

ارائه یک چارچوب نرم‌افزاری برای اجرای کارآمد مدل‌های پیش‌بینی هواشناسی

فرهاد حیدری^۱، محمد عبداللهی ازگمی^{۲*}

۱- دانشجوی کارشناسی ارشد کامپیوتر، دانشکده کامپیوتر، دانشگاه علم و صنعت، استان تهران، تهران، پست الکترونیکی: f_heidari@vu.iust.ac.ir

۲- دانشیار گروه نرم‌افزار، دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت، استان تهران، تهران، پست الکترونیکی: azgomi@iust.ac.ir

تاریخ پذیرش: ۹۹/۴/۲۶

* نویسنده مسوول

تاریخ دریافت: ۹۸/۹/۴

چکیده

مدل پژوهش و پیش‌بینی هواشناسی (WRF) به شدت وابسته به قدرت رایانشی، ورودی-خروجی (I/O) و ارتباطات است. اجرای این مدل در سریع‌ترین رایانه‌ها نیز با چالش‌های بزرگی همراه است. اجرای کارآمد و در مهلت زمانی مقرر، مورد نظر بسیاری از سازمان‌های هواشناسی و مراکز پژوهشی است. امروزه اجرای نرم‌افزارهایی که نیازمند قدرت رایانشی بالا هستند اغلب بر روی زیرساخت‌های رایانش خوشه‌ای و ابری انجام می‌شود. در این زیرساخت‌ها معمولاً از مجازی‌سازی برای مدیریت دسترسی به منابع رایانشی استفاده می‌شود. هدف این مقاله نشان دادن نحوه بهره‌برداری از فناوری حامل‌ها به‌عنوان یک رویکرد مجازی‌سازی سبک‌وزن برای اجرای مدل WRF است، به نحوی که سربار کمی را به منابع رایانشی در زیرساخت‌های ابری تحمیل نماید. استفاده از زمان‌بندی اجرای مدل، سازمان‌ها را قادر می‌سازد که مدل پیش‌بینی را به‌صورت دوره‌ای و گاهی اوقات چند بار در روز اجرا نموده و اعضای مدل را افزایش دهند تا خروجی دقیق‌تری به دست آورند. بنابراین، برای محاسبه زمان اجرای مدل در مقیاس‌های مختلف جغرافیایی بر روی یک سکوی نرم‌افزاری و سخت‌افزاری یک رابطه ریاضی پیشنهاد شده است. رابطه فوق بر اساس تعداد نقاط شبکه، فاصله شبکه‌ای، زمان شبیه‌سازی مدل و تعداد پردازنده‌های مورد استفاده در سخت‌افزار، زمان اجرای مدل را برحسب دقیقه محاسبه می‌نماید.

کلمات کلیدی: رایانش ابری، اجرای مدل WRF، حامل‌ها.

۱. مقدمه

هواشناسی (WRF)^۱ استفاده می‌شود، میزان زیادی از قدرت رایانشی^۲ را درخواست می‌کند [3]. اگر مدل WRF برای مدل‌سازی یک منطقه بزرگ جغرافیایی با وضوح بالا (مثلاً بزرگتر از ۱ کیلومتر) استفاده شود میزان قدرت رایانشی

پیش‌بینی به‌موقع (سروقت) آب‌وهوا با استفاده از آخرین پیشرفت‌های علوم جوی، حتی در سریع‌ترین رایانه‌ها نیز یک چالش است [2]. روش عددی که توسط مدل پژوهش و پیش‌بینی

¹ Weather Research and Forecasting (WRF) model

² Computing power

می‌دهد و زمان به‌دست‌آمده بسیار نزدیک به زمان اجرا تحت سیستم عامل لینوکس بومی (بدون مجازی‌سازی) است. امروزه به‌جای استفاده از روش سنتی که در آن یک مدل واحد از یک وضعیت اولیه تنها، برای پیش‌بینی هواشناسی استفاده می‌کند، از روش گروهی که در آن از یک یا چند مدل با آشفتگی‌هایی در ورودی اولیه و با رفتارهای مختلف از فرآیندهای فیزیکی، برای گزارش موارد غیرقابل پیش‌بینی از طبیعت استفاده می‌شود [5]. از آنجایی‌که روش گروهی نیاز به تعداد معینی از اعضای مدل برای تولید پیش‌بینی دارد، پیش‌بینی می‌باید به‌صورت دوره‌ای، گاهی اوقات چند بار در روز اجرا شود، افزایش سرعت مدل پیش‌بینی هوا و راهبرد ذخیره انرژی برای اجرای مدل‌ها برای آژانس‌های پیش‌بینی هوا بسیار مهم می‌شود [5]. با سرعت بخشیدن به اعضای مدل، سازمان‌های پیش‌بینی می‌توانند اعضای مدل را افزایش دهند تا خروجی مدل بهتری به دست آورده و در یک بازه زمانی مشخص پیش از اعلام پیش‌بینی بعدی، اعلام شود [5].

از آنجایی‌که سازمان‌ها و مؤسسات هواشناسی نیازمند ظرفیت رایانشی بیشتری برای پیش‌بینی وقایع و تجزیه و تحلیل تغییرات آب و هوایی در درازمدت هستند، مقدار قابل توجهی از کار و عملیات بر روی مدل WRF موازی متمرکز شده است [2]. بنابراین زمان‌بندی اجرای مدل WRF برای سازمان‌ها و مؤسسات هواشناسی بسیار مهم است. آن‌ها با استفاده از زمان‌بندی اجرا می‌توانند مدل را چندین نوبت در روز اجرا و خروجی‌های موردنظر را در زمان مناسب تحلیل کنند. زمان‌بندی اجرای مدل در پروژه‌های مرتبط با هواشناسی نیز می‌تواند مورد استفاده قرار گیرد. در این مقاله با انتخاب پنج فاصله شبکه‌ای با تعداد نقاط شبکه‌ای مختلف (یکی از آن‌ها به‌عنوان مرجع و چهار مورد دیگر نسبت به مرجع سنجیده می‌شوند) و اجرای آن‌ها با استفاده از مدل WRF، جامعه آماری از زمان‌های اجرا با مقیاس‌های مختلف تشکیل شده و با استفاده از شبیه‌سازی عددی یک فرمول ریاضی برای محاسبه زمان اجرای مدل در مقیاس‌های مختلف پیشنهاد می‌گردد.

۲. پیشینه تحقیق

در مقاله (Duran-Limon Hector A., 2016) و همکاران [3] نتیجه می‌گیرد ماشین‌های مجازی سبک‌وزن یک سازوکار مناسب

به‌طور چشمگیری افزایش می‌یابد [3]. برای پاسخ به نیازهای رایانشی مدل WRF، نیاز به تجمیع تعداد زیادی منابع رایانشی از طریق زیرساخت‌هایی مانند رایانش خوشه‌ای^۱ یا رایانش ابری^۲ وجود دارد [3]. سگوه‌های^۳ موجود رایانش ابری برای اجرای برخی از برنامه‌های نیازمند رایانش با کارایی بالا (HPC)^۴ مناسب نیستند، زیرا بازده ارائه‌شده توسط آن‌ها کم و ضعیف است [3]. دلیل اصلی این کار سربار مجازی‌سازی^۵ است که توسط اکثر سیستم‌عامل‌های رایانش ابری به‌عنوان وسیله‌ای برای بهینه‌سازی استفاده از منابع استفاده می‌شود [3]. با به‌کارگیری یک نوع مجازی‌سازی سبک‌وزن^۶ بازده به‌طور قابل‌ملاحظه‌ای افزایش می‌یابد و مزایای رایانش ابری از لحاظ سهولت مدیریت، کاهش هزینه‌ها، بهینه‌سازی استفاده از منابع و غیره حفظ می‌گردد [3].

