

## بررسی انواع تحمل پذیری اشکال در رایانش ابری

مصطفی رضازاده

دانشگاه آزاد اسلامی، واحد علوم و تحقیقات، گروه کامپیوتر، تهران، ایران  
Mostafarezazadeh@Rocketmail.com

چکیده - تحمل پذیری اشکال یکی از مهم ترین موضوع در بحث رایانش ابری است. نوع های مختلفی از تحمل پذیری اشکال وجود دارند که یکی از آنها تحمل پذیری اشکال در رایانش ابری بلادرنگ است. سیستم های بلادرنگ با داشتن امتیازاتی از قبیل ابزارهای محاسبات قوی از اهمیت بسزایی برخوردارند. در رایانش ابری بلادرنگ استفاده از نودهای پردازشگر احتمال اشکال را بالا می برد و از طرفی چون سیستم های بلادرنگ از لحاظ ایمنی بحرانی هستند باید قابلیت اطمینان آنها افزایش یابد از اینرو تقاضا برای دستیابی به سیستم هایی با قابلیت تحمل اشکال در سیستم های بلادرنگ رو به افزایش است به همین ترتیب در تحمل پذیری اشکال بایزنتاین که مبحث مهمی در ابرهای بهم پیوسته و در رایانش ابری با منابع اختیاری است. مدل های برای تحمل پذیری اشکال در اینگونه ابرها ارائه شده که در این مدل ها برای بهبود قابلیت اطمینان از افزونگی استفاده شده است و برای سازگاری محیط ابری با منابع اختیاری سعی شده نودهایی با خصیصه هایی بالایی همچون قابلیت اطمینان و کارایی انتخاب شود به گونه ای که اگر چنین مدلهایی برای تحمل پذیری اشکال ارائه نشود سبب خسارتهای جبران ناپذیری می شود.

کلید واژه ها - رایانش ابری، تحمل پذیری اشکال، ماشین مجازی

این مشکل استفاده شده است و اهمیت این موضوع در بسیاری از زمینه ها غیر قابل انکار است به طور نمونه در سیستم های بلادرنگ که با استفاده از رایانش ابری پیاده سازی شده است اگر سیستم تحمل پذیر اشکال نباشد موجب خسارتهای جبران ناپذیری خواهد شد در ادامه انواع تحمل پذیری اشکال را به طور مشروح مورد بررسی قرار خواهیم داد.

### ۲.۱. تحمل پذیری اشکال در رایانش ابری بلادرنگ

سیستم های بلادرنگ با داشتن امتیازاتی از قبیل ابزارهای محاسبات قوی، محیط مجازی سازی با مقیاس پذیری بالا از اهمیت بسزایی برخوردارند. در رایانش ابری استفاده از نودهای پردازشگر احتمال اشکال را بالا می برد و از طرفی چون سیستم های بلادرنگ از لحاظ ایمنی بحرانی هستند باید قابلیت اطمینان آنها افزایش یابد از اینرو تقاضا برای دستیابی به سیستم هایی با قابلیت تحمل خطا در سیستم های بلادرنگ رو به افزایش است [۳ و ۲]. در سیستم پیشنهادی که در شکل ۱ نمایش داده شده است که شامل نودهای پردازشگر است اشکال تحمل شده است در این سیستم، قابلیت اطمینان هر نود محاسبه می شود اگر هر نود نتایج درست تولید کند قابلیت اطمینان آن افزایش می یابد در غیر اینصورت کاهش می یابد. در این سیستم هم از

### ۱. مقدمه

با نیاز روزافزون کاربران به منابع مختلف، رایانش ابری به عنوان یکی از تکنولوژی های بشر مطرح گردیده و امروزه به سرعت در حال پیشرفت است. فروشندگان، یک نرم افزار گران قیمت را که توان محاسباتی بالا و حافظه زیادی نیاز دارد از طریق ابر در اختیار متقاضیان قرار می دهند. اهمیت سرویس ابر اینست که تمام منابع و سرویس های مختلف در اختیار فراهم کنندگان بوده و مشتریان نهایی نیازی به اطلاعات تخصصی و پرداخت هزینه نصب این گونه سرویس ها ندارند [۱]. بنابراین مشتریان تنها برای استفاده از سرویس هزینه پرداخت می کنند. موضوعات متفاوتی در رایانش ابری مطرح است که تحمل پذیری اشکال به عنوان یکی از موضوعات مهم در این زمینه به دلیل وجود منابع چندگانه مطرح است. در این مقاله سعی شده است انواع تحمل پذیری اشکال مورد بررسی قرار گیرد.

