گزارش ارزیابی کارایی سیستم‌های اینترنت اشیا پزشکی و اینترنت اشیا براساس مجموعه داده‌های CICIoMT2024 و مدلینگ ریاضیاتی

استاد ناظر: آقای دکتر مهدی امینیان

علیرضا سلطانی نشان

۲۳ دی ۱۴۰۳

فهرست مطالب

مجوز

مهم‌ترین انگیزه برای توسعه این تحقیق وجود کمبود در داده‌های موجود ارزیابی کارایی تجهیزات اینترنت اشیا پزشکی و پیشرفت امنیتی تمام شبکه‌هایی که در خصوص جریان‌های داده‌ای و پردازش داده‌های پزشکی کار می‌کنند، می‌باشد بخصوص برای دستگاه‌های اینترنت اشیا پزشکی به دلیل اطلاعات حیاتی‌ای که می‌توان به واسطه آن‌ها از بیماران با بیماری‌های مختلف مانیتور و دریافت کرد.

یکپارچه‌سازی سیستم‌های IoT با سیستم‌های ابری چالش‌های زیادی داشته مثل:

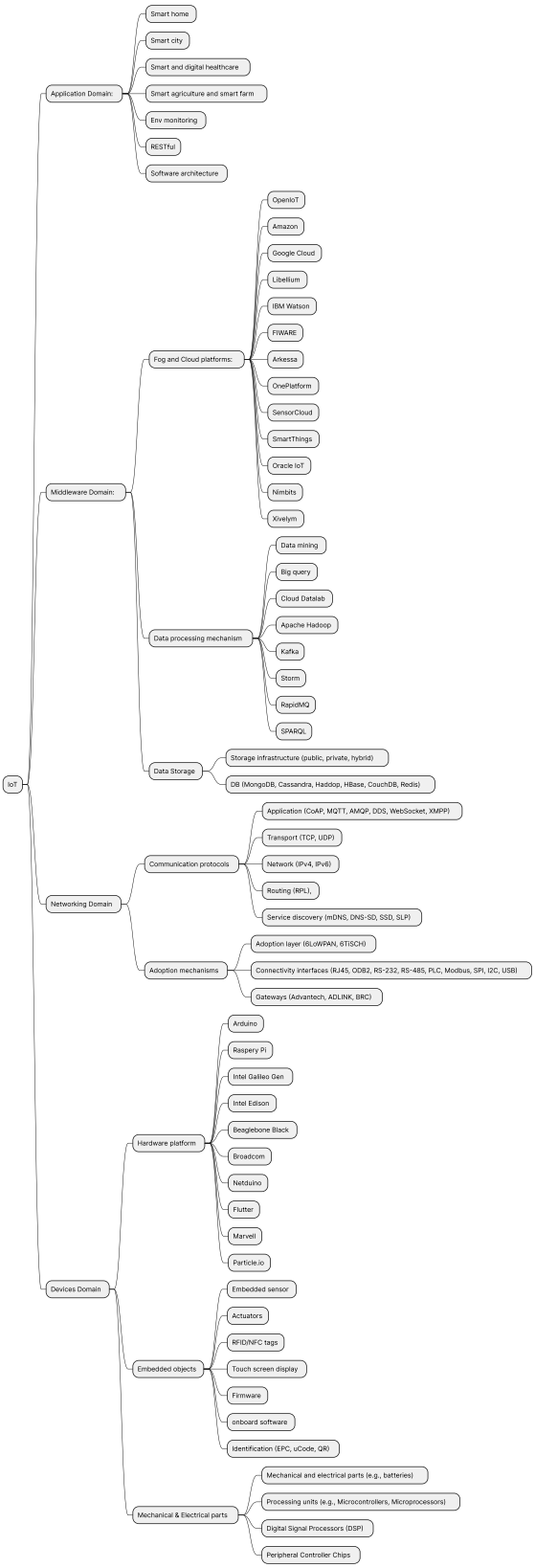
- تاخیرهای شبکه‌ای
- گذردهی
- مصرف انرژی
- قابلیت اطمینان

