# یک نظرسنجی در مورد سیستم های چندگانه موازی و توزیع شده برای شبیه سازی محاسبات با کارایی بالا

6.3 : پراکندگی: پراکندگی اساس مدل های موازی است . به عبارت دیگر پراکندگی به روی اجرا تاثیر مستقیم می گذارد، از یک طرف دیگر مدل های توسعه یافته ی پیچیده و از سوی دیگرعملکرد اجرایی است .

### D – MASON 6.3.1

از نظر پراکندگی ، D-MASON یک پلت فرم کلی بعنوان نماینده ایست که می تواند در موقعیت یک فضای دکارتی در نمایندگی محیط وجود داشته یانداشته باشد . محیط زیست باید به دو قسمت سلول ها و پارتیشن ها که به فرایند توزیع شبیه سازی شده اختصاص داده شده تقسیم شود . D-MASON سه روش را برای توزیع شبیه سازی بر روی چندین هسته پیشنهاد می دهد. دو روش آن مبتنی بر پراکندگی میدان یا پراکندگی شبکه است و سومین و آخرین روش آن مبتنی بر جداسازی شبکه است . مناطق همپوشانی ( مناطق مورد توجه یا منطقه ی AOI ) برای گارانتی متداوم در درک عامل از طریق گره های مرزی تعریف شده است .جداسازی میدان یا شبکه شکل 2 دو مکانیزم پراکندگی در دسترس را در D-MASON برای جداسازی شبکه ایی نشان می دهد ." پراکندگی "Y شامل تقسیم محیطی در سلول های افقی در محور Y است که در شکل 2 a نشان داده شده است . "پراکندگی "XY شامل تقسیم محیطی در سلول های مربع در محورهای X , Y مشخص شده است که در شکل b2 نشان داده شده است . شبکه ی محیطی توسط سلول های آن تقسیم می شود . و آنها را بر روی گره های پلت فرم اجرایی قرار می دهد . D-MASON در مناطق هم پوشانی که "مناطق مورد علاقه " (AOI)نامگذاری شده است استفاده می شوند (شکل 3) .این مکانیزم شامل کپی کردن بخشی از سلول های مجاور یا جداسازی مجاور است . مناطق همپوشانی فرصت هایی برای عامل فراهم میکندتا زمینه ی ادراک جهانی راحتی زمانیکه محیط از چند گره قطع شده است را برای حفظ تبادل اطلاعات هر سلول با سلول مجاور در هر زمان را

فراهم می کند . در مرحله ی پراکندگی فقط محیط برای بررسی شبیه سازی شده مورد بررسی قرار میگیرد . عامل ها در این مرحله به حساب نمی آیند . اگرچه تراکم آنها در مرحله ی تعادل بار به حساب می آید .

**جداسازی شبکه :** میدان شبکه ها راهی است برای نشان دادن ساختار گراف در شبیه سازی است . در گراف پراکندگی در چندین فرایند از چهار چوب هایی مانند ParMetis [51] یا Metis استفاده می شود . ParMetis [51] یک MPI مبتنی بر چهارچوب های موازی است که الگوریتم ها را برای بخش گراف غیر ساختاری ، Meshes , و غیره اجرا می کند ParMetis قابلیت گسترش عملکرد در Metis برای نمودارهای پراکنده و نمودارهایی با مقیاس بزرگ را دارد . در اینجا ما نتوانستیم به اطلاعات بیشتری در مورد مکانیزم جداسازی شبکه هایی که در پلت فرم های به روز شده ی اخیر اضافه شده است دست یابیم . لازم به ذکر است که در پلت فرم D-MASON می توانیم از انواع مختلف لایه ها در شبیه سازی استفاده کنیم . به عبارت دیگر ، پراکندگی فضایی می تواند در حالی که برای جداسازی شبکه ایی برای تعامل بین عوامل نشان داده شده است استفاده شود . در این حالت ، پراکندگی تنها براساس جداسازی میدان انجام می شود . برای توزیع شبیه سازی عوامل ، RepastHPC از یک مکانیزم بنام "Projection" استفاده می کنیم که از یک پلت فرم Repast S اقتباس شده است . این امر نشان دهنده ی آن است که عاملان در محیط تکامل پیدا می کنند . Projection ها بر دو نوع هستند : خواه می توانند میدنی یا شبکه ایی باشند . Projection ها برای اعمال ساختارها در عواملی که باعث تکامل آنها شده اند استفاده می شوند . مناطق همپوشانی (یا مناطق بهبود یافته ) برای مدیریت تداوم عوامل در مرز گره ها تعریف شده اند

**RepastHPC6.3.2** : پراکندگی در.RepastHPC شباهت بسیاری به خصوصیات DMASON دارد . Projection شبکه ایی تقریبا معادل جداسازی میدانی و Projection های شبکه ایست که معادل شبکه های جداسازی شده است . Projection های میدانی عوامل را در یک فضای دکارتی نشان می دهند . برای پراکندگی یک مدل بیش از چند پردازنده ، محیط به سلول های یکسان تقسیم می شوند . هر پردازنده مسئول قسمت زیرین شبکه است که به مناطق هم پوشانی متصل می شوند

شبکه ی پروجکشن ( Projectio ): شبکه ی پروجکشن عوامل را در فضای دکارتی نمایش می دهد . در پراکندگی یک مدل بیش از چند پردازنده ،در محیط به سلول های مساوی تقسیم می شوند . این سلول ها به طور مرتب در پردازنده ها توزیع می شوند . هر پردازنده مسئول قسمت زیرین شبکه است . قسمت های زیرین در شبکه با مناطق همپوشانی متصل می شوند . شکل 4 طرح پراکندگی در یک شبکه ی پروجکشن رادر 4 فرایند نشان می دهد . محدوده ی شبکه از مختصات (0،0) به مختصات (7و5) است . فرایند p1 مسولیت بخش –زیرین (0و0)در (2و3) است . p2 مسئول پاسخگویی به قسمت (3و5 )در (3و0 ) است . در مثال ما اندازه ی مناطق همپوشانی در 1 مشخص شده است . در این حالت p1 شامل یک منطقه ی بافر است که شامل کل ستون 3 در فرایند p2 در خط 4 در پردازنده ی p3 است .

پروجکشن شبکه : پروجکشن شبکه [28] روشی برای نمایش گراف ساختاری است . (شکل 5 ) نموداری از پروجکشن شبکه را با دو عامل پراکندگی در دو پردازنده نشان می دهد . به منظور تقسیم گراف در چند پردازنده در حالیکه ارتباط بین راس ها در پردازنده های مختلف حفظ میشود . یک کپی از لبه هاو راس های مجاور ساخته شده اند . متاسفانه هیچ اطلاعاتی در مورد RepastHPC که چطور گراف ها در چند پردازنده که به ندرت استفاده می شود موجود نیست. در مورد D-MASON که برای چند پروجکشن در مدل های یکسان استفاده می شود . به عنوان مثال ، یک پروجکشن شبکه ایی برای یک گرافیک محیطی و یک پروجکشن شبکه ایی برای تعامل بین عوامل است .