### ۲. تحمل پذیری اشکال در رایانش ابری

یک مبحث مهم در رایانش ابری، تحمل پذیری اشکال به دلیل وجود منابع مختلف است که از تکنیک های متفاوتی برای حل

اطمینان جدیدی داده می شود و در انتها مکانیسم تصمیم گیرنده بهترین نود را که دارای بالاترین قابلیت اطمینان است انتخاب می کند. نتایج آزمایش انجام گرفته شده در جدول ۱ آورده شده است.

جدول ۱: نتایج آزمایش برای برآورد قابلیت اطمینان [۲].

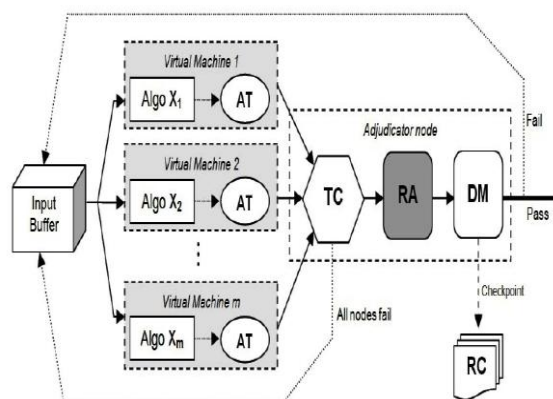
| Cycle | Real Time Limit (ms) | Virtual Machine-1 |      |      |             | Virtual Machine-2 |      |      |             | Virtual Machine-3 |      |      |             | DM Selected Node |
|-------|----------------------|-------------------|------|------|-------------|-------------------|------|------|-------------|-------------------|------|------|-------------|------------------|
|       |                      | AT                | TC   | Time | Reliability | AT                | TC   | Time | Reliability | AT                | TC   | Time | Reliability |                  |
| Start |                      | -                 | -    | -    | 1           | -                 | -    | -    | 1           | -                 | -    | -    | 1           | -                |
| 1     | 2500                 | Pass              | Pass | 2174 | 1.020       | Fail              | Fail | 2997 | 0.980       | Pass              | Pass | 2238 | 1.020       | VM-1             |
| 2     | 3700                 | Pass              | Fail | 3901 | 1.000       | Pass              | Fail | -    | 0.941       | Pass              | Pass | 3599 | 1.040       | VM-3             |
| 3     | 2150                 | Pass              | Fail | 3477 | 0.960       | Pass              | Pass | 2101 | 0.960       | Pass              | Pass | 2084 | 1.061       | VM-3             |
| 4     | 6300                 | Pass              | Fail | -    | 0.902       | Pass              | Pass | 5732 | 0.979       | Pass              | Pass | 6113 | 1.082       | VM-3             |
| 5     | 1950                 | Fail              | Fail | -    | 0.830       | Pass              | Pass | 1906 | 0.998       | Fail              | Fail | 1892 | 1.061       | VM-2             |
| 6     | 2800                 | Pass              | Pass | 2791 | 0.846       | Pass              | Pass | 2682 | 1.018       | Fail              | Fail | 2653 | 1.018       | VM-2             |
| 7     | 2350                 | Pass              | Pass | 2269 | 0.863       | Pass              | Pass | 2312 | 1.039       | Pass              | Fail | 2771 | 0.957       | VM-2             |
| 8     | 3200                 | Pass              | Pass | 3075 | 0.881       | Pass              | Pass | 3102 | 1.059       | Pass              | Fail | -    | 0.881       | VM-2             |
| 9     | 3900                 | Pass              | Pass | 3618 | 0.898       | Pass              | Fail | 3949 | 1.038       | Pass              | Pass | 3772 | 0.898       | VM-1             |
| 10    | 2650                 | Fail              | Fail | 2589 | 0.880       | Pass              | Pass | 2601 | 1.059       | Pass              | Fail | 3110 | 0.826       | VM-2             |