در این مقاله نتایج پژوهشی ارائه می‌شود که در آن دو مدل مجازی‌سازی سبک‌وزن و سنگین‌وزن به‌عنوان کنترل‌کننده منابع در ابرها و خوشه‌های رایانشی بررسی و مقایسه شده و از آن‌ها برای ارائه یک زیرساخت نرم‌افزاری کارآمد برای اجرای مدل WRF استفاده شده است.

در حوزه فناوری‌های مجازی‌سازی، حامل (کانتینر)^۷ به واحد استاندارد نرم‌افزاری گفته می‌شود که به‌صورت مجزا قابل توسعه، انتقال و استقرار بوده و همه وابستگی‌های لازم برای اجرا را با خود به همراه دارد. برای نشان دادن بهره‌وری مناسب یک چارچوب مبتنی بر حامل‌ها، زمان اجرای مدل WRF تحت سیستم عامل لینوکس بومی^۸ (بدون مجازی‌سازی) به‌عنوان زمان مرجع ارزیابی انتخاب شده و مقایسه‌ای بین زمان اجرای مدل در حامل‌های مورد نظر در این مقاله (یعنی Docker و Singularity) و مجازی‌سازی تحت ابرناظر^۹ KVM نسبت به زمان مرجع انجام می‌شود. در این پژوهش نشان می‌دهیم که ویژگی مجازی‌سازی سبک‌وزن مبتنی بر حامل‌ها در مقایسه با مجازی‌سازی سنگین‌وزن مبتنی بر ابرناظر، زمان کلی اجرای مدل WRF را به‌عنوان یک برنامه کاربردی علمی نیازمند رایانش با کارایی بالا [4] کاهش

¹ Cluster computing

² Cloud computing

³ Platforms

⁴ High Performance Computing (HPC)

⁵ Virtualization

⁶ Lightweight Virtualization

⁷ Container

⁸ Native Linux

⁹ Hypervisor

برای بررسی عملکرد مجازی‌سازی از ابزار ABAQUS استفاده می‌کند.

در این تحقیق برای بررسی عملکرد مجازی‌سازی سنگین‌وزن و سبک‌وزن در اجرای مدل هواشناسی WRF از روش تحلیل آماری استفاده می‌شود. همچنین در این تحقیق عملکرد شبکه بررسی نمی‌شود.

به‌منظور معرفی زیرساخت نرم‌افزاری برای اجرای کارای مدل پژوهش و پیش‌بینی آب‌وهوای WRF، مدل در محیط‌های لینوکس بدون مجازی‌سازی، حامل‌های Docker و Singularity و مجازی‌سازی KVM نصب و پیکربندی شد.

زمان اجرا در هر محیط ثابت و موردبررسی و تحلیل آماری قرار می‌گیرد. مقایسه اصلی بین زمان اجرا در حامل‌ها و مجازی‌سازی KVM نسبت به لینوکس بدون مجازی‌سازی است تا مشخص شود چه محیطی برای اجرای مدل WRF که یک برنامه نیازمند رایانش با کارایی بالا است، بهینه‌تر است. دلیل اینکه KVM برای ابرناظر انتخاب می‌شود این است که KVM روش کامل مجازی‌سازی است و اغلب با Docker سازگار است [9]. با تحلیل نتایج آماری، میزان سربار هر محیط در اجرای مدل محاسبه می‌شود. همچنین مقیاس‌پذیری هر محیط در اجرای مدل بررسی می‌شود. در نهایت نتایج حاصل نشان می‌دهند کدام محیط دارای کمترین سربار و بهترین مدیریت منابع را در اجرای مدل داشته است.

کلیه آزمایش‌ها با استفاده از یک دستگاه لپ‌تاپ مدل Lenovo W530 با مشخصات زیر انجام شده است:

- CPU : Intel Ci7 3720QM 2.6 GHz
- RAM : 8GB
- HDD : 700 GB

برای محاسبه زمان‌بندی اجرای مدل WRF تاکنون روشی ابداع نشده است. در این مقاله برای اینکه بتوانیم زمان‌بندی اجرای مدل را به دست آوریم، پارامترهایی که در سرعت اجرای مدل تأثیر دارند را بررسی کردیم.

زمان یا سرعت اجرای مدل WRF تابع زمان شبیه‌سازی مدل، تعداد هسته‌های پردازشی، تعداد نقاط و فاصله شبکه‌ای است که انتخاب می‌شود. وقتی که مدل را برای محاسبه بر روی یک منطقه

برای اجرای برنامه‌های کاربردی HPC با حفظ مزایای رایانش ابری مانند بهینه‌سازی استفاده از منابع هستند. سربار کلی توسط ماشین‌های مجازی سبک‌وزن حدود پنج درصد است. در مقابل، ماشین‌های مجازی KVM دارای سربار بسیار بالاتری هستند. این پژوهش به‌صورت کلی به مقایسه مجازی‌سازی سبک‌وزن و سنگین‌وزن می‌پردازد و از حامل‌های مورد نظر تحقیق ما یعنی Docker و Singularity استفاده نمی‌کند.

در مقاله (Kovacs A., 2017) [6]، حامل‌های LXC، Docker، و Singularity با KVM (ماشین مجازی مبتنی بر هسته) مقایسه شده و در آن عملکرد مجازی‌سازی با لینوکس بدون مجازی‌سازی از دیدگاه سربار عملکرد CPU و شبکه بررسی شده است. این تحقیق نشان می‌دهد که عملکرد CPU و شبکه حامل‌ها تقریباً در سطح عملکرد لینوکس بدون مجازی‌سازی قرار دارند و این برای اجرای یک برنامه حساس به CPU و شبکه و استفاده برای برنامه‌های نیازمند رایانش با کارایی بالا بسیار خوب است. این تحقیق به بررسی اجرای مدل WRF در حامل‌ها و KVM نمی‌پردازد و فقط بازده آن‌ها را بررسی می‌کند.

در مقاله (Krishnan S. P. T., 2014) [1] نشان داده‌اند که اکثر زمان اجرای مدل WRF برای هم‌رسانی رابط انتقال پیام^۱ صرف می‌شود که به‌عنوان یک عملیات ارتباطی جمعی تعریف شده که شامل یک گروه از پردازنده‌ها^۲ می‌شود. در این حالت سرعت و عملکرد WRF تا حد زیادی وابسته به هم‌بندی شبکه^۳ است که اتصال بین گره‌های رایانشی را برقرار می‌کند. این تحقیق اجرای مدل WRF را از دید سرعت شبکه ارتباطی بررسی می‌کند و اجرای مدل روی حامل‌ها را انجام نمی‌دهد.

۳. داده‌ها و روش‌ها

روش‌های نرم‌افزاری مختلفی برای بررسی میزان بازده مجازی‌سازی سبک‌وزن و سنگین‌وزن مورد استفاده قرار می‌گیرد. در [6] و [8] از نرم‌افزار LINPACK و SYSBENCH برای بررسی عملکرد CPU و IPerf و NETperf برای بررسی عملکرد شبکه برای مقایسه مجازی‌سازی استفاده می‌کند. همچنین در [10]

¹ Message Passing Interface (MPI) Broadcast

² Processes

³ Network topology

فاصله شبکه‌ای برای شبکه‌های غیر مرجع 20km, 25km, 15km و 12km که ضریب درستی از فاصله شبکه‌ای مرجع هستند در نظر گرفته شد. با کاهش فاصله شبکه‌ای، تعداد نقاط نیز افزایش می‌یابد اما هر دو به صورت ضریبی از شبکه و نقاط مرجع هستند.