**6.3.3 برافروختگی :** عوامل پراکندگی در شبیه سازی برافروختگی از دو پراکندگی قبلی بعنوان عملکرد آماری[52] متفاوت تر است . این عامل پراکندگی در شبیه سازی های اولیه که در طول اجرا تغیر نکرده اند تعریف و محاسبه شده است . گزینه های پراکندگی مبتنی بر گراف ارتباطات بین عامل و بهینه سازی تولید شده ی فوقانی است که توسط ارتباطات پردازنده تولید شده است . دو روش را می توانیم برای شبیه سازی پراکندگی در برافروختگی پلت فرم ها استفاده کنیم .هندسی ( یا جداکننده ) و روند رابین . روش پراکندگی در ابتدای شبیه سازی شده از یک پارامتر استفاده می کند .

**توزیع هندسی :** این پراکندگی عامل ها را در سرتاسر فرایند مبتنی بر موقعیت آنها در فضای دکارتی جدا می کنند . این جداسازی مبتنی بر مختصات 2D یا 3 D است . توجه داشته باشید که در این حالت محیط شبکه برش را قطع نمی کندو هنوز هم با پراکندگی آنها بصورت مداوم و جداسازی در نظر گرفته می شود . هدف از این پراکندگی گروهبندی عوامل نزدیک به هم است که بطور بالقوه می توانند با هم ارتباط برقرار کنند و ارتباطات بسیاری تولید کنند .

**پراکندگی رویند رابین :**اگر عامل ها در یک فضای دکارتی قرار نگیرند ، برافروختگی یک مکانیزم پراکندگی را برای عامل هایی با روش رویند رابین ارائه می دهند . عامل هایی که یک به یک به هر پردازنده در روند اختصاص داده می شوند . این روش رفتار عامل ها را به حساب نمی آورد اما ممکن است یک تبعیض گر از نوع دیگر عامل را برای تاثیر در پراکندگی اضافه کند . به عنوان مثال اگر انواع عوامل با عاملی از نوع خودشان بیشتر از عاملی با نوع دیگر ارتباط برقرار کنند می توانند به یک یا چند بخش گروه بندی شوند.

### 6.3.4: پاندورا

پراکندگی پلت فرم پاندورا توسط تقسیم محیط در چند بخش مانند جداسازی شبکه درپلت فرم D-MASON و RepastHPC ساخته می شود . (شکل 6 الف) یک نمایش از پلت فرم پاندورا را نشان می دهد . ین نمودار نشان می دهد که پراکندگی از تقسیم محیط و همپوشی مناطق ساخته شده است که برای نگهداری حفظ تداوم محیطی حتی زمانیکه آنها در حال پراکندگی در چند فرایند هستند مورد استفاده قرار می گیرند .

توجه داشته باشید که بجز برافروختگی ، تمام سیتم عامل ها از یک محیط جغرافیایی استفاده می کنند تا عوامل به صورت پویا توزیع شوند . بنابراین شبیه سازی و نتایج آنها به این محدودیت پیوند خواهند خورد . از سوی دیگر برافروختگی تنها توزیع آماری در عوامل اولیه از پیکربندی را فراهم می کند و هیچ تعادلی در هدایت گره ها در طی راه ها ندارد .

**6.3.5: ارتباطات :** ارتباطات یک عملکرد کلیدی در خط اتصال پلت فرم برای روش های موازی است . به طور کلی این مورد بعنوان عامل یا پیام های نا همزمان در فراخوانی از راه دور پیشنهاد می شود .

**D-MASON 6.3.6**: ارتباط بین عوامل از پیام هایی که اجرا نشده است استفاده می کنند . ارتباط بین عامل ها بوسیله ی متقابل به عوامل هدف انجام می شود . این قابلیت فقط برای عوامل اجرا شده پردازش یا مناطق همپوشانی یکسان اجرا می شود . اگر هدف عامل ها اجرای پردازش یکسان در AOI نباشد ، برقراری ارتباط با استفاده از میدان محیط شبکه وجود ندارد . با این وجود با استفاده از میدان شبکه ، این محدودیت برطرف می شود . در حقیقت ،میدان شبکه به هر عاملی اجازه می دهد که زمانیکه پیوند یا لبه ایی بین آنها باشد با عامل دیگر ارتباط برقرار کند . علاوه بر این فرایند D-MASON پیشنهاد میکند که از پارامترهای جهانی برای چندین فرایند استفاده شود . این امر می تواند پشتیبانی مفیدی برای انتشار اطلاعات به همهی عوامل شبیه سازی شده باشد .

**RepastHPC6.3.7 :**ارتباطات از متن هایی که اجرا نشده اند استفاده می کنند .ارتباطات بین عوامل بوسیله ی روش هایی که در عوامل هدف وجود دارند انجام می شوند . اما این روش تنها برای عواملی که در یک پردازش یکسان وجود دارد در دسترس هستند . اگر عامل نامیده شده در یک پردازش یکسان وجود نداشته باشند ، عامل نامیده شده می تواند یک کپی درخواست کند . به این ترتیب ، فراخوانی روش ها در عامل های راه دور امکانپذیر است . به عبارت دیگر اگر یک نسخه عامل اصلاح شود ، تغییرات به عامل اولیه گزارش نمی شود . به روز رسانی عامل های کپی گرفته شده از عامل اصلی می تواند انسجام اطلاعات را حفظ کند .

### 6.3.8 برافروختگی

در برافروختگی یک عامل نمی تواند مستقیما پیامی به عوامل دیگر ارسال کند . ارتباطات با استفاده از برد پیام ها یا جداول پیام ها اجرا می شود . هر صاحب پردازنده یک برد پیام است که بین عوامل آن تقسیم شده است . عوامل می توانند ( دریافت کننده ) یا (ارسال کننده ) پیام ها در قسمت برد پیام ها با استفاده از برد پیام API باشد . عامل ها می توانند پیام هایی را از انواع مختلف فایل های XMML در شبیه سازی دریافت یا ارسال کنند اما نمی توانند پیام هایی را از همان حالت یا رفتار ارسال یا دریافت کنند . این عمل در دو حالت جداگانه اجرا می شود . تقسیم بندی یک شبیه سازی بر اساس پراکندگی برد پیام ها در شکل 7نشان داده شده است . عوامل می توانند می توانند بر روی پردازنده های مختلف اجرا شوند که وابسته به پراکندگی ها هستند اجرا شوند . افروختگی از پخش آنها