در سیکل اول، ماشین مجازی اول و سوم دارای قابلیت اطمینان برابر بوده اما بخاطر اینکه آدرس IP ماشین مجازی اول کمتر است انتخاب خواهد شد. در سیکل دوم تا چهارم ماشین مجازی سوم بدلیل بدست آوردن قابلیت اطمینان بیشتر انتخاب شده است در سیکل پنجم، ماشین مجازی سوم دارای بالاترین قابلیت اطمینان است اما بدلیل اینکه در تست پذیرش موفق نبوده است به مرحله بعدی انتقال داده نمی شود و ماشین مجازی دوم انتخاب می شود در سیکل دهم، ماشین مجازی اول با اینکه زمان آن به پایان نرسیده است در چک کننده زمان با شکست مواجه شده است بدلیل آنکه در تست پذیرش موفق نبوده است.

طرح پیشنهادی انتخاب خوبی برای مکانیسم تحمل اشکال در محاسبات بلادرنگ بوده و با استفاده از مکانیسم تصمیم سعی بر آن شده که نود دارای بیشترین قابلیت اطمینان انتخاب شود. احتمال شکست در سیستم پیشنهادی خیلی پایین است چرا که در این مدل هم از ترمیم رو به جلو و هم از ترمیم رو به عقب استفاده شده است از ترمیم رو به جلو استفاده می شود تا زمانی که تمام نودها با شکست مواجه شوند یعنی این که اگر  $n$  نود وجود داشته باشد تحمل خطای آن  $n-1$  است اگر تمامی نودها با شکست مواجه شوند آنگاه از ترمیم رو به عقب استفاده می شود نداشتن اثرات زنجیره ای و استفاده از check point از دیگر مزایای این طرح است اما معایب این طرح به این شرح است که قابلیت اطمینان را بالاتر از یک در نظر گرفته است و در مورد قابلیت اطمینان ماژول هایی مانند تست پذیرش، چک کننده زمان صحبتی نکرده است.

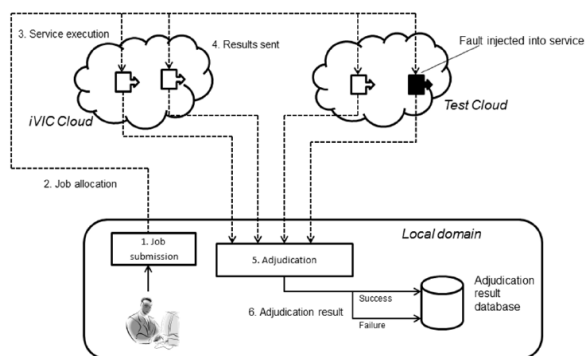
ترمیم رو به جلو و هم از ترمیم رو به عقب استفاده می شود اما تمرکز آن روی ترمیم رو به جلو است و بخاطر استفاده از افزونگی سیستم تحمل پذیر اشکال شده است. این مدل که در شکل ۱ نشان داده شده است اشکال را براساس قابلیت اطمینان هر نود تحمل می کند هر ماشین مجازی براساس قابلیت اطمینان انتخاب می شود و می تواند حذف شود در صورتی که نتواند بخوبی برنامه بلادرنگ را اجرا کند [۲].

در این مدل ما دو نوع نود داریم یکی از اینها ماشین مجازی است که برروی زیرساخت ابر اجرا می شود و دیگری هم نود قضاوت است. ماشین مجازی شامل الگوریتم های برنامه کاربردی بلادرنگ و تست پذیرش برای تصدیق منطقی آن است و نود قضاوت کننده شامل چک کننده زمان، تخمین زننده قابلیت اطمینان و مکانیسم تصمیم گیرنده می باشد. مکان نود قضاوت کننده به نوع سیستم بلادرنگ و سناریوهایی که در آن اجرا می شود دارد می تواند به بخشی از زیرساخت ابر یا بعنوان بخشی از زیرساخت کاربر باشد به طور کلی نزدیک سنسورها و یا محرکها واقع شده است.



شکل ۱: مدل سیستم پیشنهادی [۲].