در طول آزمایش اجرای مدل WRF در محیط لینوکس بدون مجازی‌سازی انجام می‌شود. در این آزمایش مدل WRF به صورت پایه انتخاب شده است یعنی ماژول‌های جانبی آن مانند باد، باران، برف، تگرگ و غیره انتخاب نشده‌اند و تنها ماژول WRFDA² که موجب می‌شود داده‌های خروجی مدل دقیق‌تر باشد مورداستفاده قرار گرفت. این ماژول در زمان اجرای مدل تأثیری بسیار ناچیزی دارد و همچنین سایز فایل خروجی را در حد چند ده مگابایت افزایش می‌دهد.

کلیه آزمایش‌ها بر روی یک دستگاه کامپیوتر خدمتگزار³ مدل HP 560 G9 انجام شد. مشخصات آن به شرح زیر است:

- Model = ProLiant DL560 Gen9
- CPU : 2× Intel(R) Xeon(R) CPU E5-4669 v4 @ 2.20GHz
- CPU Core : cores=44, threads=44
- Memory : 256GiB

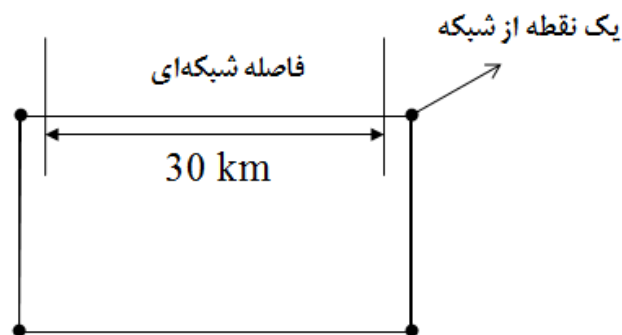
کامپیوتر فوق دارای ۸۸ هسته است. با توجه به اینکه سخت‌افزارهای موجود اعم از خدمتگزار و شخصی حداقل دارای ۴ هسته پردازشی هستند، آزمایش‌ها از تعداد ۴ هسته شروع و تا ۸ هسته (حداکثر تعداد هسته کامپیوترهای شخصی موجود) به صورت افزایش ۱ هسته و بعد از آن از ۱۰ هسته تا ۸۴ هسته به صورت افزایش ۵ هسته‌ای در هر اجرا، انجام شد.

برای محاسبه زمان‌بندی اجرای مدل WRF در این مقاله از روش تحلیل آماری استفاده شده است. اجرای مدل WRF بر روی کامپیوتر فوق و تعداد معینی هسته CPU برای هر فاصله و نقاط شبکه‌ای انجام شد. سپس با استفاده از جامعه آماری ایجادشده و شبیه‌سازی عددی در نرم‌افزار MATLAB یک رابطه زمان‌بندی برای اجرای مدل به استخراج گردید. رابطه فوق بر اساس پارامترهای تعداد نقاط شبکه، فاصله شبکه‌ای، تعداد ساعت

جغرافیایی تنظیم می‌کنیم، در قسمت آماده‌سازی¹ مدل تعداد نقاط و فاصله شبکه‌ای موردنظر را مشخص می‌کنیم. این نقاط به صورت یک صفحه در راستای محور x و y در نظر گرفته می‌شوند که حاصل آن یعنی (x×y) تعداد نقاط شبکه موردنظر را تعیین می‌کند. در هر لحظه از اجرای مدل، نقاط یک صفحه در راستای عمودی (z یا ارتفاع) محاسبه می‌شود.

برای اینکه بتوانیم مرجعی برای راینش خود ایجاد کنیم، یک فاصله شبکه‌ای مرجع را طوری انتخاب کردیم که فاصله‌های شبکه‌ای دیگر مورد آزمایش، ضریب درستی از فاصله شبکه‌ای مرجع باشند. این کار از آن جهت صورت گرفت که تعداد نقاط شبکه‌ای در مقایسه با نقاط شبکه‌ای مرجع دقیق محاسبه شود و در جریان محاسبه، نقاط از دست نرود. این ضریب (فاصله شبکه‌ای مرجع تقسیم بر فاصله شبکه‌ای دیگر) در تعداد نقاط شبکه‌های دیگر ضرب می‌شود تا ضمن کم شدن فاصله شبکه‌ای، تعداد نقاط نیز افزایش یابد.

با کاهش طول شبکه، تعداد نقاط شبکه برای همان منطقه شبیه‌سازی افزایش می‌یابد به طوری که این افزایش تابعی از همین فاصله شبکه است. در این آزمایش فاصله شبکه‌ای ۳۰ کیلومتر به عنوان مرجع آزمایش‌ها در نظر گرفته شد و عدد یک به آن اختصاص داده شد. افزایش نقاط با کاهش فاصله شبکه‌ای رابطه دقیقی از نسبت فاصله شبکه‌ای مرجع (۳۰ کیلومتر) به فاصله شبکه‌ای موردنظر است. به عبارت دیگر تعداد نقاط مورد آزمایش و به تبع آن زمان آزمایش تابعی از نسبت فاصله شبکه‌ای موردنظر به فاصله شبکه‌ای مرجع خواهد بود. به همین دلیل رابطه نهایی به دست آمده از این آزمایش‌ها تابعی از نسبت تعداد نقاط و فاصله شبکه‌ای خواهد بود.



شکل ۱: فاصله شبکه‌ای و نقاط شبکه‌ای در حالت مرجع

² WRF Data Assimilation

³ Server

¹ Preprocessing

• در Singularity سادگی جابجایی کانتینرهای حاوی مسائل علمی و پردازشی مورد توجه قرار می‌گیرد. اگرچه Singularity از namespaces استفاده می‌کند، اما در اصل برای قابلیت حمل برنامه به جای مجازی‌سازی میزبان استفاده می‌شود؛ به عبارت دیگر Singularity فقط آنچه را که برای دستیابی به کانتینرهای اجرای برنامه در زمان و محیط‌های قابل حمل لازم است مجازی می‌کند [13].

• کانتینر Singularity بر فراهم کردن محیط‌های قابل حمل که می‌تواند از مهاجرت علوم محاسباتی به ابر استفاده کند، تمرکز دارد [13].

• حامل دارای سربار کمتری است. در بهترین حالت می‌توانید پنجاه عدد VM را در بالای میزبان مجازی فیزیکی اجرا کنید و به یک میزبان قدرتمند نیاز دارید تا حتی آن تعداد را راه‌اندازی کنید. اما چون حامل دارای سربار کمتری است، می‌توانید صد حامل را روی میزبان تک اجرا کنید [7].

• کار با حامل‌ها آسان است. یکی از ویژگی‌های جالب حامل docker این است که شما می‌توانید یک تصویر حامل را در چند ثانیه بکشید و اجرا کنید. همچنین می‌توانید تصویر VM را از اینترنت دانلود کنید و آن را راه‌اندازی کنید اما این روند به صورت جریان‌ی (stream) نیست [7].

• بیشتر حامل‌ها استاندارد شده‌اند. چندین سکوی VM وجود دارد که هرکدام با شیوه خاصی کارها را انجام می‌دهند [7]. حامل‌ها اغلب منبع باز هستند. بعضی از سکویهای VM مانند KVM منبع باز هستند، اما اکثر آن‌ها محصول تجاری هستند، که تنها تا حدی باز هستند. در مقابل سکوی مانند docker به طور کامل منبع باز است که باعث می‌شود حامل‌ها انتخاب بهتری باشند اگر شما در مورد lockin فروشنده نگران هستید [7].