بعنوان یک روش ارتباطی به جای تلاش برای یافتن آنها تلاش می کند . با این وجود ممکن است تعداد گیرنده ها در هر پیام در سرتاسر فیلتر محدود باشد . در این مورد این پیام به سادگی برای یک گروه از عوامل پخش می شود . کتابخانه ی برد پیام یا لیبم برد برای مدیریت پراکندگی، ایجاد ، حذف و هماهنگی و در دسترس بودن پیام ها مورد نیاز است . کتابخانه ی لیم برد ، MPT را برای برقراری ارتباط بین فرایند ها استفاده می کند . و از موضوعات POSIX برای جابجایی مدیریت داده ها و ارتباط بین فرایند ها استفاده می کند

**6.3.9.پاندورا**

. ارتباطات با استفاده از MPI و اطلاعات دقیق تر با استفاده از کتابخانه ی µsik(55) استفاده می شوند . کتابخانه ی µsik مدیریت اطلاعات بین گره ها را طراحی می کند و این عمل بویژه برای اجری شبیه سازی رویداد گسسته بر روی منابع موازی طراحی شده است . ارتباطات با استفاده ازMPI و پلت فرم پاندورا به طور خودکار کدها را برای استفاده از ارسال کد MPI خود داری می کنند . با این وجود، پلت فرم پاندورا ارتباطات میان عامل را محدود می کند: عوامل تنها می توانند در مفهوم میدان ، ارتباط برقرار کنند .

**6.4هماهنگی /هماهنگ سازی :** هماهنگ سازی بین چهار پلت فرم متفاوت است : هرکدام از مسائل هماهنگی راهی است برای ارائه و توسعه ی شفافیت های لازم .

**D-MASON 6.4.1 :** اجرای پلت فرمD-MASON یک هماهنگ سازی محافظه کارانه است که خطا های علیتی را تضمین می کند . برای دستیابی به این هماهنگ سازی محافظه کارانه است . که هر مرحله از شبیه سازی به دو مرحله تقسیم می شود . (1)ارتباطات / هماهنگ سازی و اجرای شبیه سازی :در پایان هر مرحله یک مانع هماهنگ سازی وجود دارد . عامل در سلول i تا زمانیکه شبیه سازی خود را با سلول های مجاورi-1 به پایان نرسانیده است قابل اجرا نیست . سپس هر سلول اطلاعاتی را برای سلول های مجاور در مورد عواملی که در AOI یا سلول های مهاجر میفرستند . برای مرتبه ی i در رفتار سلول C براساس داده های مرحله ی i-1 سلول های مجاور محاسبه می شود . پلت فرم D-MASON ، استراتژی ارتباطات مختلف برای هماهنگ سازی مبتنی بر فعالیت های MQ و MPI [56] است . در نسخه های اولیه D-MASON از فعالیت MQ JMS با استفاده از

ناشر / مشترک استفاده می کند . اگرچه این نمونه توسط MPI پشتیبانی نمی شود . و نویسنده در سراسر MPI آن را اجرا می کند . . در پلت فرم D-MASON ، فرایند MPI به کارگران D-MASON اختصاص داده می شوند و آنها از (گروه ارتباطی ) استفاده می کنند . موردی که ارتباطات درون سلول ها را توصیف می کند . مدل هماهنگ سازی پنج مورد ادعایی را مورد استفاده قرار می دهد .

- هر سلول نوع خود را برای بروز رسانی منتشر می کند

- هر سلول حداقل در یک موضوع ( حداقل با یک سلول مجاور ) مشترک می شود . یک سلول نمی تواند یک گام شبیه سازی جدید را تا زمانی که به روز رسانی شود را از همه ی موضوعاتی که با آن اشتراک دارد را شروع کند .

-اشتراک یک استاتیک است : هر سلول اشتراک گذاری مسیر را قبلاز شروع شبیه سازی به اشتراک می گذارد .

-انتشار در مورد موضوعی که چندین میدان دارد مثلا( در سلول های دیگر ) مشترک هستند و می توانند زمانی اتفاق بیفتند که همه ی سلول های به روز شده در هر میدانی در دسترس باشند .

- انتشار فقط در پایان مرحله ی شبیه سازی شده مورد استفاده قرار می گیرد . علاوه بر 5 مدل بالا ، در اینجا 3 مدل دیگر از استراتژی ارتباطات برای هماهنگ سازی روش معمول MPIوجود دارد : BCast ، که پخش را مورد استفاده قرار می دهدو جمع آوری که برای بدست آوردن اطلاعات استفاده میشود و مورد آخر موازی است که در یک فرایند همگام سازی برای ارسال و دریافت پیام ها مورد استفاده قرار می گیرد .

**RepastHPC6.4.2 :** هماهنگ سازی بین پردازنده ی پلت فرم RepastHPC مطابق با چهار مرحله اجرا می شود . [29]. اول اینکه زمانیکه فرایند ها نیازمند یک یا چند کپی از فرایند های دیگر هستند این هماهنگ سازی نیازمند ادامه به شبیه سازی در حالت سازگار است . دوم اینکه ، زمانیکه یک پردازشگر عامل غیر محلی یا لبه از پردازنده ی دیگری است سپس کپی از آنها باید به روز و از آخرین اطلاعات شی اصلی باشد . سوم اینکه مناطق همپوشانی در میدان باید در هر مرحله از زمانبه روز شود . و در آخر هنگامیکه عامل به طور کامل از یک پردازنده به پردازنده ی دیگر مهاجرت می کند باید به روز شود . به عنوان مثال زمانیکه یک عامل به محیط دیگر انتقال می یابد

.

اکثر مکانیزم های هماهنگ سازی مسول برنامه نویسی نیستند . برنامه نویسان فقط بسته های الگویی را انتقال داده که اطلاعات ضروری را در هماهنگ سازی عوامل شناسایی می کند . الگو متشکل از دو وش ارائه دهنده است ، استفاده ازسریال داده ها قبل از ارسال و دریافت ، استفاده از سریال داده ها در زمان دریافت ، این دو روش راهی برای سریال سازی اطلاعات عامل به منظور هماهنگ سازی داده های مورد نیاز زمانی که یک عامل از یک فرایند به فرایند دیگر انتقال می یابد است.