در انتها آزمایش با استفاده از اینترفیس ProActive گرید در ابر آمازون EC2 انجام گرفته است در این آزمایش سه ماشین مجازی ایجاد شده است و الگوریتم های بلادرنگ در آن نصب شده است الگوریتم ده سیکل محاسبه دارد در اینجا یک بافر ورودی وجود دارد که ورودی یکسان به تمام الگوریتم ها می دهد ماشینهای مجازی پس از اجرا خروجی خود را به ماژول تست پذیرش برای بررسی منطقی می دهند و نتایج درست به ماژول چک کننده زمان فرستاده می شود تا بررسی شود که آیا هر نود توانسته نتیجه خود را در مهلت زمانی مقرر تولید کند یا نه. پس از آن در ماژول تخمین زننده قابلیت اطمینان به هر نود قابلیت



شکل ۲: مدل ایجاد شده با استفاده از FT\_FC [۴].

**Service execution:** هر ماشین مجازی شامل یک نسخه

پشتیبان است که در صورت خطا به آن سوئیچ می‌کند.

**Result sent:** هر سرویس نتایج خود را به نود قضاوت ارائه می‌کند.

**Adjudication:** نود قضاوت کننده تصمیم می‌گیرد که آیا نتایج برگشت داده شده درست است یا در بازه زمانی مقرر تولید شده است.

**Adjudication result:** نتیجه ممکن است درست باشد یا غلط. به هر حال نتیجه در یک پایگاه داده موجود در قلمرو محلی ثبت می‌شود [۴].

این مدل به صورت تجربی مورد آزمایش قرار گرفت و در آن از دو ماشین مجازی متفاوت با نامهای iVIC و Test cloud برای انجام آزمایش مورد استفاده قرار گرفت نتایج آزمایش در جدول ۲ قابل مشاهده است.

جدول ۲: نتایج بدست آمده با نرخ شکست متفاوت [۴].

| Fault Type | Failure %<br>iVIC, Test<br>Cloud | Experiment<br>1<br>adjudicate<br>success % | Experiment<br>2<br>adjudicate<br>success % | Abj.<br>difference<br>% |
|------------|----------------------------------|--|--|-------------------------|
| Value      | 10-5                             | 95.68                                      | 94.2                                       | 1.57                    |
|            | 5-1                              | 99.76                                      | 98.8                                       | 0.97                    |
| Time       | 10-5                             | 96.4                                       | 94.1                                       | 2.44                    |
|            | 5-1                              | 99.44                                      | 98.6                                       | 0.85                    |
| Omission   | 10-5                             | 96.6                                       | 92.6                                       | 4.31                    |
|            | 5-1                              | 99.48                                      | 98   | 1.51                    |

سه نوع خطا در این آزمایش در نظر گرفته شده است در ستون دوم ۵-۱۰ نشان دهنده آن است که نرخ شکست iVIC ۱۰ است و ۵ هم نشان دهنده سیستم ابری است.

## ۲.۲. تحمل پذیری اشکال بایزنتاین در رایانش ابری

اشکال بایزنتاین، اشکالی است که در آن سرویس دهنده ممکن است به دلخواه هر پاسخ را در همزمان تولید نماید که ناشی از حملات دشمن، خطای کاربر و خطای نرم افزار است که یک روش برای جلوگیری از آن استفاده از تحمل پذیری اشکال بایزنتاین (BFT) است. دو جنبه متفاوت در راستای این موضوع وجود دارد که در ادامه به شرح آن می‌پردازیم.

### ۲.۲.۱. تحمل پذیری اشکال بایزنتاین در رایانش ابری بهم پیوسته

دو قاعده کلی که در این موضوع وجود دارد یکی هماهنگی بین ابرهای چندگانه که تحت نظارت سازمان‌های متفاوت و دامنه مدیریتی متفاوت هستند و دوم قادر بودن انتقال و تعریف اطلاعات است که سعی شده کارایی BFT با انجام آزمایش واقعی در فدراسیون ابر موجود به صورت واقعی تست شود. BFT در سیستم‌های ابری تکی استفاده نمی‌شود چرا که استفاده از آن در سیستم‌های مجزا سبب به مخاطره انداختن قابلیت اطمینان کل سیستم و ایجاد هزینه‌های گزاف می‌شود اما به طور کلی استفاده از BFT مزایایی هم دارد استفاده از این طرح باعث می‌شود هزینه‌های اضافی بعداً نکنیم و دیگر اینکه BFT می‌تواند جامعیت داده شود و محاسبات را بهبود ببخشد و راهی را برای ماسک کردن ناسازگاری بالقوه ایجاد کند [۴].