• حامل‌ها داخل یک دستگاه فیزیکی عموماً آدرس‌های IP جداگانه ندارند، که امروزه سود عظیمی است زیرا آدرس‌های IPv4 بسیار محدود هستند [11].

• تصاویر حامل Docker نیاز به فضای دیسک کمتری نسبت به تصاویر VM دارند، زیرا بخشی که چند حامل به صورت عمومی دارند، می‌تواند در اغلب اوقات بین آن‌ها به اشتراک گذاشته شود. این باعث تسریع استقرار حامل‌های Docker می‌شود، زیرا تصاویر معمولاً بر روی شبکه و گره

شبیه‌سازی مدل و تعداد هسته پردازشی مورد استفاده، زمان اجرای مدل WRF را محاسبه می‌نماید.

۴. ارزیابی و مقایسه زیرساخت نرم‌افزاری

۴-۱ تعریف مجازی‌سازی و حامل

روش پایه لازم برای رایانش ابری، فناوری سرور مجازی شده است که ابرناظر نامیده می‌شود [9]. ابرناظر به معنی فناوری است که چندین ماشین مجازی که دارای پردازنده، حافظه و شبکه مستقل هستند، بر روی یک سرور فیزیکی بارگذاری می‌شوند و سیستم عامل سرور منابع سرور را به اشتراک می‌گذارد [9]. حامل یک نرم‌افزار مجازی‌سازی سطح برنامه کاربردی است که می‌تواند به مدیریت سیستم کامپیوتری که جدا از سیستم عامل است با استفاده از Lib container که سیستم حامل لینوکس است، کمک کند [9]. در واقع، مفهوم پایه حامل بسیار متفاوت از مجازی‌سازی موجود نیست. تنها روش متفاوتی از همان رویکرد دارد. مجازی‌سازی موجود سخت‌افزار و سیستم عامل را یکبار بارگیری می‌کند [9].

برخی مزایای حامل‌ها عبارت‌اند از:

- کانتینر Docker از آن زمانی که معرفی گردید به عنوان انتخاب اصلی در زمینه مجازی‌سازی ظاهر و به طور گسترده‌ای در صنعت و پژوهش بکار گرفته شده است، زیرا باعث افزایش قدرت زیرساخت‌ها برای ارائه سرویس‌های متنوع می‌گردد [13].
- Docker ایجاد، توزیع و اجرای کانتینرهای لینوکس را ساده کرده است. علاوه بر تهیه ابزار برای ساخت کانتینرها و اجرای آن‌ها، از مخازن Docker^۱ نیز پشتیبانی می‌کند و امکان بارگذاری و بارگیری تصاویر کانتینر را فراهم می‌کند. با این ویژگی‌های بسیار جذاب اکنون Docker به طور گسترده‌ای در زمینه ارائه سرویس و برنامه‌های کاربردی استفاده می‌شود [14].
- کانتینر Singularity باهدف افزایش تحرک محاسبه در کاربردهای علمی ایجاد شده است [15]

^۱ Dockerhub

نامعلوم است از توزیع t استفاده می‌گردد که در آن کمیت محوری T بصورت زیر تعریف می‌شود:

$$T = \frac{\bar{X} - \mu}{\frac{S}{\sqrt{n}}}$$

\bar{X} : برآورد نقطه‌ای برای میانگین جامعه آماری

μ : میانگین جامعه آماری

S : انحراف معیار

n : تعداد نمونه

و واریانس نمونه‌ای از فرمول زیر محاسبه می‌گردد:

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$$

با توجه به اینکه T دارای توزیع t با $n-1$ درجه آزادی و یا $T \sim t(n-1)$ است؛ بنابراین T می‌تواند یک کمیت محوری باشد و با استفاده از فرمول زیر فاصله اطمینان میانگین جامعه قابل محاسبه خواهد بود:

$$\left(\bar{X} - t_{(n-1, 1-\frac{\alpha}{2})} \frac{S}{\sqrt{n}}, \bar{X} + t_{(n-1, 1-\frac{\alpha}{2})} \frac{S}{\sqrt{n}} \right)$$

با استفاده از فرمول مذکور بازه اطمینانی برای هر محیط آزمایش محاسبه شد که نتایج در جدول (۲) آمده است:

جدول ۱: زمان اجرای مدل WRF در محیط‌های عملیاتی

ردیف	3Hour - 1 Domain			
	Singularity (Second)	Docker (Second)	Native (Second)	KVM (Second)
1	156.713	157.35	155.38	165.666
2	155.569	156.86	155.01	165.662
3	155.054	157.48	155.43	164.991
4	154.762	158.04	156.26	166.700
5	155.972	158.21	155.92	164.968
6	155.344	157.91	155.41	166.035
7	155.086	157.25	156.33	164.969
8	155.398	156.98	155.10	164.680
9	155.853	157.88	155.30	164.998
10	155.377	158.70	155.42	166.485
11	155.974	156.97	155.22	165.360

جدول ۲: بازه اطمینان زمان اجرای مدل WRF در محیط‌های آزمایش

محیط آزمایش	Average (Seconds)	Lower limit (Seconds)	Upper limit (Seconds)
Singularity	155.5547	155.5547	156.0097
Docker	157.6027	157.1113	158.0947
محیط لینوکس Native	155.5255	155.1543	155.8977
محیط KVM	165.5013	164.941	166.061

رایانشی قبل از اینکه نمونه بتواند شروع شود، باید کپی شود [11].

- زمان بارگذاری سریع. حامل‌ها می‌توانند بسیار سریع‌تر از VMها شروع شوند زیرا برخلاف VMها، حامل‌ها لازم نیست یک نسخه دیگر از سیستم‌عامل را اجرا کنند [11].
- بدون نیاز به مهاجرت به حامل‌ها در عرض چند ثانیه می‌توان یک حامل جدید را شروع و ترافیک را به حامل جدید ایجادشده ارسال کرده و حامل قدیمی را حذف کرد [11].
- حامل برنامه سبک‌وزنی است که موجب به حداقل رساندن منابع و پهنای باند موردنیاز برای استقرار و گسترش می‌شود [11].
- با دسترسی مستقیم به توابع سخت‌افزاری حامل‌ها می‌توانند بر روی هر محیط یا سخت‌افزاری که قادر به اجرای یک سیستم‌عامل لینوکس باشد اجرا شوند [11].

۲-۴ پیاده‌سازی و آزمایش

این آزمایش شامل اجرای مدل WRF در هر یک از محیط‌های لینوکس بدون مجازی‌سازی، Docker، Singularity و KVM است. در هر یک از محیط‌ها دو طرح اجراشده است که به صورت زیر است:

۲-۴-۱ طرح اول

در هر یک از محیط‌های فوق مدل پژوهش و پیش‌بینی هواشناسی WRF به تعداد ۱۱ بار در حالت شبیه‌سازی ۳ ساعته داده‌های ورودی و روی یک دامنه جغرافیایی اجرا و زمان اجرای آن‌ها ثبت شد. هدف از تشکیل این جامعه آماری بررسی زمان متوسط اجرای مدل در هر یک محیط‌ها است. جدول شماره (۱) نشان‌دهنده زمان‌های اجرای مدل WRF در هر محیط است. برای اطمینان از صحت داده‌های آماری جدول فوق، بازه اطمینان ۹۵ درصدی (یعنی با احتساب ۵ درصد خطا) طبق روش زیر برای آنها محاسبه گردید.

