



صنعت فناوری اطلاعات به دنبال راهی برای بهینه‌تر کردن عملکرد مراکز داده است. یکی از رویکردها، بازنگری در معماری و نحوه ساخت سرورهای پردازشی است. تقسیم‌بندی سرورها (Server Disaggregation) یکی از راهکارهایی است که در این نوشتار آن را بازگو خواهیم کرد.

در پرونده ویژه شماره 205 ماهنامه شبکه به بررسی سرویس‌های پردازش ابری پرداختیم. تنوع در سرویس‌ها (از سخت‌افزار به عنوان سرویس گرفته تا نرم‌افزارهای آماده به سرویس‌دهی) بیانگر این واقعیت است که پردازش ابری به یک صنعت بزرگ و حیاتی بدل شده است. با این حال، تمام سرویس‌های ابری برای پیاده‌سازی به **مراکز داده** (دیتاسنتر) نیازمند هستند که این خود عامل بزرگ مصرف انرژی به شمار می‌رود. نکته جالب توجه این‌که در حال حاضر، حداقل 7 درصد تولید الکتریسیته جهان در **مراکز داده** به مصرف می‌رسد. انتظار می‌رود با رونق یافتن هر چه بیشتر صنعت پردازش ابری در سال‌های آینده، سهم **مراکز داده** از تولید الکتریسیته جهانی از میزان فعلی نیز فراتر رود.

## آغاز ماجرا

در ماه ژانویه سال جاری میلادی، یعنی حدود شش ماه قبل، پژوهشگران کمپانی IBM مقاله‌ای را در کنفرانس HiPEAC در منچستر منتشر کردند که یافته‌های آنان را در خصوص معماری کامپیوتر تقسیم‌بندی شده شرح داده است. این پژوهش بخشی از پروژه dReDBox است که توسط اتحادیه اروپا تامین مالی شده است. پروژه dReDBox که نام آن کوتاه شده یک عبارت انگلیسی به معنای «مرکز داده تقسیم‌بندی در یک باکس» است، خود بخشی از برنامه‌ای با عنوان چشم‌انداز 2020 است که با یک بودجه 80 میلیارد یورویی، در تلاش است اتحادیه اروپا را در پژوهش و نوآوری پیشرو کند.

**تقسیم‌بندی سرور** به معنای آن است که **سرورها** به اجزای پردازشی و حافظه کاری تقسیم شوند که این اجزا بر حسب نیاز به هر ورک‌لود (workload) تخصیص داده می‌شوند. در حال حاضر، سرورها واحد سازنده **مراکز داده** هستند و هیچ ورک‌لودی نمی‌تواند منابعی بیشتر از آنچه را که به یک سرور اختصاص داده شده، استفاده کند. (مگر این‌که یک سرور جدید به آن داده شود.) همچنین سرورها به سادگی نمی‌توانند منابعشان را با یکدیگر به اشتراک بگذارند. به گفته دکتر آندرا ریل، پژوهشگر کمپانی IBM، «برنامه‌هایی که در **مراکز داده** تعبیه می‌شوند، معمولاً یک عدم تناسب بزرگ در میزان استفاده از منابع دارند. برخی برنامه‌ها نیاز فراوانی به پردازنده و درعین حال نیاز کمی به حافظه دارند، و از سوی دیگر برخی برنامه‌ها حافظه‌ای به میزان چهار برابر بیشتر از پردازنده مصرف می‌کنند.» اگر این موضوع که تخصیص منابع در **مقیاس سرور** انجام می‌شود و عدم تناسب برنامه‌ها در مصرف منابع را کنار هم قرار دهیم، نتیجه‌اش این می‌شود که بسیاری از سرورها پردازنده یا حافظه بلااستفاده خواهند داشت و این به معنای هدررفتن منابع است. آمار و ارقام نشان می‌دهد، 16 درصد از توان پردازشی و 30 درصد از حافظه به

این صورت هدر می‌رود. لازم به ذکر است، وقتی یک سرور روشن می‌شود، همه بخش‌های آن برق مصرف می‌کنند. اما چه می‌شد اگر یک مرکز کنترل نرم‌افزاری داشتیم که به هر سرور همان میزان پردازنده یا حافظه تخصیص دهد که ورک‌لود مستقر روی آن نیاز دارد؟ و این آغاز ماجرای تقسیم‌بندی سرورها است.