یک قالب کاری برای تحمل پذیری اشکال در سیستم‌های ابری بهم پیوسته (FT\_FC) ارائه شده است که شامل ویژگی‌های زیادی می‌باشد از جمله یک ابزار زمانبندی کار به صورت اتوماتیک، که اجازه می‌دهد برنامه کار به صورت اتوماتیک به چند ابر ناهمگن ارائه شود و یک سیستم پیام که ارتباط مطمئن را بین ابر و FT\_FC ایجاد می‌کند و یک سیستم قضاوت تحمل پذیر اشکال هم ارائه داده است. شکل ۲ سیستم ایجاد شده با استفاده از قالب کاری FT\_FC را نشان می‌دهد.

هریک از ماژولها وظایف خاصی دارند که در ادامه شرح داده می‌شود:

**Job submission:** سیستم به صورت اتوماتیک کارها را به ابرهای مجزا ارائه می‌کند.

**Job allocation:** کارها به سرویس‌ها که به روش N-copy تهیه شده‌اند تخصیص داده می‌شود.

هریک از ماژولها وظایف خاصی دارند که در ادامه شرح داده می شود:

**Primary Selection:** پس از دریافت درخواست از ماژول ابر، با استفاده از الگوریتم انتخاب Primary یک نسخه اصلی با توجه به نیازمندیهای QOS انتخاب می شود.

**Replicas Selection:** در این فاز مجموعه ای از نسخه های پشتیبان بر اساس نیازمندیهای QOS و با توجه به الگوریتم انتخاب نسخه پشتیبان انتخاب می شوند. و نسخه اصلی درخواست را برای اجرا به نسخه های پشتیبان می فرستد. نسخه اصلی با نسخه های پشتیبان تشکیل یک گروه BFT می دهند.

**Request Execution:** در این فاز تمام اعضای گروه، درخواست را به صورت محلی اجرا می کنند و نتیجه را در بازه زمانی معین به ماژول ارائه می کنند اگر نتایج درست بودند ماژول ابر درخواست بعدی را می فرستد اگر نتایج ناسازگار بودند رویه تحمل پذیر خطا برای تحمل  $f$  خطا تریگر می شود و نسخه اصلی و پشتیبان بروز می شوند اگر بیش از این مقدار خطا وجود داشت ماژول ابر درخواست را دوباره می فرستد و این فاز دوباره تکرار می شود.

**Primary Updating:** در این فاز نسخه اصلی معیوب موجود در گروه مشخص شده و با نسخه اصلی جدید تعویض می گردد.

**Replica Updting:** در این فاز نسخه های پشتیبان معیوب بر اساس اطلاعات بدست آمده از فاز اجرای درخواست بروزرسانی می شوند الگوریتم بروزرسانی پشتیبان، پشتیبان های معیوب را با نودهای مناسب موجود در ابر تعویض می کند [۵].

در انتها این مدل با دیگر مدلهای موجود مورد مقایسه قرار گرفته در این نوع از سیستم تحمل پذیری اشکال برای بهبود قابلیت اطمینان از افزونگی استفاده شده است و برای سازگاری محیط ابری با منابع اختیاری سعی شده نودهایی با خصیصه هایی بالایی همچون قابلیت اطمینان و کارایی انتخاب شود.

## ۲.۳. میان افزار با قابلیت تحمل پذیری اشکال در رایانش ابری

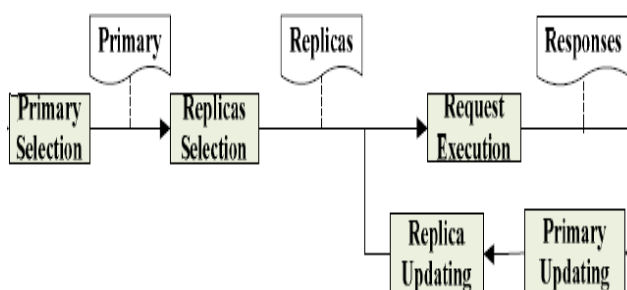
یکی از چالش ها در رایانش ابری این است که باید سرویس های کاربران بدون وقفه انجام شود [۷۶]. این میان افزار تحمل پذیری اشکال را برای برنامه های کاربردی توزیع شده در محیط رایانش ابری یا مرکز داده فراهم می کند همانند سرویس هایی که توسط مالکان ابر ارائه می شود. این میان افزار

