i=1}^n delta_{ji} = N_i,
  P_min <= P <= P_max
}
End
```

شکل ۹- الگوریتم پایه زمان‌بندی محاسبات ابری مبتنی بر معماری سرویس‌گرا

حال با توجه به این الگو - مفهوم کامل یکپارچگی بین سازمانها توسط لایه Broker (واسطه‌ای) که بیان شده است - می‌توانیم کلیه سازمانها را در یک حیطه منحصر به فرد در راستای بهبود سطح فعالیت‌های بین سازمانی تعریف کنیم. در این حیطه، یک الگو برای همه سازمانها توسط لایه واسطه‌ای ایجاد شده است و کلیه داده‌ها، فرآیندها، منابع در این حیطه یکی شده و سطح امنیت در این معماری جدید بهبود می‌یابد و همچنین رضایت مشتری از توجه به صرفه جویی از سایر پارامترهای تحت لایه واسطه‌ای بیشتر خواهد شد.

۷. نتیجه گیری

محاسبات ابری، به عنوان یک شاهکار خارق العاده، همه ویژگی‌های عالی را داراست، هرچند هنوز محاسبات ابری دور از واقعیت ذات اصلی خود بوده و تحقیق و کنکاش بیشتر در این زمینه لازم است. معماری سرویس‌گرا، می‌تواند در پیشرفت فناوری ابری کمک شایانی نماید. فناوری قدرتمند محاسبات ابری، سبب ایجاد یک پایه و اساس ایده‌آل برای همگام شدن و مکمل شدن با معماری سرویس‌گرا شود که با توجه به موقعیت IT سال‌های اخیر در دنیای امروزی، می‌تواند به رشد و پیشرفت در این زمینه منجر شود. در این مقاله الگوریتم‌هایی را مورد بحث قرار دادیم که برای محاسبات ابری بسیار ضروری می‌باشند و محاسبات ابری زمان‌بندی منابع خود را بر اساس این الگوریتم‌ها انجام می‌دهد. از سوی دیگر، شناخت و بررسی لایه‌های SOCCA، شمار زیادی از مسئله‌ها مانند عملکرد مفید در الگوریتم زمان‌بندی محاسبات ابری را برای ما آشکار می‌سازد. همچنین، معماری لایه‌ای SOCCA به ما در فهم منطق میان دو مقوله SOA و محاسبات ابری کمک می‌کنند؛ زیرا بدون اطلاع از لایه‌ها در SOCCA، نمی‌توان به مدل ایده‌آل و ترکیبی در رابطه با الگوریتم زمان‌بندی در حیطه رضایت‌مندی مشتری دست پیدا کرد. با توجه به مساله زمان‌بندی در محاسبات ابری می‌توان از حیطه SOCCA بهره گرفت، همچنین می‌توانیم به کمک نیازهای مشتری در رابطه با بحث SOCCA یکسری از درخواست‌ها را با توجه به پارامترهای ریشه، حافظه و دیسک‌ها با توجه به محدودیت‌های مشتری بهبود بدهیم. این طرح ارائه شده همانند الگویی از پیش تعریف شده وارد عمل می‌شود، که با توجه به مساله یکپارچگی سازمان‌ها می‌تواند به بهترین نحو عمل کند و فاکتوری مهم برای رضایت‌مندی مشتریانی باشد که تحت محیط محاسبات ابری، سعی در اجرای اهداف خود دارند. در آینده می‌توان این ایده را با دیگر پارامترهایی نظیر سرعت و انرژی مصرفی در مراکز داده‌ای توزیع شده در محاسبات ابری در نظر گرفت که با توجه به آنتولوژی‌ها در SOCCA و لایه واسطه‌ای، می‌توان این بحث را کامل نمود.

منابع

1. Tsai W., Sun X., Balasooriya J., 2010. Service-Oriented Cloud Computing Architecture, Seventh International Conference on Information Technology, Las Vegas, NV
2. Barry D.K., Dick D., 2013, Web Services, Service-Oriented Architectures, and Cloud Computing. Elsevier. Second Edition. Theusa
3. فریدون شمس و امیر مهجوریان، ۱۳۸۹، " معرفی اصول و روش‌های معماری سازمانی سرویس‌گرا "، مرکز چاپ و انتشارات دانشگاه شهید بهشتی، چاپ اول، تهران
4. National Institute of Standards and Technology (NIST). <http://www.nist.gov/itl/csd/cloud-102511.cfm>
5. Dillon T., Wu C., Chang E., 2010, Cloud Computing: Issues and Challenges, 24th IEEE International Conference on Advanced Information Networking and Applications, Perth, Australia
6. Rittinghouse W.J., Ransome F. J., 2010, Cloud Computing Implementation Management and Security. CRC Press Taylor & Francis Group. USA, NY
7. Magoulès, F., Pan, J., & Teng, F. (2012). *Cloud computing: Data-intensive computing and scheduling*. CRC Press.
8. Waris, M., Khan, S. A., & Fakhar, M. Z. (2013, October). Factors effecting service oriented architecture implementation. In *Science and Information Conference (SAI), 2013* (pp. 1-8). IEEE
9. Williams, W. (2014). Security for Service Oriented Architectures. CRC Press
10. Zhang L., Zhou Q., 2009, CCOA: Cloud Computing Open Architecture, IEEE International Conference on Web Services, Los Angeles, CA
11. He, Z., & Liang, G. (2012, October). Research and evaluation of network virtualization in cloud computing environment. In *Networking and Distributed Computing (ICNDC), 2012 Third International Conference on* (pp. 40-44). IEEE.
12. Marks, E. A. (2008). *Service-oriented architecture (SOA) governance for the services driven enterprise*. John Wiley & Sons.
13. Yang, Z., Yin, C., & Liu, Y. (2011, October). A cost-based resource scheduling paradigm in cloud computing. In *Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2011 12th International Conference on* (pp. 417-422). IEEE.