**6.4.3 برافروختگی** : هماهنگ سازی بین پردازنده های پلت فرم افروختگی به صورت محافظه کارانه است و این عملکرد از طریق MessageBoards انجام میشود . در واقع تمام مبادله ی اطلاعات توسط پیام هایی که متکی هماهنگ سازی در برد پیام های هماهنگ شده اجرا می شود . عملی که به نوبه ی خود متکی به پخش پیام ها برای شرکت در فرایند های شبیه سازی شده است . به این ترتیب ، تمام پردازنده ها نمای یکپارچه ایی از شبیه سازی دارند . هماهنگ برد پیام ها در دو مرحله انجام می شود . که اول شامل درخواست برای هماهنگ سازی و سپس اجرای آن است .مرحله ی اول ، زمانیکه یک پردازنده اجرای یک عامل ر به پایان می رساند بردهای پیام خود را قفل میکند و درخواست هماهنگ سازی را در یک صف می فرستد . بعد از این مرحله هنوز می توان اقداماتی را انجام داد که نیازی به برد پیام نباشد . زمانیکه همه ی فرایند ها برد پیام خود را قفل می کنند ، مرحله ی بعد انجام می شود . در مرحله ی دوم هماهنگ سازی پیام ها بین برد پیام ها مبادله می شود . مرحله ی هماهنگ سازی یک مرحله ی بلوکه شده است . پس از این دو مرحله برد پیام ها باز شده و شبیه سازی ادامه می یابد .

**6.4.4 پاندورا** : هماهنگ سازی در پاندورا مانند پلت فرم RepastHPC و D-MASON محافظه کارانه است . داده ها و عوامل در مناطق همپوشانی کپی شده و به عوامل مجاوری که در هر زمان داده ها را به روز رسانی می کنند فرستاده می شود . اندازه ی منطقه ی همپوشانی به اندازه ی حداکثری میدان درک شده در هر عامل است . [57,54]. برای حل مشکل همگام سازی در مناطق همپوشانی بین فرایند ها و حل اختلافات و و همچنین مشکلات بین فرایند ی مختلف عوامل و پلت فرم پاندورا ها از روش های مختلف پلت فرم های دیگر استفاده می کنیم .هر

بخش محیطی که در میان چندین پردازشگر توزیع شده است نیز در چهار زیر مجموعه ی برابر از 0 تا 3 شماره گذاری شده است . شکل 6

در طول مرحله ی اجرا ی عوامل همه ی فپردازنده ها هر کدام از بخش های زیر ر به طور جداگانه اجرا می کنند . به عبارت دیگر همه ی پردازنده ها زیر پخش 0 را اجرا می کنند بعد ازآن زیر بخش 1 و همانطور به ترتیب . به عبارت دیگر ، از انجایی که تمام قسمت های زیر بخش نزدیک نیستند در اینجا امکان هیچگونه تضادی وجود ندارد.پس از پایان یک بخش، تغییرات منطقه ی همپوشانی به مناطق دیگر ارسال می شود . هنگامی که تمام قسمت های زیر اجرا شوند کل حال شبیه سازی دنبال می شود و یک مرحله ی جدید زمانی دنبال می شود .

**6.4.5 تعادل بار :** ما در اینجا اطلاعاتی را که در رابطه با تعادل بار در سیستم عامل مورد آزمایش یافتیم ارائه می دهیم **6.4.6 D-MASON :** تعادل باردر پلت فرم D-MASON به صورت پویا انجام می شود . [58] روش پیشنهادی که می تواند در محیط 3D , 2D استفاده شود . تعادل بار بعنوان یک گام اضافی در مرحله ی زمانی علاوه بر هماهنگ سازی و شبیه سازی اجرا می شود . در پایان هر مرحله ی زمانی ، چگالی موجود در هر سلول و محیط محاسبه می شود . عامل ها در گره ی چگالی بالا برای تعادل بار به گره های مجاور ارسال می شوند .

**6.4.7برافروختگی :** در اینجا هیچگونه تعادل باری در پلت فرم flame وجود ندارد . تعادل بار در ابتدای شبیه سازی با توزیع پیشنهادی به صورت ایستا انجام می شود . به یاد داشته باشید که مارکیوز و همکاران در [44] پیشنهاد داده اند که طرح تعادل باری برای پلت فرم flam است که هنوز در رها سازی اصلی اجرا نشده است .

**6.4.8 RepastHPC:** ما هیچگونه اطلاعاتی در مورد مکانیزم اجرایی پلت فرم RepastHPC برای اجرای پویای تعادل بار مطالعه نکردیم . بنابراین این ویژگی در اینجا بیان نمی شود .

**6.4.9 پاندورا :** اگر چه پلت فرم پاندورا در پروژه های مختلفی مانند سیمی پست استفاده می شوند [59] اما در اینجا هیچگونه اطلاع صریحی وجود ندارد . اجرای پلت فرم ها و مکانیزم های استفاده شده به درستی مستند نشده است. همچنین ما هیچگونه اطلاعاتی در مورد نحوه ی تعادل بار پیدا نکردیم .

**6.5: ترکیبات خواص موازی :** جدول 4 خواص چهار پشتیبانی موازی سیستم عامل را ارزیابی میکند .

**7 ارزیابی عملکرد :** هدف از مدل موازی سازی ، بدست آوردن اجرای بهتر یا بزرگتر است . یا بدست آوردن مدل های دقیق تر که برای حافظه ی یک ماشین ، بزرگ و مناسب هستند . مساله ی اندازه ی حافظه با توجه به اضافه کردن ماشین های بزرگتر مورد بررسی قرار گرفته است . این حافظه به مقدار جهانی اضافه شده است . مساله ی عملکرد با دادن هسته ی بیشتر برای شبیه سازی مورد بررسی قرار گرفته است . برای اثبات کارایی PDMAS در بررسی تجربه ی این دو مشکل باید زمانیکه مدل های بزرگتر در هسته های بیشتر اجرا می شوند مورد استفاده قرار بگیرند . سپس ما مدل مرجع مان را در چهار PDMAS انتخاب شده ی قبلی اجرا می کنیم . ما در این بخش به بررسی نتایج عملکرد و مشاهدات مان در هنگام استفاده از منبع HPC مانند خوشه ی HPC می پردازیم .