## ۲.۲.۲. تحمل پذیری اشکال بایزنطین در رایانش ابری با منابع اختیاری

بیشتر ابرها بر دو نوع از زیرساختها استقرار یافتند یکی اینکه از لحاظ مدیریتی قویتر و دارای زیرساختار قدرتمند می باشد و توسط تهیه کنندگان ابر بزرگ مدیریت می شوند مانند آمازون و گوگل. و دیگری با زیرساختار منابع اختیاری که شامل تعداد زیادی کاربران که منابع محاسباتی خود را ترکیب کرده اند. در منابع اختیاری بعلت پویا بودن محیط، قابلیت اطمینان یک سیستم یک مسئله بحرانی است که برای حل این مشکل یک سیستم تحمل پذیر خطای بایزنطین (BFT) ارائه شده است. اگر تعداد منابع ما  $3f+1$  باشد  $f$  خطای بایزنطین را تحمل می کند. در این نوع از تحمل پذیری اشکال در مقیاس بزرگ که شامل ۲۵۷ منابع اختیاری در ۲۶ کشور مورد ارزیابی قرار گرفته شده است. BFT توانسته قابلیت دسترسی بالا را در این زیرساختار فراهم کند و عملکرد خوبی از این سیستم را تضمین نماید. معمولاً زیرساختار ابر با منابع اختیاری نسبت به ابرهای بزرگ ارزانتر و پویایی آنها بیشتر، قدرت کمتر و دارای قابلیت اطمینان کمتری می باشند و لینک ارتباطی بین ماژولها قابل اطمینان نیست [۵].

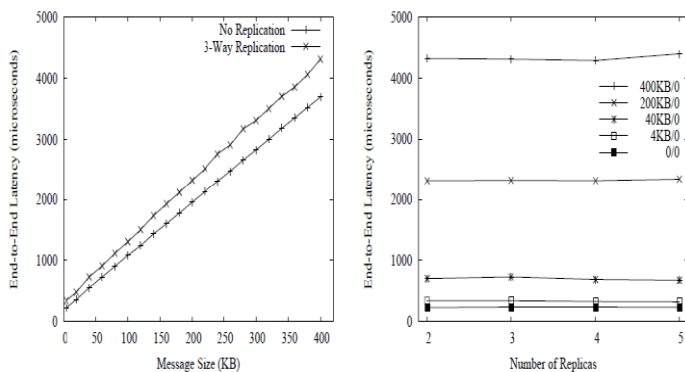
در این نوع از تحمل پذیری اشکال کارهایی همچون بررسی مسائل تحمل پذیری اشکال و ارائه یک قالب کاری جدید، بررسی اینکه این سیستم در بهبود قابلیت اطمینان در مقایسه با دیگر رویکردها بهتر بوده است یا نه و ارزیابی سیستم در مقیاس بزرگ با ۲۵۷ منابع اختیاری در ۲۶ کشور انجام شده است. اما رویه های کاری BFTcloud همانطور که در شکل ۳ نمایش داده شده است دارای فازهای مختلف است.

### Byzantine Fault Tolerant Cloud



شکل ۳: رویه های کاری BFTcloud [۵].

نسخه پشتیبان اجرا می‌کند. این میان افزار سازگاری قوی نسخه‌ها را حفظ می‌کند و برنامه‌های کاربردی را با شفافیت ارائه می‌دهد و تاخیر انتها به انتها را کاهش می‌دهد. میان افزار با زبان برنامه‌نویسی ++C در سیستم عامل یونیکس پیاده‌سازی شده است [۶]. نتایج اندازه‌گیری تاخیر انتها به انتها برای نسخه Semi-active در شکل ۵ نشان داده شده است.



شکل ۵: نتایج اندازه‌گیری [۶].