برای محاسبه بازه اطمینان داده‌های مربوط به اجرای مدل WRF در هر یک از محیط‌های آزمایش با توجه به اینکه تعداد داده‌های آماری برای هر محیط ۱۱ عدد و واریانس نمونه‌ها نیز

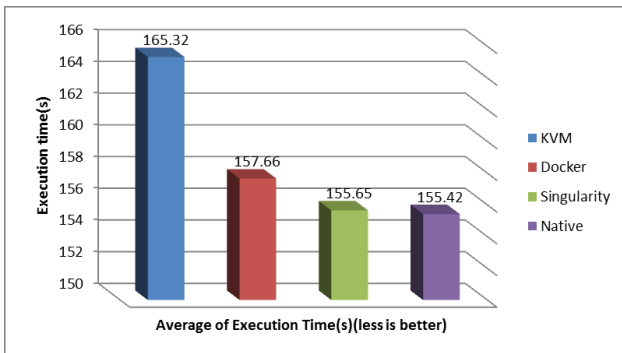
اعداد بدست آمده از آزمایش فوق که در جدول (۴) قرار دارند با روش Min-Max نرمال‌سازی شدند و نتایج در جدول (۵) آورده شده است.

جدول ۴: نتایج سربار اجرای مدل در هر محیط عملیاتی نسبت به محیط لینوکس بدون مجازی‌سازی

3Hour – 1 Domain			
Overhead	Singularity	Docker	KVM
	0.02	1.34	6.42

جدول ۵: مقایسه سربار زمانی نرمال شده اجرای مدل در هر محیط عملیاتی نسبت به محیط لینوکس بدون مجازی‌سازی

3Hour – 1 Domain			
Overhead	Singularity	Docker	KVM
	0	0.206	1



شکل ۲: متوسط زمان اجرای مدل در محیط‌های عملیاتی

نتایج جدول (۵) نشان می‌دهد، اجرای مدل WRF در محیط عملیاتی Singularity کمترین میزان سربار زمانی را نسبت به دیگر محیط‌ها دارد.

در اینجا این توضیح لازم است که در هر سیستم عاملی با اجرای یک برنامه کاربردی مقداری از منابع سیستمی در اختیار آن برنامه قرار می‌گیرد. هرچقدر تعداد برنامه‌های در حال اجرا افزایش یابد، با توجه به محدودیت منابع سیستمی، اجرای آنها با کندی همراه خواهد بود و زمان بیشتری نیاز است تا هر برنامه به نتیجه برسد؛ بنابراین یک سربار زمانی ناشی از سربار پردازش دیگر برنامه‌ها برای هر برنامه ایجاد می‌شود.

در این آزمایش منظور از سربار در حقیقت سربار زمانی در اجرای مدل WRF در هر یک از محیط‌های آزمایش است. برای اجرای مدل، لازم است مسیر داده‌های ورودی برای مدل مشخص گردد. در محیط‌های کانتینر Docker و Singularity این موضوع بدین‌صورت انجام شد که داده‌ها روی سیستم میزبان (لینوکس

جدول (۲) بازه اطمینان برای هر محیط را نشان می‌دهند. با توجه به بازه اطمینانی که برای هر محیط محاسبه شده است تعدادی از داده‌های جدول (۱) خارج از بازه قرار می‌گیرند. این داده‌ها به‌صورت زیرخط‌دار و قرمز رنگ در جدول فوق مشخص شده‌اند. بنابراین مدل WRF در محیط‌های آزمایش مجدداً اجرا شد و زمان‌های اجرایی که در بازه اطمینان قرار داشتند با داده‌های خارج از بازه اطمینان جدول (۱) جابه‌جا شدند و جدول آماری (۳) تشکیل گردید.

جدول ۳: زمان اجرای مدل WRF در محدوده بازه اطمینان در محیط‌های آزمایش

ردیف	3Hour – 1 Domain			
	Singularity (Seconds)	Docker (Seconds)	Native (Seconds)	KVM (Seconds)
1	155.201	157.35	155.38	165.666
2	155.569	157.431	155.627	165.662
3	155.983	157.48	155.43	164.991
4	155.871	158.04	155.292	165.241
5	155.972	157.711	155.526	164.968
6	155.344	157.91	155.41	166.035
7	155.587	157.25	155.491	164.969
8	155.398	157.924	155.522	165.314
9	155.853	157.88	155.3	164.998
10	155.377	157.489	155.42	165.294
11	155.974	157.845	155.22	165.36
Average	155.65	157.66	155.42	165.32

برای بررسی و مقایسه زمان اجرا در هر محیط، متوسط زمانی هر ستون جدول (۳) محاسبه شد. همانطور که در شکل (۲) مشاهده می‌شود متوسط زمان اجرا در محیط Singularity و سپس Docker خیلی نزدیک به محیط لینوکس Native است. در این آزمایش بیشترین متوسط زمان اجرا مربوط به محیط KVM است.

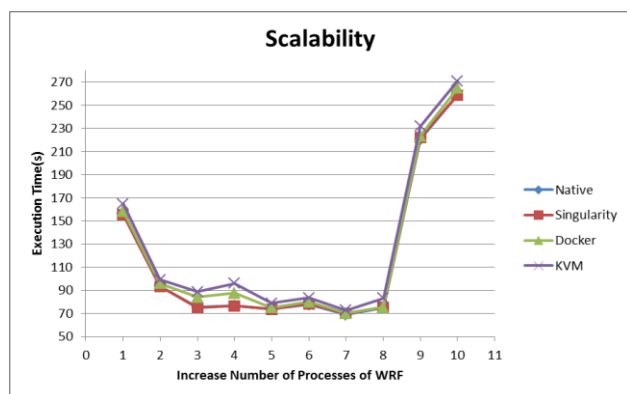
۱-۱-۲-۴ محاسبه سربار محیط‌های عملیاتی در اجرای مدل WRF

یکی اهداف در این آزمایش بررسی میزان سربار اضافی اجرای مدل WRF در هر یک از محیط‌های عملیاتی نسبت به اجرای این مدل در محیط لینوکس بدون مجازی‌سازی است، با استفاده از فرمول زیر این سربار محاسبه می‌گردد.

$$[3] \text{Overhead} = ((100 \times E_i) / E_j) - 100$$

- E_i متوسط زمان اجرای مدل WRF در هر یک از محیط‌های مجازی شده است
- E_j متوسط زمان اجرای مدل WRF در لینوکس بدون مجازی‌سازی است

افزایش تعداد هسته‌های CPU برای پردازش موازی مدل WRF، زمان اجرای مدل کاهش می‌یابد.



شکل ۳: نمودار مقایسه مقیاس‌پذیری اجرای مدل در محیط‌های عملیاتی با افزایش تعداد پردازش‌های موازی و هم‌زمان

در شکل (۲) می‌توان میزان مقیاس‌پذیری محیط‌ها را باهم مقایسه کرد. این شکل نشان می‌دهد با افزایش تعداد پردازش‌های موازی WRF.exe زمان اجرای مدل کمتر می‌شود. این کاهش از ۱ پردازش WRF.exe به ۲ پردازش به دلیل اجرای موازی مدل با شیب بیشتری همراه است. همچنین زمان اجرای مدل در حامل Singularity بسیار نزدیک به محیط لینوکس بدون مجازی‌سازی است و زمان اجرا در حامل Docker بافاصله بیشتری نسبت به محیط بدون مجازی‌سازی قرار دارد. بیشترین زمان متعلق به محیط مجازی‌سازی KVM است. از آنجایی که سخت‌افزار مورد استفاده دارای ۸ هسته است وقتی تعداد پردازش‌های WRF از تعداد هسته‌های موجود بیشتر می‌شود (در نقاط ۹ و ۱۰ روی شکل ۲)، زمان اجرا به شدت افزایش می‌یابد. این افزایش ناشی از رقابت پردازش‌های WRF.exe اضافی برای دسترسی به منابع (در اینجا CPU) سیستم است. و برای دسترسی به منابع، هر پردازش باید صبر کند تا دسترسی آن به منابع ایجاد شود، این تأخیرها موجب افزایش زمان کلی اجرای مدل می‌شود. زمان اجرا در نقاط ۹ و ۱۰ شکل (۲) در محیط KVM بیشتر است و محیط لینوکس بدون مجازی‌سازی و Singularity به هم نزدیک هستند.