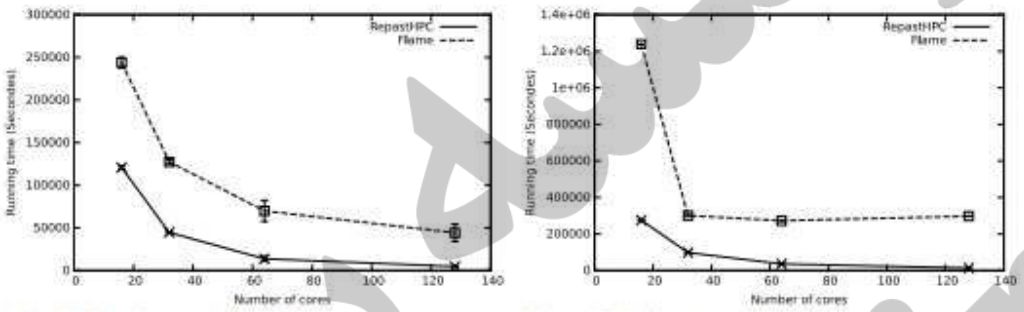
**7.1 تنظیمات آزمایشی :** برای ارزیابی عملکرد باید مدل مرجع را در بخش 6.2 در چهار سیستم عامل کاربردی RepastHPC ، D-MASON،Flame وPandora(پاندورا) اجرا کنیم . در طی اجرای این مدل ما مشکلات جدی برخورد نکردیم . توجه داشته باشید که با لطف پشتیبانی PDMAS این مدل هر گونه اندازه ی پلت فرم های اجرایی را تغییر نمی دهد . ما مدل مرجع یک هسته ی خوشه ای 1280 را با استفاده از سیستم SGE batch اجر کردیم .هر گره موجود در خوشه یک دو پردازنده است ، با اجرای پردازش Xeon E5 (8*2 cores) در فرکانس 2.6 Ghz و با حافظه ی 32 Go . گره هااز طریق یک شبکه ی بی نهایت DDR infinyBand به یک درخت چربی متصل می شوند . سیستم و به ویژه شبکه با کاربران دیگر به اشتراک گذاشته شده است . سیستم دسته ایی تضمین می کند که هر فرایند بدون در نظر گرفتن سایر فرایندها بر روی هسته ی خود اجرا می شود . ( بدون اشتراک گذاری زمان ) . توجه داشته باشید که پیکربندی رایج برای اجرای موازی عملکرد است زیرا HPC در محاسبه ی مرکزی با کاربران دیگر به اشتراک گذاشته می شود . هنگام ارسال یک کار تعداد هسته های درخواست شده با توجه به زمان بندی ارسال می شوند . . این گزینه بدان معنی است که برنامه ی زمانبندی تلاش می کند که حداکثر فرایند ها را در گه های مختلف ارسال کند . این تنظیمات آزمایشی ممکن است تغییرات اندکی را در زمان اجرا ایجاد کنند بنابراین ما نتیجه ی این عملکرد را در محاسبات در نظر می گیریم . تغییرات مشاهده شده

کم است کمتر از 0.21٪(با برافروختگی در 128 هسته ) به همین دلیل هر اندازه گیری نیازمند زمان است و ما میانگین آن را 10 بار گزارش می کنیم . ( قطعه ی جعبه ها اطلاعات بیشتری را در اختیار ما قرار نمی دهند .)

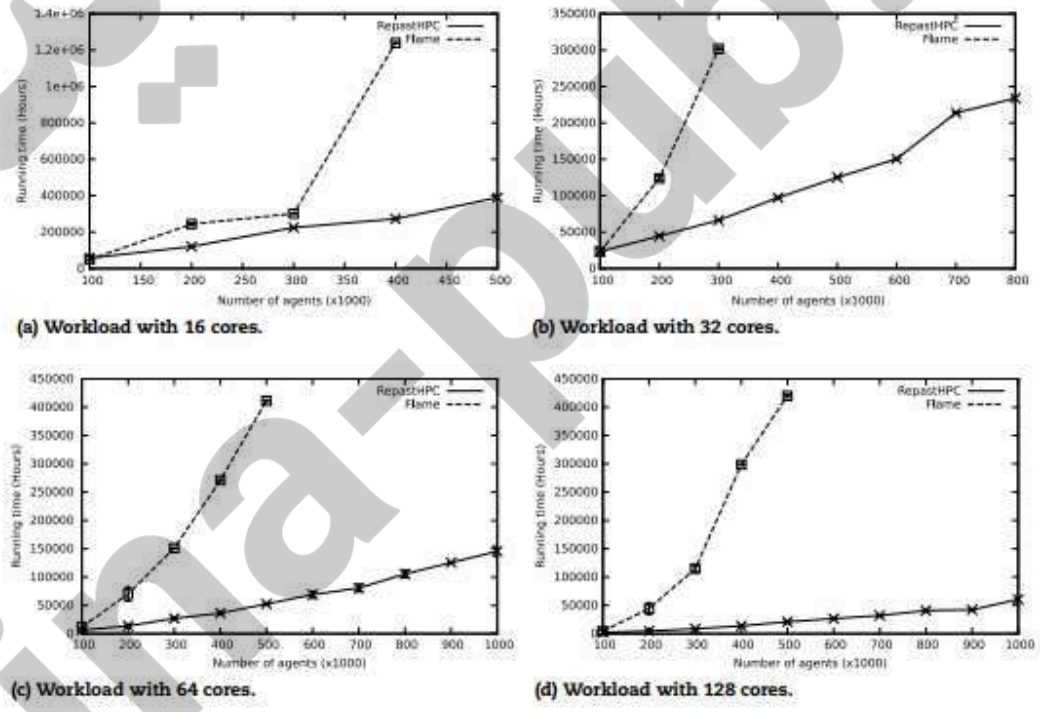
**7.2 نتایج عملکرد :** اگرچه ما توانا به اجرای چهار عملکرد D-MASON, Flame, Pandora, RepastHPC در ایستگاه کاری استاندارد بودیم ، اما تنها سه مورد از آنها RepastHPC, Flam و D-MASON در سیستم HPC با موفقیت اجرا شدند . شبیه سازی Pandora ,(پاندورا) حتی اگر ما از مثال هایی که در پلت فرم زده شد استفاده کنیم باز دچار مشکلاتی است . و ما نمیتوانیم نتایج بدست آمده را با عملکرد خودمان مقایسه کنیم . هنگام تجزیه و تحلیل اجرای شبیه سازی D-MASON، متوجه شدیم که پلت فرم از موضوعات شبیه سزی شده ی هر پردازشگر استفاده می کند که ما نمی توانیم این موضوعات را در طی اجرای شبیه سازی کنترل کنیم . برنامه ریزان Batch بعنوان SGE و SLURM در محاسبات مرکزی استفاده می شوند . با این حال فرض می کنیم که تنها یک موضوع در هر دسته ی اختصاص داده شده اجرا می شود . در غیر این صورت موضوعات اضافی ممکن است از استفاده شود که به سایر مشاغل اختصاص داده شود . اگرچه D-MASON محدودیتHPC را رعایت نمی کند و نمی تواند با پلت فرم RepastHPC وFlame مقایسه شود . این اعداد یک رشته را با اعداد چند رشته ایی مقایسه می کنند که درست نیست . از سوی دیگر ما متوجه شده ایم که DMASON عملکرد خوبی را تولید می کند . این احتمالا یک پلت فرم جالب PDMAS از پلت فرم های که به سطح بلوغ اولیه ملحق خواهند شد بدست آمده است . ( هنوز در اینجا برخی از مشکلات و برخی از محدودیت ها در تعادل بارگذاری را می توان بهبود بخشید ) . به همین دلیل نتایج ارائه شده تنها flam(برافروختگی ) و پلت فرم RepastHPC را در نظر می گیرد. در (شکل 8) بسیاری از پلت فرم های مقیاس پذیر و در (شکل 9) بار دستگاه و در(شکل 10 ) مصرف حافظه را نشان داده ایم . برای ارزیابی تعداد گره های متنوع مورد استفاده برای اجرای شبیه سازی ها در حالیکه تعدادی از عوامل رارفع می کنیم . برای بار دستگاه ما تعداد گره های 16 ، 32 ، 64 و 128 را ثابت و تعداد عامل های شبیه سازی شده را تغییر دادیم . . نتایج مقیاس مدل ها را با 200،000 و 400،000 عامل را در شکل 8 نشان داده ایم . توجه داشته باشید که اولین اجرا با 16 هسته انجام شد . ما تعداد 16 هسته را انتخاب کردیم چونکه اکثر ایستگاه های کار از 8