## مطلب پیشنهادی



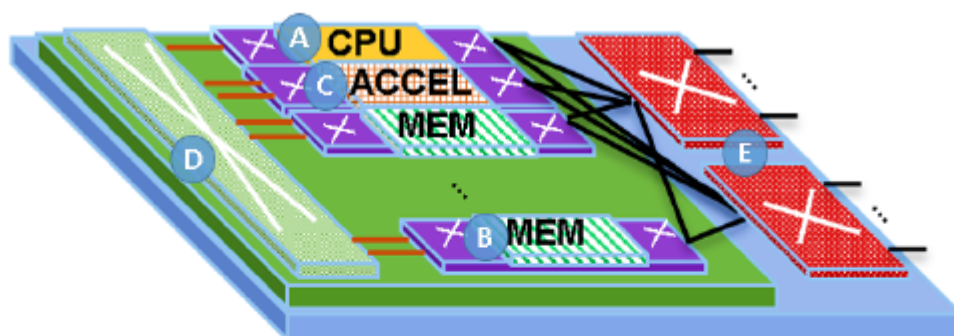
ذخیره‌سازها سریع‌تر، چاق‌تر و ارزان‌تر می‌شوند  
**8 تغییر بزرگ صنعت ذخیره‌سازی مراکز داده در سال 2018**

## تقسیم‌بندی پردازش و حافظه

پروژه dReDBox در تلاش است ماژول‌های پردازنده و حافظه موسوم به Brick را در معماری سرورها استفاده کند. Brick در لغت به معنای آجر است که در این پروژه نیز ماژول‌های پردازنده و حافظه همانند آجرهای ساختمان مرکز داده هستند. بریک‌ها با لینک‌های پرسرعت به یکدیگر متصل می‌شوند؛ بنابراین می‌توان هر تعداد بریک پردازنده را به هر تعداد بریک حافظه که مورد نیاز است، متصل کرد. در واقع این رویکرد به خوبی مفهوم معماری ماژولار را نمایان می‌کند. با صرف‌نظر از جزئیات پیاده‌سازی، با این رویکرد می‌توان به هر ورک‌لود به میزانی که توان پردازشی و حافظه نیاز دارد، بریک اختصاص داد و بنابراین هدر رفت منابع به صفر می‌رسد. همچنین در صورت پایان کار ورک‌لود، منابع آزاد شده و برای سایر ورک‌لودها قابل تخصیص می‌شود.

البته دو نوع بریک دیگر نیز در این سیستم تعبیه شده است: بریک‌های شتاب‌دهنده که از نوع GPU یا FPGA هستند و برای برنامه‌های از نوع یادگیری ماشین که مقدار زیادی پردازش موازی دارند، سرعت کار را افزایش می‌دهند. بریک آخر نیز کنترلر است که همه بریک‌های دیگر را کنترل می‌کند. معماری dReDBox در شکل (1) نمایش داده شده است. در این شکل، بلوک‌های A، B، C، D و E به ترتیب بریک پردازنده، حافظه، شتاب‌دهنده، کنترلر داخلی و کنترلر خارجی هستند.

یک مزیت معماری ماژولار، امکان ارتقای آسان است. اپراتور به سادگی می‌تواند بریک‌های پردازشی موجود را با نمونه بهتر جایگزین کند یا این‌که بریک‌های حافظه موجود را با بریک با حجم بیشتر جایگزین کند. در این صورت دیگر نیازی به دور انداختن یک سرور نیست. یک جزء کلیدی در این معماری فناوری ارتباطی بین بریک‌ها است. بدیهی است اگر ارتباط مناسب بین بریک‌ها برقرار نشود، کارایی مجموعه کاهش می‌یابد و اهداف پروژه تامین نخواهد شد؛ بنابراین طراحی لینک‌های پرسرعت و با تاخیر کم مورد توجه بوده که در بخش بعدی نحوه دستیابی به آن بررسی خواهد شد.



1 □□□

- A** Compute Module (dCOMPUBRICK)
- B** Memory Module (dMEMBRICK)
- C** Acceleration Module (dACCELBRICK)
- D** On-tray MACOM Cross-connect
- E** Off-tray POLATIS Cross-connects

## معماری با تاخیر کم

در طراحی و ساخت نمونه اولیه، طراحان پروژه یک ماتریس سویچ الکتریکی در نظر گرفته‌اند که ارتباط میان بریک‌های درون یک طبقه (tray) از رک را برقرار می‌کند. یک ماتریس سویچ نوری نیز برای ارتباط بین بریک‌های دو طبقه مختلف تعیین شده است. بر خلاف عرف معمول، این سویچ‌ها به صورت راهیابی مداری (circuit switched) کار می‌کنند، به این معنا که پس از پیکربندی، یک ارتباط اختصاصی بین دو بریک برقرار می‌شود. یادآوری می‌شود در یک شبکه راهیابی بسته‌ای مثل اترنت، مسیر مجزایی برای هیچ جریانی وجود ندارد و آدرس بسته‌ها ملاک قرار می‌گیرد. در نقطه مقابل، شبکه تلفن ثابت یک نمونه از شبکه‌های راهیابی مداری است که در آن یک ارتباط اختصاصی بین دو طرف تماس برقرار می‌شود. به همین دلیل کیفیت ارتباط تلفنی تنها در صورت خرابی تجهیزات شبکه دچار افت می‌شود.