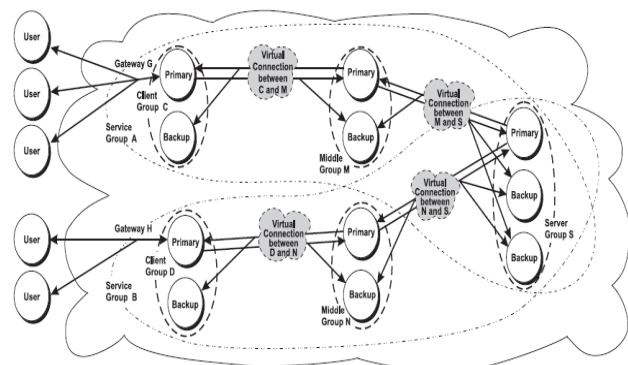
نمودار سمت چپ وابستگی تاخیر انتها به انتها روی اندازه پیام با تکرار نسخه و بدون تکرار نسخه را نشان می‌دهد. این میان افزار بخاطر طراحی خویش باعث می‌شود سیستم اشکالهای همزمان زیادی را تحمل کند که این نشان دهنده این است که میان افزار دارای مقیاس پذیری بالایی است همانطور که در نمودار سمت راست نشان داده شده است.

## ۲.۴. سیستم رایانش ابری Multiple Master با قابلیت تحمل پذیری اشکال

بیشتر سیستم‌های موجود بر مبنای معماری Single Master Multiple Slave هستند که از لحاظ مدیریت خطا و مقیاس‌پذیری دارای محدودیت هستند در این مقاله یک معماری جدید را ارائه می‌کند که Multiple Master Multiple Slave نام دارد که دارای مقیاس‌پذیری بالا و تحمل‌پذیر خطا می‌باشد و در این مقاله یک الگوریتم پویا برای ایجاد و مدیریت معماری MMMS ارائه شده است [۸]. در شکل ۶ معماری MMMS نشان داده شده است.

این مدل مقیاس‌پذیری سیستم را بهبود می‌بخشد و یک مدیریت اشکال کارا را فراهم می‌کند. این معماری همانند معماری SMMS دارای دو نوع نود می‌باشد نود Slave که نقش آن در این معماری همانند نود Slave در معماری SMMS است اما نقش نود Master در اینجا پیچیده است هر Master

فرآیندهای برنامه کاربردی را با استفاده از رویکرد تکرار leader/follower تکرار می‌کند. نسخه‌های یک فرآیند شامل گروهی از فرآیندها هستند که سرویس را برای کاربران فراهم می‌کنند که در شکل ۴ نشان داده شده است. در این گروه از فرآیندها یکی به عنوان اصلی و بقیه به عنوان پشتیبان در نظر گرفته شده‌اند نسخه اصلی در گروه فرآیندها پیام را به گروه فرآیند مقصد از طریق کانال ارتباطی مجازی می‌فرستد نسخه اصلی در گروه فرآیند مقصد پیامها را مرتب می‌کند و یک سری اعمال را اجرا می‌کند و اطلاعات مرتب شده را برای اعمال غیرقطعی که در نسخه پشتیبان ذخیره شده‌اند تولید می‌کند [۶].



شکل ۴: گروهی از سرویس‌های در تعامل با یکدیگر از طریق کانال مجازی در داخل یک ابر [۶].

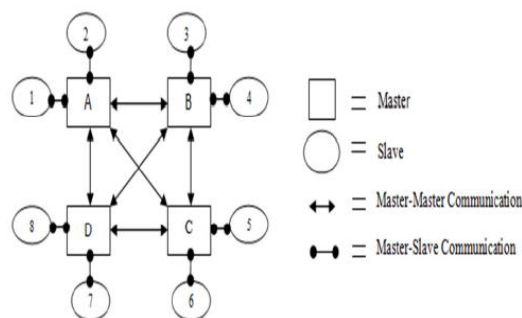
این میان افزار سازگاری قوی نسخه از تمام وضعیت نسخه‌ها هم در اعمال بدون خطا و هم در رخدادهای با خطا نگهداری می‌کند اگر خطایی رخ دهد مکانیسم‌های ترمیم یا پیکر بندی مجدد تضمین می‌کند نسخه پشتیبانی وجود دارد یا می‌تواند بدست آید و به پیامها و اطلاعات مرتب شده برای دوباره تولید کردن اعمال نسخه‌ای اصلی نیازمند آنها وضعیت را از نسخه موجود به نسخه جدید انتقال می‌دهند و اعمال نسخه جدید را با نسخه موجود همگام می‌کنند.