۴-۲-۱ پیشنهاد زمان‌بندی اجرای مدل WRF

طرح‌هایی که پس از تنظیمات لازم برای مدل WRF بر اساس فاصله شبکه‌ای روی کامپیوتر خدمت‌گزار اجرا شد دارای مشخصات زیر هستند:

قرار گرفت و مسیر داده‌ها برای کانتینرها معرفی و سوار گردید. هنگام اجرای مدل در محیط‌های کانتینر، داده‌ها از مسیر معرفی شده موجود بر روی میزبان خوانده می‌شود و در مدل مورد پردازش قرار می‌گیرد؛ بنابراین در کانتینرها قسمتی از سربرار زمانی ایجاد شده در اجرای مدل، مربوط به انتقال داده‌ها از سیستم فایل میزبان به کانتینر است. در ابرناظر KVM این سربرار اضافی وجود ندارد زیرا داده‌ها ورودی به سیستم عامل مهمان انتقال داده شد. با این حال زمان اجرای مدل WRF در ماشین مجازی موجود بر روی KVM بیش از دیگر محیط‌ها است. قسمتی از سربرار زمانی نیز مربوط به خود برنامه کانتینرهای Singularity، Docker و ابرناظر KVM است که برای مجازی‌سازی در حافظه بارگیری و اجرا می‌شوند. اجرای مدل در سیستم میزبان (لینوکس Native) هیچ‌کدام از سربرار مطروحه بالا را ایجاد نمی‌کند، بنابراین کمترین زمان اجرای مدل WRF را دارا است.

۴-۲-۲ طرح دوم

هدف از این آزمایش بررسی میزان مقیاس‌پذیری محیط‌های عملیاتی Singularity، Docker، KVM و لینوکس بدون مجازی‌سازی است. با توجه به توانایی سخت‌افزار مورد استفاده که یک لپ‌تاپ با CPU دارای ۸ هسته است، تعداد پردازش‌های هم‌زمان و موازی مدل WRF یعنی WRF.exe با استفاده از بسته نرم‌افزاری موازی‌سازی MPI

جدول ۶: نتایج مقیاس‌پذیری محیط‌های عملیاتی در اجرای مدل WRF

Number of WRF.exe Process	3Hour - 1 Domain			
	Native (Second)	Singularity (Second)	Docker (Second)	KVM (Second)
1 × wrf.exe	155.87	155.68	158.45	164.82
2 × wrf.exe	93.84	93.55	95.98	99.38
3 × wrf.exe	75.55	75.35	84.27	88.96
4 × wrf.exe	76.75	76.68	87.71	95.99
5 × wrf.exe	73.88	73.80	75.17	79.14
6 × wrf.exe	77.73	78.48	80.1	83.53
7 × wrf.exe	69.17	70.1	70.39	73.02
8 × wrf.exe	75.38	75.57	75.61	83.46
9 × wrf.exe	220.53	221.88	223.35	231.95
10 × wrf.exe	259.08	258.72	264.96	270.89

از ۱ هسته تا ۸ هسته مورد آزمایش قرار گرفت. بدین معنی که در هر مرحله هر پردازش WRF.exe توسط یک هسته اجرا می‌شود و به ترتیب تعداد هسته‌های بیشتری درگیر اجرای wrf.exe می‌شوند. داده‌های جدول شماره (۶) نشان می‌دهد با

وابستگی مکانی دارند و در نتیجه با افزایش تعداد نقاط شبکه با ضریب ۲۳ زمان محاسبه افزایش می‌یابد [12].

همچنین وقتی فاصله شبکه‌ای کم یا نصف می‌شود این کاهش فقط در دو بعد است و در محور قائم این کاهش وجود ندارد. در حقیقت رایانش بر روی صفحاتی که در مقطع قائم جو بالا می‌رود، انجام می‌شود. جدول (۸) بر حسب تعداد نقاط هر فاصله شبکه‌ای و تعداد هسته به صورت زیر تشکیل داده شد.

جدول ۷: نتایج اجرای مدل WRF در فاصله شبکه‌ای مختلف

Number of CPU Core (WRF.exe Process)	Time (minutes)				
	30km	25km	20km	15km	12km
4	2.984	4.831	8.913	21.957	40.015
5	2.636	4.439	8.001	18.811	38.642
6	2.15	3.492	6.18	15.781	30.26
7	1.957	3.299	5.738	14.446	28.524
8	1.734	2.717	5.026	12.262	22.606
10	1.628	2.317	4.172	10.55	20.182
15	1.128	1.839	3.195	7.682	14.68
20	0.865	1.729	2.541	5.997	11.519
25	0.861	1.381	2.32	5.402	10.509
30	0.732	1.229	2.062	5.23	10.467
35	0.693	1.01	1.987	5.152	8.607
40	0.645	1.003	1.91	4.611	8.548
45	0.633	0.978	1.82	4.225	8.001
50	0.607	0.971	1.799	4.11	7.931
55	0.607	0.97	1.754	4.061	7.674
60	0.6	0.921	1.726	4.049	7.656
65	0.585	0.906	1.655	3.841	7.202
70	0.581	0.892	1.636	3.762	7.193
75	0.549	0.877	1.584	3.644	6.803
80	0.542	0.854	1.534	3.499	6.681
84	0.54	0.85	1.493	3.481	6.483

جدول ۸: تعداد نقاط شبکه در هر فاصله شبکه‌ای

Number of CPU Core (WRF.exe Process)	تعداد نقاط هر فاصله شبکه‌ای				
	30km	25km	20km	15km	12km
4	31861	45793	71416	126721	197776
5	31861	45793	71416	126721	197776
6	31861	45793	71416	126721	197776
7	31861	45793	71416	126721	197776
8	31861	45793	71416	126721	197776
10	31861	45793	71416	126721	197776
15	31861	45793	71416	126721	197776
20	31861	45793	71416	126721	197776
25	31861	45793	71416	126721	197776
30	31861	45793	71416	126721	197776
35	31861	45793	71416	126721	197776
40	31861	45793	71416	126721	197776
45	31861	45793	71416	126721	197776
50	31861	45793	71416	126721	197776
55	31861	45793	71416	126721	197776
60	31861	45793	71416	126721	197776
65	31861	45793	71416	126721	197776
70	31861	45793	71416	126721	197776
75	31861	45793	71416	126721	197776
80	31861	45793	71416	126721	197776
84	31861	45793	71416	126721	197776

۱) شبکه 30 km با تعداد نقاط $31861 = 211 \times 151$ عدد به عنوان شبکه مرجع

۲) شبکه 25 km با تعداد نقاط $45793 = 253 \times 181$ عدد برای محاسبه تعداد نقاط شبکه‌های غیر مرجع بدین صورت عمل شد:

[(تعداد نقاط شبکه مرجع - ۱) × (فاصله شبکه‌ای جدید /

فاصله شبکه‌ای مرجع)] + ۱

$$[(211 - 1) \times (30/25)] + 1 = 253$$

$$[(151-1) \times (30/25)] + 1 = 181$$

عمل جمع و منها برای این انجام شد که در جریان افزایش تعداد نقاط شبکه جدید، هیچ نقطه‌ای از شبکه جدید بیرون حوزه شبیه‌سازی قرار نگیرد و یا اینکه تعداد نقاط به گونه‌ای نباشد که نتواند کل حوزه شبیه‌سازی را پوشش دهد (نقاط شبکه جدید ضریبی از شبکه مرجع هستند).