## مطلب پیشنهادی

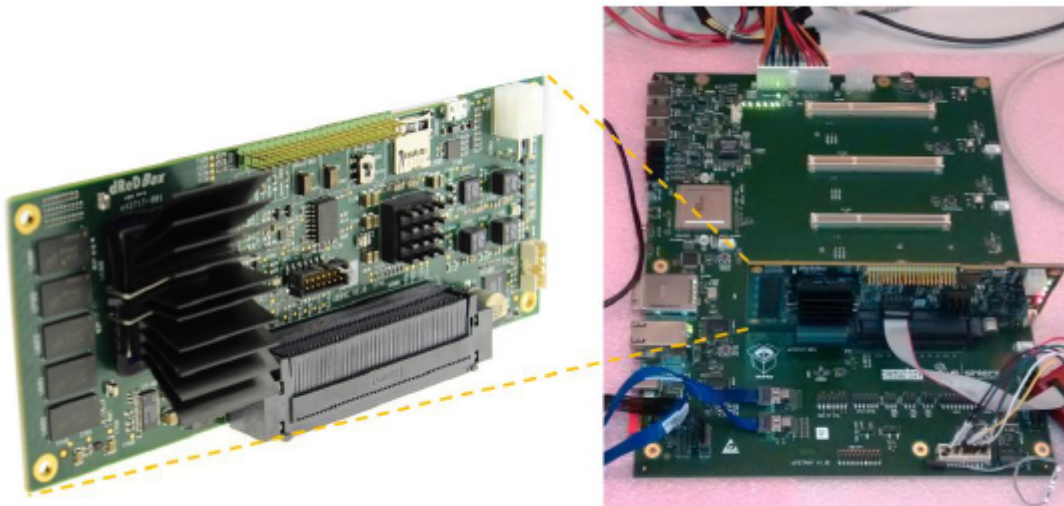


### به این 5 دلیل زمان به‌روزرسانی سرور شما فرارسیده است!

به گفته دکتر ریل، نگرش راهیابی مداری با هدف ایجاد معماری با تاخیر کم همخوانی دارد: «راهیابی مداری در مقایسه با راهیابی بسته‌ای این امکان را می‌دهد که با تاخیر بسیار کمتری درخواست‌های دسترسی به حافظه را از بریک‌های پردازنده به بریک‌های حافظه برقرار کنید.» در همین راستا، دکتر ریل ادعا می‌کند، با یک سخت‌افزار سطح پژوهشی (که هنوز بهینه نشده است) توانسته‌اند تاخیر انتها به انتها زیر یک میکروثانیه را برای دسترسی به حافظه ریموت فراهم کنند. با یک پردازنده عملیاتی که با حداکثر سرعت کلاک کار می‌کند، تاخیرهای کمتر نیز امکان‌پذیر است. یک مزیت دیگر راهیابی مداری این است که شرایط برای نرم‌افزار همانند یک سرور عادی به نظر می‌رسد، که در آن CPU مستقیم به حافظه متصل است. دکتر ریل در همین رابطه می‌گوید: «ما از برخی افزونه‌های موجود سیستم‌عامل مانند پشتیبانی NUMA در لینوکس به منظور دسترسی غیریکسان به حافظه استفاده می‌کنیم تا برای برنامه‌هایی که از تاخیر در دسترسی به حافظه آگاهی دارند، مقدار آن را تعیین کنیم. اما بقیه برنامه‌ها که چنین آگاهی را ندارند، می‌توانند فرض کنند که حافظه محلی است و نیازی به دانستن فاصله حافظه ندارند.» با تفاسیر فوق، به نظر می‌رسد انتخاب راهیابی مداری برای این سیستم یک انتخاب هوشمندانه بوده است، با این حال باید در عمل کارایی را بررسی کرد. در بخش بعدی به نحوه ارزیابی سیستم dRedBox می‌پردازیم.