هسته برای پردازش استفاده کرده بودند . بنابراین این روش ممکن است باعث نمایش قدرت دو پردازش گر در دسک تاپ شود . ما توجه داشتیم که کقیاس هر دو پلت فرم بالای 64 هسته است اما عملکرد آنها نسبت به زمانیکه هسته ی بیشتری استفاده می شود بهبودی نمی یابد . نتایج بدست آمده از RepastHPC بهتر از نتایج پلت فرم flame است : برای 16 هسته RepastHPC در حدود 2،02 بار بهتر از flame ها هستند اگر چه برای 128 هسته اختلافی بهتر برابر 9،26 بار دارند . این اختلافات می تواند توسط استراتژی استفاده شده در پلت فرم ارتباط عوامل استفاده شود . مرجع این مدل ها ارتباط مدل ها و flame هایی است که اطلاعات و ارتباطات هماهنگ سازی را بخش می کنند . و این احتمال وجود دارد که وقت زیادی را صرف جابجایی اطلاعات کنند . علاوه بر این ما بر روی این نکته توجه داریم که 400،000 عامل در مقیاس پلت فرم flame بهتر از 64 هسته است . اما هسته های بیشتر منجر به کند شدن عملکرد می شوند . این زیر – عملکرد همچنین می تواند توسط اجرای عملکرد در flame ها توضیح داده شود . شکل 9 عملکرد دستگاه بار در دو پلت با مقدار 16 ، 32 ، 64 و 128 هسته را نشان می دهد . افزایش بار توسط مکان داخلی بار عوامل ( توسط خوشه های DFT تعمیم می یابد ) و با افزایش تعداد عوامل شبیه سازی شده بدست می آید . مقدار DFT خوشه ایی در اینجا 128 تعیین شده است . توجه داشته باشید که قطعات نتایج را برای همهی ارزش ها در محور X نشان نمی دهند . و این امر به خاطر این است که پلت فرم ها نمی توانند با مقدار زیاد عوامل در این تعداد از هسته اجرا شوند . ما می بینیم که flame ها مقیاسی بهتر ازRepastHPCدر نقاط کوچکی که قطعه ای در شکل هستند ندارند . به عنوان نمونه ، برای 128 هسته ما نمی توانیم شبیه سازی را تا بیشتر از 500،000 عامل اجرا کنیم . شکل 9 نشان می دهد که RepastHPCبه طور واقعی واکنش بار را از Flame بهبود می بخشد . اگر چه در شکل 9 برای 16 هسته مشخص شد که پلت فرم flame نتیجه ی بهتری از RepastHPC برای 100000 عامل در همان تعداد هسته بدست آورد . بدیهی است که مدل ها از همه ی قدرت مورد استفاده در flame استفده نمی کنند و این محدود به اصطلاحات ارتباطات درون عاملی است . سوال جواب ها این است که : آیا این به خاطر استفاده از مفاهیمی مانند X-Machines و یا هماهنگ سازی مکانیزم است . ؟ یکی دیگر از دلایل ممکن است هزینه ی هماهنگ سازی ارائه شده توسط flame

هنگام استفاده از عوامل از راه دور باشد که توسط RepastHPC مدیریت نشده است . آخرین شکل ، شکل شماره ی 10 است که نشان دهنده ی مصرف حافظه در سیستم عامل شبیه ساز بیشتر از 16 یا 128 هسته است . در شکل 10 برای 16 هسته می توانیم از flame هایی با حافظه ی دوبار کمتر از حافظه نسبت به RepastHPC استفاده کرده است .




**شکل 8 - مقیاس پذیری پلت فرم از 16 تا 128 هسته.**



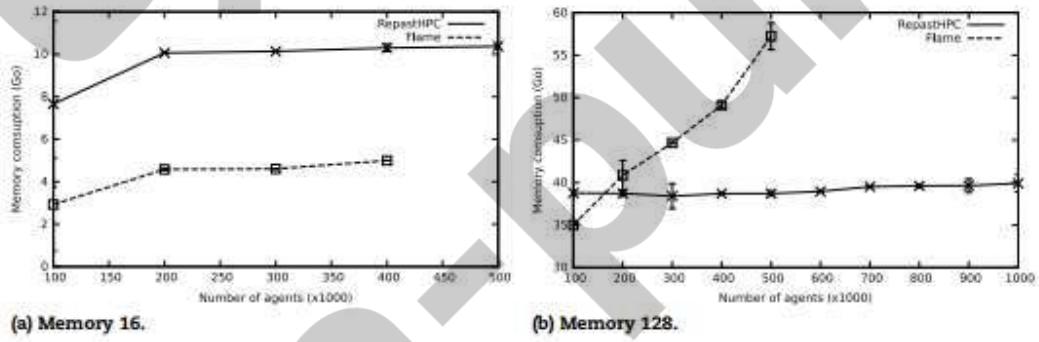

شکل.9_ بار کاری پلت فرم برای نتیجه محاسبه DFT برای 128تایی در هر عامل

در مقابل، شکافی در مصرف حافظه بین دو پلت فرم برای 128 هسته وجود دارد. Flame در واقع سریع تر افزایش می یابد در حالی که RepastHPC تقریبا ثابت است. Flame بیش از Go 57 از حافظه را برای اجرای یک شبیه سازی در بیش از 128 هسته، بکار میگیرد در حالی که Rapast HPC در حدود Go 39 از حافظه برای شبیه سازی مشابه استفاده می کند. این نتایج را می توان با پیاده سازی درونی عامل ها در پلت فرم و همچنین نحوه مدیریت ارتباطات و هماهنگ سازی ها در طول شبیه سازی توضیح داد.

شایان ذکر است که RepastHPC تنها، بدون هیچ گونه مشکلی و بدون مصرف حافظه ی زیادی به 1 میلیون عامل دسترسی پیدا کرده است.