۳) شبکه 20 km با تعداد نقاط $71416 = 316 \times 226$ عدد

۴) شبکه 15 km با تعداد نقاط $126721 = 421 \times 301$ عدد

۵) شبکه 12 km با تعداد نقاط $197776 = 526 \times 376$ عدد

۵. نتایج ارزیابی

جدول (۷) نشان‌دهنده زمان اجرای مدل WRF در فاصله

شبکه‌ای مختلف است.

شکل (۳) نشان می‌دهد زمان پردازش صفحات شبکه‌ای در فاصله شبکه‌ای 30km , 25km , 20km , 15km و 12km کاملاً شبیه هم بوده و با افزایش تعداد هسته‌های CPU به صورت نمایی کاهش می‌یابند. رابطه آن‌ها در حالت کلی می‌تواند به صورت زیر باشد:

$$Y = aX^b + c$$

که در آن a, b, c ضرایب ثابت، X تعداد هسته پردازشی و Y زمان اجرای مدل است. می‌توان نتیجه گرفت روابط زمان و تعداد هسته نسبت به هم از نوع نمایی خواهد بود.

در اینجا باید یادآوری شود که بخشی از معادلات مدل وابستگی زمان و مکان دارند و در نتیجه با کاهش طول شبکه (مثلاً افزایش تعداد نقاط از 30km به 15km) زمان شبیه‌سازی ضریبی از ۲۴ خواهد بود [12]. اما بخش دیگر معادلات مدل صرفاً

البته این رابطه برای یک نقطه شبکه و زمان شبیه‌سازی ۶ ساعته مدل به‌دست آمده است و مدت زمان اجرا برای یک نقطه شبکه در طی بازه ۶ ساعته در مدل WRF را بر حسب دقیقه می‌دهد. اگر تعداد نقاط شبکه را no و زمان شبیه‌سازی مدل WRF را T ساعت در نظر بگیریم رابطه تصحیح شده به شکل زیر خواهد بود:

$$f(x, y, T, no) = \left(\frac{T}{6}\right)(no)(2.914 \times 10^{-4} xy^{-0.9156} + 5.754 \times 10^{-6} x^{0.5521} y^{0.1639})$$

حاصل تابع فوق عددی بر حسب دقیقه است.

متغیرهای مستقل این تابع به ترتیب عبارتند از:

x : تعداد هسته‌ها و بدون بعد

نسبت فاصله شبکه‌ای بر حسب کیلومتر به فاصله شبکه‌ای ۳۰

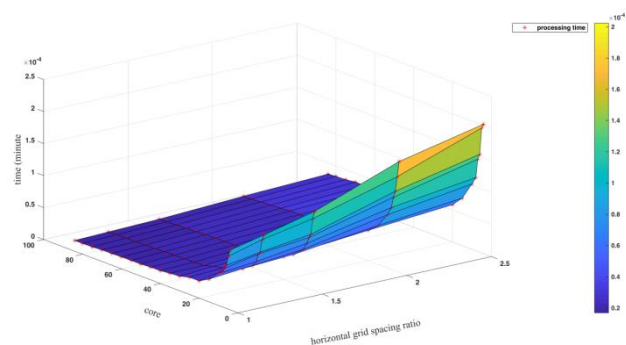
کیلومتر و بدون بعد

y : زمان شبیه‌سازی بر حسب ساعت که با تقسیم بر ۶ کسری بدون

بعد می‌شود

T : بعد می‌شود

no : تعداد نقاط شبکه و بدون بعد



شکل ۴: نتیجه اجرای مدل در فاصله شبکه‌ای مختلف و تعداد مختلف هسته CPU مورد استفاده

برای اینکه رابطه اصلی نسبت بین زمان اجرا و تعداد هسته‌های مورد استفاده در هر اجرا به دست آید جدول (۹) از زمان مورد نیاز برای محاسبه یک نقطه در هر فاصله شبکه‌ای به‌صورت زیر تهیه شد.

رابطه کلی به‌صورت تابعی از نسبت فاصله شبکه‌ای (x) و تعداد هسته‌ها (y) از جدول زیر که خلاصه کل آزمایش‌های زمان شبیه‌سازی مدل WRF است بر حسب دقیقه به‌دست می‌آید:

$$f(x, y) = 2.914 \times 10^{-4} xy^{-0.9156} + 5.754 \times 10^{-6} x^{0.5521} y^{0.1639}$$

جدول ۹: زمان اجرای هر نقطه در هر فاصله شبکه‌ای و تعداد هسته CPU

Number of CPU Core (WRF.exe Process)	زمان اجرای یک نقطه در هر فاصله شبکه‌ای				
	30km	25km	20km	15km	12km
4	9.36568218197797e-05	0.000105492833693650	0.000124796964265711	0.000173270412954443	0.000202325695062428
5	8.27448395635207e-05	9.69325733918576e-05	0.000112031384190284	0.000148444219979325	0.000195383497829194
6	6.74649257713192e-05	7.62489172289797e-05	8.65328964564430e-05	0.000124536054271457	0.000153000532589057
7	6.14125942897796e-05	7.20452179736350e-05	8.03484746648744e-05	0.000113997153852426	0.000144225453711943
8	5.44343659437348e-05	5.93285727804104e-05	7.03717187558344e-05	9.67650718244543e-05	0.000114299342016557
10	5.10812592197357e-05	5.05936132887850e-05	5.84159478734924e-05	8.32563926526253e-05	0.000102045580184990
15	3.54142473033908e-05	4.01589762627476e-05	4.47378738658004e-05	6.06213650460460e-05	7.42279144082194e-05
20	2.71544103030874e-05	3.77568624025506e-05	3.55779283820619e-05	4.73218066987056e-05	5.82409729525659e-05
25	2.70184028540640e-05	3.01610872112914e-05	3.24880512303499e-05	4.26290827881724e-05	5.31333427716204e-05
30	2.29800278292165e-05	2.68308839051092e-05	2.88684141742280e-05	4.12691398163419e-05	5.29218240163957e-05
35	2.17611918437379e-05	2.20630518492637e-05	2.78182293416975e-05	4.06549295959891e-05	4.35180878030364e-05
40	2.02441856815543e-05	2.19065504917054e-05	2.67377058362272e-05	3.63843930103666e-05	4.32197705147911e-05
45	1.98780117803375e-05	2.13569759570240e-05	2.54774840371905e-05	3.33370159642048e-05	4.04540153169916e-05
50	1.90567360304657e-05	2.12113932988302e-05	2.51834322840820e-05	3.24295104994437e-05	4.01026076638891e-05
55	1.90410428632707e-05	2.11786372007367e-05	2.45603226167806e-05	3.20441494832480e-05	3.88023150769895e-05
60	1.88318006340040e-05	2.01122442294674e-05	2.41729211007804e-05	3.19533989367719e-05	3.87087749103363e-05
65	1.83662366738855e-05	1.97774041156217e-05	2.31787461259848e-05	3.03067368470893e-05	3.64132486584149e-05
70	1.82354602805938e-05	1.94862387992342e-05	2.29010305813823e-05	2.96898961761139e-05	3.63685853355985e-05
75	1.72154044129186e-05	1.91404799860241e-05	2.21729024308278e-05	2.87534557545053e-05	3.43991856106572e-05
80	1.70113932393836e-05	1.86564176475298e-05	2.14821141854300e-05	2.76144706349645e-05	3.37781126122482e-05
84	1.69329274034086e-05	1.85508702203394e-05	2.09056793995743e-05	2.74697958507272e-05	3.27795081304102e-05