این میان افزار شامل پروتکل پیام با تاخیر پایین و پروتکل عضویت با رهبر تعیین شده (Leader-Determined) و یک تعیین کننده قالب مجازی (Virtual Determinizer) می‌باشد. پروتکل پیام یک سرویس دریافت پیام قابل اطمینان و با ترتیب کلی را با استفاده از چندپخش گروه به گروه که ترتیب آن توسط نسخه اصلی هر گروه مشخص می‌شود فراهم می‌کند. پروتکل عضویت پیکربندی سریع و سرویس ترمیم را وقتی که نسخه‌ای دچار خطا می‌شود یا نسخه‌ای وارد گروه یا از گروه خارج می‌شود انجام می‌دهد و تعیین کننده قالب مجازی اطلاعات مرتب شده را از نسخه اصلی گرفته و بهمان ترتیب در



- [4] P. Garraghan and P. Townend and J. Xu, "Byzantine Fault-Tolerance in Federated Cloud Computing" *Proceedings of The 6th IEEE International Symposium on Service Oriented System Engineering*, pp. 280-285, 2011.
- [5] Y. Zhang and Z. Zheng and M. Lyu, "A Byzantine Fault Tolerance Framework for Voluntary-Resource Cloud Computing" *IEEE 4th International Conference on Cloud Computing*, pp. 444-451, 2011.
- [6] W. Zhao and P. Melliar and L. Moser, "Fault Tolerance Middleware for Cloud Computing" *IEEE 3rd International Conference on Cloud Computing*, pp. 67-74, 2010.
- [7] G. Santos and L. Lung and C. Montez, "A Fault Tolerant Infrastructure for Web Services" *Ninth IEEE International EDOC Enterprise Computing Conferenc*, 2005.
- [8] M. Obaidat and H. Bedi and other, "Design and implementation of fault tolerant multiple master cloud computing system" *IEEE International Conferences on Internet of Things, and Cyber, Physical and Social Computing*, pp. 82-88, 2011.

مجموعه‌ای از Slave را به صورت انحصاری نگه می‌دارد که این مجموعه Sub-cluster نام دارد و سیاست زمانبندی هم خیلی پیچیده خواهد شد و الگوریتم زمانبندی بسته به طراحی سیستم می‌تواند توزیع شده یا محلی باشد [۸].



شکل ۶ معماری MMMS [۸].

### ۳. نتیجه‌گیری

در این مقاله به بررسی انواع تحمل پذیری اشکال در رایانش ابری پرداختیم. در رایانش ابری بلادرنگ، مدل ارائه شده طرح خوبی برای تحمل اشکال بوده و با استفاده از مکانیسم تصمیم سعی بر آن شده نودی با بالاترین قابلیت اطمینان انتخاب شود. در تحمل پذیری اشکال در ابر بهم پیوسته مدل ارائه شده مورد آزمایش واقعی قرار گرفته و مدلی ارائه شد که مبتنی بر گوناگونی بود. در رایانش ابری با تحمل پذیری اشکال بایزنتاین با منابع اختیاری برای بهبود قابلیت اطمینان از افزونگی استفاده شده است و برای سازگاری محیط ابری با منابع اختیاری سعی شده نودهایی با خصیصه‌هایی بالایی همچون قابلیت اطمینان و کارایی انتخاب شود. در میان افزار با قابلیت تحمل اشکال سازگاری قوی نسخه با استفاده از میان افزار بدون تغییر در برنامه کاربردی فراهم آمد. اندازه‌گیری کارایی نشان داد که این میان افزار با تاخیر کم سرویس دریافت پیام و سرویس‌های عضویت را تهیه کرده است. در سیستم رایانش ابری Multiple Slave مقیاس‌پذیری سیستم بهبود بخشیده شد و یک مدیریت اشکال کارا فراهم شد.

### ۴. مراجع

- [1] I. Foster and Y. Zhao and I. Raicu and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared".
- [2] S. Malik and F. Huet, "Adaptive Fault Tolerance in Real Time Cloud Computing" *IEEE World Congress on Services*, pp. 280-287, 2011.
- [3] S. Malik and M. Rehman, "Time Stamped Fault Tolerance in Real Time Systems".