## 8. ترکیب

در این بخش بازخورد تجربه ما در مورد پلت فرم های مورد آزمایش ارائه می شود. این تجربه در طی تحقیق کتابشناسی، کشف پلت فرم ها، پیاده سازی مدل های مختلف و آزمایش ها به دست آمد. این کار در طی حدود نیم سال کار انباشته در طول دو سال گذشته و بیش از 250.000 ساعت محاسبات ارایه شده است.برداشت و نظرات غیررسمی ما در مورد نقاط مثبت و منفی هر پلت فرم در زیر خلاصه شده است.




شکل.10_مصرف حافظه پلت فرم ها برای 16 و 128 هسته و 128DFTتایی

### RepastHPC 8.1

RepastHPC یک پلت فرم جامع است که استفاده از ساختارهای مختلف (شبکه ها، توری ها) را امکان پذیر می سازد. این ساختارها را می توان در یک شبیه سازی یکسان برای نشان دادن انواع مختلف تعاملات جمع آوری کرد. توسعه و ایجاد کار توسط تعدادی خود آموز تسهیل می شود که نحوه استفاده از RepastHPC را توضیح می دهد.

این آموزشها واضح و با مراحل مختلفی شرح داده شده .RepastHPC دو روش برای پیاده سازی شبیه سازی های چند عامله ارائه می دهد: ReLogo و++ C. استفاده از ReLogo برای ایجاد یک مدل بسیار ساده است اما از بعضی از ویژگی های RepastHPC سود نمی برد. از سوی دیگر، بکار گیری ++ C این امکان را می دهد تا شبیه سازی های پیچیده تر با موانع را اجرا کنیم که مهارت های خوبی در ++ C (مفهوم قالب ها) مورد نیاز است.

با این وجود برخی از محدودیت ها در استفاده از پلت فرم وجود دارد، زیرا هیچ ارتباطی بین عامل های راه دور مجاز نیست. این فقدان از اجرای بعضی از مدل ها، حتی ساده ترین آن ها، که نیاز به اصلاحات عامل ها بصورت از راه دور را دارند جلوگیری می کند. از سوی دیگر، RepastHPC راندمان و مقیاس پذیری بالایی برای مدل های فقط خواندنی را ارائه می دهد.

پس از اجرای مدل روی پلت فرم Repast-HPC، بازخورد این پلت فرم را سنتز کرده ایم. در جدول 5 استدلال های موافق و مخالف داده شده اند.

### Flame 8.2

Flame یک پلت فرم جالب برای ایجاد مدل های موازی است زیرا در مقایسه با پلت فرم های دیگر مانند ماشین های X یا گراف های حالت، از یک رویکرد متفاوت استفاده میکند. اگر چه این رویکرد به معنای یادگیری یک زبان برنامه نویسی جدید است، اما به نظر می رسد که اجرای شبیه سازی در Flame بیش از سایر پلت فرم ها پیچیده تر نیست. علاوه بر این، بسیاری از نمونه های ارائه شده که یادگیری نحوه پیاده سازی یک شبیه سازی را امکان پذیر می سازد از زبان XMML استفاده می کند. استفاده از X-Machine این امکان را به توسعه دهنده می دهد تا مسائل مربوط به توزیع را پنهان کند و شبیه سازی ها را بصورت شفاف روی رایانه های موازی اجر کند. بر خلاف RepastHPC، پلت فرم Flame ، اصلاح عامل ها را بصورت از راه دور با استفاده از پیام ها و Message-Boards امکان پذیر می سازد.هزینه بدست آوردن این قابلیت، کارایی کمتر، حتی برای مدل های فقط خواندنی است. دو گزینه در طراحی Flame که منجر به کارایی کمتر شده ، فقدان بهینه سازی منطقه همپوشانی و

استفاده سازمان یافته از ارتباطات پخش همگانی می باشد. توجه داشته باشید که یک نسخه جدید از پلتفرم Flame که مسئله عملکرد را حل می کند اعلام شده است [60] اما در این زمان هنوز در دسترس نیست.

جدول 6 خلاصه ای از استدلال های مخالف و موافق برای ایجاد یک مدل با پلت فرم Flame را ارائه می دهد.

### D-MASON 8.3

در مقایسه با RepastHPC و Flame ، D-Mason پلت فرم جدیدتری است و اولین اجرای آن شبکه ای از ایستگاههای کاری را با میل بیشتری نسبت به کامپیوترهای HPC هدف قرار داد. با این حال، این روش یک پلت فرم کاملا برجسته، با پشتیبانی از روش های مختلف ارتباط است: ActiveMQ یا MPI.

شبیه سازی با D-MASON بسیار آسان است، تمام روش های اولیه و اساسی در حال اجرا هستند. برای ایجاد یک مدل نیاز به مهارت و دانش خاصی در جاوا نداریم. علاوه بر این، استفاده از Maven کامپایل کردن شبیه سازی ها را آسان می سازد. D-MASON یک پلت فرم بسیار عالی است اما در محیط خوشه ای HPC امکان فقدان بلوغ وجود دارد. همانطور که قبلا گفته شد اجرای یک نسخه با ثبات از پلت فرم مبتنی بر MPI با مدل مرجع موفقیت آمیز نبود. توجه داشته باشید که همانند RepastHPC، در D-MASON اصلاح عامل ها بصورت از راه دور امکان پذیر نیست و تنها به مدل های فقط خواندنی محدود می شود.

D-MASON همراه با مستنداتی است که چگونگی تبدیل یک مدل MASON متمرکز را به مدل DMASON موازی ، توضیح می دهد. همچنین بسیاری از مدل های نمونه ای وجود دارد که به توسعه دهنده در درک قابلیت های پلت فرم کمک می کند.

پس از پیاده سازی مدل در پلت فرم D-MASON بازخوردها با استدلال های موافق و مخالف سنتز شده اند که در جدول 7 نشان داده شده است.