۶. بحث و نتیجه گیری

شبکه‌ای، یک رابطه ریاضی برای زمان اجرای یک نقطه مدل در شبکه به دست آمد. با شبیه‌سازی عددی در نرم‌افزار MATLAB یک رابطه کلی برای تمام نقاط فاصله شبکه‌ای به دست آمد. رابطه فوق با دریافت متغیرهای نقاط شبکه، فاصله شبکه‌ای، تعداد هسته‌های پردازشی و زمان شبیه‌سازی مدل، زمان اجرا مدل را بر اساس دقیقه محاسبه می‌نماید.

منابع

- Krishnan, S. P. T.; Veeravalli, B.; Krishna, V.-H.; Sheng, W.-C., 2014. Performance Characterisation and Evaluation of WRF Model on Cloud and HPC Architectures. IEEE Intl Conf on High Performance Computing and Communications, IEEE 6th Intl Symp on Cyberspace Safety and Security, IEEE 11th Intl Conf on Embedded Software and Syst (HPCC,CSS,ICSS).
- Mielikainen, J.; Huang, B.; Huang, A.-H.-L.; D. Goldberg, M., 2012. Improved GPU/CUDA based parallel Weather and Research Forecast WRF Single Moment WSM5- class Microphysics. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 5:1256 – 1265.
- Duran-Limon, H. A.; Flores-Contreras, J.; Parlavantzas, N.; Zhao, M.; Meulenert-Peña, A., 2016. Efficient execution of the WRF model and other HPC applications in the cloud, Earth Science Informatics, 9: 365-382.
- Naik, M. B.; Thomas, A. K., 2016. Optimization of Weather Model, International Journal on Recent and Innovation Trends in Computing and Communication , 4: 369-372.
- Hwang, G.; Chang S., 2017. Speed Up Weather Prediction on QCT Developer Cloud A Case study on Knights Landing Platform, IEEE 4th International Conference on Cyber Security and Cloud Computing(CSCloud), New York, NY, USA.

مدل WRF یک سیستم شبیه‌سازی اتمسفر است که برای اجرا به زیرساخت‌های رایانشی با کارایی بالا نیاز دارد. اغلب رایانش با حجم بالا در زیرساخت‌های ابر، خوشه، تور و مانند آن بدلیل سازگاری و هماهنگی بین اجزا و منابع پردازشی فراوان و مقرون‌به‌صرفه بودن آنها انجام می‌شود. در این زیرساخت‌ها از مجازی‌سازی برای مدیریت دسترسی به منابع پردازشی استفاده می‌شود. استفاده از مجازی‌سازی برای زیرساخت‌ها سربار نسبتاً زیادی ایجاد می‌کند که در برخی موارد برای اجرای برنامه‌های کاربردی نیازمند رایانش با کارایی بالا مناسب نیستند. استفاده از مجازی‌سازی سبک‌وزن مانند حامل‌ها می‌تواند همان مزایای دسترسی کنترل شده به منابع پردازشی را داشته باشد در عین حال سربار بسیار کمی برای زیرساخت ایجاد نماید. همچنین سرعت اجرای مدل و تخمین زمان تولید خروجی برای تحلیل‌های بعدی برای سازمان‌ها و مؤسسات هواشناسی و یا پروژه‌هایی که مدل را چند بار در روز اجرا می‌نمایند، دارای اهمیت فراوان است.

در این مقاله دو هدف دنبال شد: ابتدا به مقایسه اجرای مدل WRF بر روی حامل‌ها و مجازی‌سازی در مقایسه با اجرای مدل بر روی لینوکس بدون مجازی‌سازی پرداخته شد تا یک چارچوب مجازی‌سازی با سربار کم و بهینه برای اجرای مدل WRF معرفی گردد. سپس برای تخمین زمان اجرای مدل، آزمایش‌هایی انجام و با استفاده از شبیه‌سازی عددی یک رابطه ریاضی برای آن پیشنهاد گردید.

با مقایسه حامل‌های Singularity، Docker و ابرناظر KVM در مقایسه با لینوکس بدون مجازی‌سازی و محاسبه میانگین زمان اجرا در هر محیط، مشخص شد بهترین و نزدیکترین متوسط زمان اجرای مدل WRF به لینوکس بدون مجازی‌سازی، مربوط به محیط حامل Singularity است.

نتایج محاسبه سربار زمانی در هر محیط نشان می‌دهد کمترین میزان سربار زمانی اجرای مدل WRF در محیط‌های مجازی شده مربوط به حامل Singularity است.

نتایج آزمایش مقیاس‌پذیری نشان می‌دهد بهترین مقیاس‌پذیری که نزدیک به زمان اجرای مدل در لینوکس بدون مجازی‌سازی است، مربوط به محیط Singularity است.

برای محاسبه سرعت اجرای مدل WRF و یا تخمین زمان اجرای آن، با به دست آوردن زمان اجرای یک نقطه در هر فاصله

- Calinciuc, A.; Spoiala, C.; Turcu, C.; Filote, C., 2016. OpenStack and Docker: Building a high-performance IaaS platform for intensive social media applications, 13th International Conference on Development and Application Systems, Suceava, Romania.
- Stensrud, D. J., 2009. Parameterization Schemes Keys to Understanding Numerical Weather Prediction Models, Cambridge University Press, Cambridge CB2 8RU, UK.
- Arango C.; Dernat R.; Sanabria J., 2017. Performance Evaluation of Container-based Virtualization for High Performance Computing Environments, CoRR, abs/1709.10140
- Gomes J.; Bagnaschi E.; Campos I.; David M.; Alves L.; Martins J.; Pina J.; López-García L.; Orviz P.; 2018. Enabling rootless Linux Containers in multi-user environments: the udocker tool ,ScienceDirect, 232: 84-97.
- Kurtzer G. M.; Sochat V.; Bauer M. W.; 2017. Singularity: Scientific containers for mobility of compute, PloS one, 12(5):e0177459.
- Kovacs, A., 2017. Comparison of Different Linux Containers, 40th International Conference on Telecommunications and Signal Processing (TSP), Barcelona, Spain.
- Devendra, K.; Swati, K.; Shruti, K.; Shreejeet, J., 2017. HPC Cloud Burst Using Docker, International Research Journal of Engineering and Technology (IRJET), 4.
- Xavier, G. M.; Neves, V.M.; Rossi, D.R.; Ferreto, C.T.; Lange, T.; De Rose, C.A.F., 2013. Performance Evaluation of Container-Based Virtualization for High Performance Computing Environment, 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, Belfast, UK.
- Seo, K.; Hwang, H.; Moon, I.; Kwon, O.; Kim, B., 2014. Performance Comparison Analysis of Linux Container and Virtual Machine for Building Cloud, Advanced Science and Technology Letters, 66: 105-111.
- Gantikow, H.; Klingberg, S.; Reich, C., 2015. Container-Based Virtualization for HPC, 5th International Conference on Cloud Computing and Services Science, 543-550.