## ارزیابی dRedBox در عمل

برای ارزیابی عملکرد سیستم در عمل، یک نمونه اولیه با بهره‌گیری از سیستم‌روی چیپ (SoC) Xilinx Zynq + Ultrascale مبتنی بر ARM ساخته شده است. بریک‌های پردازنده، حافظه ناچیزی دارند اما بریک‌های حافظه مقدار زیادی حافظه DDR4 دارند که به بریک‌های پردازنده سرویس می‌دهند. شکل 2- سمت چپ یک بریک را نشان می‌دهد که روی یک طبقه (tray) از سیستم (شکل 2- سمت راست) قرار می‌گیرد. بریک نشان داده شده در این شکل بر اساس معماری که در شکل (1) به تصویر کشیده شده، ساخته شده است. نمونه اولیه در مقیاس کوچکی ساخته شده و تنها سه طبقه دارد. با این حال، تیم dRedBox توانسته یک سری ورک‌لود واقعی پردازش ابری را روی همین نمونه کوچک اجرا کند و خروجی بگیرد. به گفته آن‌ها، این سیستم می‌تواند از لحاظ کارایی با یک سرور معمول مطابقت کند، در حالی که بین 25 تا 50 درصد منابع کم‌تری مصرف کند. آن‌ها امیدوارند در انتهای پروژه به جایی برسند که بتوانند عملکرد یک رک کامل را تست کنند. مسئله دیگر در ایجاد یک سیستم عملیاتی، سازگاری آن با زیرساخت موجود است. بنابراین، لایه کنترلی باید اینترفیس‌های مناسبی داشته باشد. در این پروژه لایه کنترلی به صورت یک سرور خارجی پیاده‌سازی شده و با REST API قابل مدیریت است. این اینترفیس می‌تواند به صورت دستی یا به صورت یکپارچه با نرم‌افزارهای سطح انتزاع بالا همچون اپن‌استک برای کار با ماشین‌های مجازی یا کوبرنت برای کار با کانتینرها مورد استفاده قرار گیرد.



## HPE و اینتل نیز وارد بازی شده‌اند

تیم dRedBox تنها گروهی نیست که به **تقسیم‌بندی سرورها** برای افزایش کارایی روی آورده است. یک نمونه دیگر پروژه The Machine از کمپانی HPE است که با هدف ایجاد یک سیستم که بتواند یک فضای حافظه بسیار بزرگ را فراهم کند، ایجاد شد. فضای بزرگ حافظه برای اجرای برنامه‌های داده‌کاوی ضروری است. این سیستم از بلوک‌های پردازنده و حافظه مجزا بهره می‌برد که در یک‌سری کلاستر جای گرفته‌اند. ارتباط بین کلاسترها نیز از طریق یک فابریک حافظه برقرار می‌شود. در نمونه نمایشی که سال گذشته عرضه شد، HPE از لینک‌های فیبر نوری برای اتصال 40 گره به یکدیگر با 160 ترابایت حافظه مشترک استفاده کرد. در همین زمان، شرکت اینتل پروژه خود موسوم به RSD (سرنام Rack Scale Design) را راه‌اندازی کرد که اهداف مشترکی با سایرین داشت. البته تمرکز اینتل روی جداسازی فضای ذخیره‌سازی و سرورها بود، درحالی‌که پروژه‌های IBM و HPE چنانچه ذکر شد، بر جداسازی پردازنده و حافظه کاری تمرکز کرده است. اینتل API مدیریتی خود را با نام Redfish طراحی کرده که عمدتاً برای شناسایی منابع کاربرد دارد.

## مطلب پیشنهادی



سرورهای سنتی در تعریف، مفهوم و ساختار دگرگون شدند  
7 فناوری که سرورها را متحول کردند

## سخن آخر

با وجود تلاش‌های فراوان چندین شرکت سازنده سخت‌افزار در جهت ارائه یک معماری نوین برای **مراکز داده** موسوم به فناوری **سرور تقسیم‌بندی‌شده**، ورود این فناوری به بازار چشم‌انداز روشنی ندارد. به‌رحال شرکت‌ها در پذیرش یک فناوری جدید و متفاوت از آنچه که در بازار عرضه می‌شود، محافظه‌کارانه رفتار می‌کنند. به‌خصوص در بازار پردازش ابری که هزینه ریسک آن بسیار بالا است. شاید اگر شرکت‌هایی چون گوگل، آمازون یا فیس‌بوک که از مصرف‌کنندگان بزرگ بازار به شمار می‌روند، تمایلی به این فناوری جدید نشان دهند، می‌توانیم انتظار ورود آن را به بازار داشته باشیم.

منبع:

[Networkworld](#)

تاریخ انتشار:

20 مرداد 1397

---

نشانی منبع:

<https://www.shabakeh-mag.com/networking-technology/13596/%D8%AA%D9%82%D8%B3%DB%8C%D9%85%E2%80%8C%D8%A8%D9%86%D8%AF%DB%8C-%D8%B3%D8%B1%D9%88%D8%B1%D9%87%D8%A7-%DA%86%DA%AF%D9%88%D9%86%D9%87-%D8%A8%D9%87-%D8%A7%D9%81%D8%B2%D8%A7%DB%8C%D8%B4-%DA%A9%D8%A7%D8%B1%D8%A7%DB%8C%DB%8C-%D9%85%D8%B1%D8%A7%DA%A9%D8%B2-%D8%AF%D8%A7%D8%AF%D9%87-%D9%85%DB%8C%E2%80%8C%D8%A7%D9%86%D8%AC%D8%A7%D9%85%D8%AF%D8%9F>