### Pandora 8.4

همانند پلت فرم های قبلی، Pandora یک پلت فرم کامل برای اجرای مدل های موازی است. یک قابلیت بسیار قابل توجه که توسط پلت فرم ارائه شده است تولید خودکار کد موازی است. ما موفق نشدیم یک مدل Pandora را بر روی یک خوشه اجرا کنیم، اما آن را روی دسکتاپ مستقل انجام دادیم. با این وجود، مستندات یا آموزش بیشتر

می تواند جایگزین مناسبی بجای خواندن نمونه های کد برای درک نحوه شبیه سازی شود. Pandora همچنین ابتدا GIS را ادغام می کند و کد شبیه سازی C ++ برای نوشتن بسیار ساده است. Pandora ارتباطات از راه دور را فراهم نمی سازد، اما به لطف مدل هماهنگ سازی چرخشی آن، مشکل مدل خواندنی _ نوشتنی را حل می کند.بهر حال این مدل هماهنگ سازی، محیط مدل را به توری های دوبعدی محدود می کند و تا آنجا که می دانیم انواع محیط مانند شبکه ها را پشتیبانی نمی کند. جدول 8 خلاصه ای از استدلال های مخالف و موافق برای ایجاد مدل با پلت فرم Pandora را ارایه داده است.

| جدول 5 – استدلال های موافق و مخالف برای استفاده از پلاتفرم Rapast-HPC. | |
|---|---|
| جوانب مثبت | جوانب منفی |
| • مستندات ، مثال و خودآموز های جامع<br>• سهولت پیکربندی و زمانبندی انعطاف پذیر: به راحتی قابل تنظیم است حتی برای انجام زمانبندی های پیچیده<br>• نیاز به دانش خاصی درباره MPI ندارد<br>• جامعه بزرگ و بسیار پاسخگو<br>• پیکربندی و اجرای آسان بر روی خوشه ها<br>• به روز شدن به طور منظم | • تمام موارد ارتباطات بین عامل ها را شامل نمی شود<br>• هنگامی که یک عامل کپی شده اصلاح می شود اطلاعات با عامل اصلی همگام سازی نمی شود. |

| جدول 6 - استدلال های موافق و مخالف برای ایجادبا استفاده از پلتفرم Flame | |
|---|---|
| جوانب مثبت | جوانب منفی |
| • ایجاد آسان یکبار هنگامی که گرامرXML شناخته شده است.<br>• تمرکز بر گذار حالت وضعیت<br>• شبیه سازی گرافیکی بعد از اجرا با مرورگر Flame (اختیاری اما استقبال می شود) | • دشواری در اعتبارسنجی و درک مدل شبیه سازی XMML<br>• نمی تواند در یک وضعیت پیام ها را هم ارسال و هم دریافت کند<br>• ارتباطات نقطه به نقطه ندارد، تمامی ارتباطات باید |

| | |
|---|---|
| • فرا مدل XMML برای اعتبار سنجی شبیه سازی<br>• تولید اتوماتیک کد موازی<br>• جابجایی بین اجراهای ترتیبی و توزیع شده اسان است | از طریقMessageBoard انجام شود. |

| جدول 7 - سنتز استفاده از پلت فرم D-MASON. | |
|---|---|
| جوانب مثبت | جوانب منفی |
| • GUI ساده و بصری<br>• تسهیل کامپایل کردن: Maven<br>• جامعه پاسخگو<br>• به روز شدن به طور منظم<br>• دو لایه ارتباطات پیشنهاد شده (MPI یا ActiveMQ JMS)<br>• سه راه برای همگام سازی با استفاده از MPI (BCast، Gather و موازی) پیشنهاد شده | • تمام موارد ارتباطات بین عامل ها را شامل نمی شود |

| جدول 8– استدلال های موافق و مخالف برای ایجاد با استفاده از پلتفرم Pandora. | |
|---|---|
| جوانب مثبت | جوانب منفی |
| • تولید کد موازی و ترتیبی<br>• پشتیبانی GIS<br>• نمونه سازی سریع<br>• جابجایی بین اجراهای ترتیبی و توزیع شده اسان است | • مستندات ضعیف<br>• تمام موارد ارتباطات بین عامل ها را شامل نمی شود<br>• تمام موارد همگام سازی بین عامل ها را شامل نمی شود |

| | |
|---|---|
| • نیاز به دانش خاصی در مورد MPI ندارد.<br>• ابزارهای تجزیه و تحلیل را پیشنهاد می دهد (Cassandra) | |

**9. نتیجه گیری**

در این مقاله مقایسه هایی درباره چندین پلت فرم چند عامله موازی ارائه شده است. این مقایسه در دو سطح انجام می شود، ابتدا در سطح کیفی با استفاده از معیارهای پشتیبانی و دوم در سطح کمی، با استفاده از پیاده سازی مدل مرجع چند عامله و ارزیابی عملکرد اجرایی ارائه شده است. تجزیه و تحلیل کیفی بر ده پلت فرمی متمرکز است که در طی جستجوی کتابشناسی یافت شد. مقایسه ویژگی های پلتفرم های مورد مطالعه را در رابطه با ایجاد و اجرای موجودیت های عامل ، ضمانت های ارائه شده و پشتیبانی موازی سازی را نشان می دهد.

سپس چهار پلت فرم را در نظر می گیریم ، آنهایی که آزادانه در دسترس و کاربردی هستند. برای ارزیابی دقیق تر، ما یک مدل مرجع را اجرا کرده ایم که یک مدل ارتباطی در هر پلت فرم است. با این پیاده سازی، پشتیبانی ارائه شده توسط پلت فرم ها، برای ایجاد مدل های موازی ارایه شده را ارزیابی می کنیم و چهار ویژگی را مورد توجه قرار می دهیم: توزیع، ارتباط، هماهنگ سازی و توازن بار. سپس در ارزیابی کمی، عملکرد پلت فرم ها برای اجرای مدل مرجع بر روی خوشه HPC ارزیابی می شود. فقط دو پلت فرم بنام RepastHPC و Flame توانستند در این محیط اجرا شوند و محدودیت های مربوطه را رعایت کنند. این مقایسه تفاوت قابل ملاحظه ای را از لحاظ مقیاس پذیری بین پلت فرم ها نشان می دهد. Repast HPC نسبت به پلت فرم Flame نتایج عملکرد بهتری را برای مدل مرجع ارائه می دهد.

هنگام پیاده سازی مدل مرجع متوجه شده ایم که پشتیبانی همگام سازی پلت فرم ها، سطح سرویس یکسانی را ارائه نمی دهد: پلت فرم های RepastHPC و DMASON پشتیبانی ارتباطاتی برای عامل های از راه دور را فراهم نمی سازد در حالی که Flame آن را انجام می دهد. توجه داشته باشید که پلت فرم ها یا نواحی همپوشانی را برای بهینه سازی دسترسی عامل به داده های از راه دور ارایه می دهند و یا تبادلات پیام ها را برای اصلاح عامل های از راه دور انجام می دهند. این پشتیبانی یک نقطه کلیدی در عملکرد پلت فرم به نظر می رسد. این تفاوت به مشکلات

همگام سازی مرتبط است. به همین دلیل، در کار آینده ، قصد داریم که کارایی مکانیزم های همگام سازی در پلت فرم های موازی چند عامله را بهتر بررسی کنیم. برای مثال چگونه  هماهنگ سازی ها را در طی اجرا پیاده سازی کنیم ؟  و نحوه بهبود مکانیزم های هماهنگ سازی در سیستم های چند عامله موازی چگونه است؟

تقدیر

محاسبات بر روی امکانات سوپر کامپیوتر از Mésocentre de calcul de Franche-Comté انجام شده است.