یه سری مفاهیم جدیدی در حوزه پردازش‌ها مطرح شده که حتی می‌تواند کاربردهای مختلفی در استفاده از اینترنت اشیا باشه. این مفاهیم جدید مثل Fog computing، edge computing، mobile edge computing، mobile cloud computing و Cloudlet ها هستش.
در این مقاله یک مدل ریاضیاتی برای توصیف رسمی سیستم‌های IoT ارائه داده شده است. علاوه‌بر این یک ارزیابی آنالیز شده برای طراحی این سیستم‌ها با استفاده از مطابقت با معماری، تکنولوژی‌ها، پروتکل‌ها و مدل‌های یکپارچه‌سازی برای بهینه‌سازی عملکرد نیز ارائه می‌دهد.
Approach of this article:
بعد از خوندن این مقاله به یک روش بهینه برای بهینه‌سازی کارایی مبتنی بر فرایندهای offloading مانند load balancing آشنا می‌شیم.
مدلینگ ریاضیاتی سیستم‌های IoT یک نمایی از سیستم ایجاد می‌کنند که به فهمیدن المان‌ها، تعاملاتشون، و رفتارهاشون کمک می‌کند.

۱. مدل مفهومی یا conceptual model یک ساختار سطح بالایی برای توصیف عملیاتی است که در سیستم‌های IoT انجام می‌شود.

۲. مدل رفتاری یا behavioral model ممکنه شامل جزیئیات باشه. مثل جریان داده بین المان‌ها.

به طور کلی مودلینگ به مشخص شدن و پاسخ به مسائل مربوط به کارایی کمک بسزایی می‌کنه و اجازه میده که سیستم‌ها بهینه‌تر، کاراتر و مطمئن‌تر باشن.
هر موقع در مورد مودلینگ یک سیستم IoT صحبت میشه در حقیقت قراره یه چهارچوبی درست بشه که بتونیم باهاش تست کنیم، تایید یا validation انجام بدیم و یا بتوانیم سیستم را به تقاضاهایی که داریم optimize کنیم.
فرایند سیستم‌های IoT معمولاً شامل شناسایی، دریافت اطلاعات ،(Sensing) فعالیت‌های تحت شبکه، و محاسبات کوچک هستش که باعث میشه با محیط فیزیکی و هر اشیایی ارتباط برقرار کند.
در این مقاله سیستم‌های IoT با کاربردی که دارند به ۴ دسته تقسیم می‌شوند:



شکل ۱: دسته‌بندی سیستم‌های IoT

عمومیت جریان داده در سیستم‌های IoT:

- دریافت داده

- انتقال داده
- پردازش داده
- ذخیره‌سازی داده
- آنالیز و معنادار کردن داده

۱ شاخش‌های محاسبه و ارزیابی عملکرد

ثابت‌ها	تعاریف
D_{rate}	نرخ ورود اطلاعات
E_{dev}	مصرف انرژی
$T_{exe.}$	تاخیر سرویس‌دهی
$IoT_{sys_{sp}}$	مشخصات سیستم IoT
t_{ws}	زمان ورکلودها
k	تعداد دستگاه‌ها
$IoT_{sys}(P) = \{P_{i_{req}}, P_i\}$	نیازمندی مشخص کارایی $P_{i_{req}}$ جایی که i^{th} P_i مقدار از کارایی است
$P_i = \langle T_{exe.}, E_{dev} \rangle$	مدت زمان سرویس‌دهی

جدول ۱: تعریف ثابت‌های مورد استفاده در فرمول‌ها

۱.۱ فرمول شانون

نرخ داده‌ها می‌تواند از طریق فرمول شانون محاسبه شود.

$$D_{rate} = B_{i,j} \log_2(1 + \frac{|h_{ij}|^2.P_{tx}}{P_{Nj}}) \quad (۱)$$

- B_{ij}: پهنای باند
- h_{ij}: بهره کانال بین دستگاه مبدا و مقصد که نشان می‌دهد سیگنال چگونه در مسیر بین فرستنده و گیرنده تقویت یا تضعیف می‌شود.
- P_{tx}: توان ارسال
- P_N: میزان نویز مقصد

کاربرد زیادی در سناریوهایی دارد که در آن یک ارسال کننده و یک دریافت کننده وجود دارد.

۲.۱ فرمول محاسبه بار سیستم یا System load

مجموع بار سیستم از طریق فرمول زیر بدست می‌آید:

$$D = \sum_{k=1}^{N} D_{rate,k} \times T_{w,k} \quad (۲)$$

- $D_{rate,k}$: نرخ تولید داده توسط دستگاه IoT
- $t_{w,k}$: مدت زمان ورودی در دستگاه IoT یا به عبارتی دیگر، مدت زمانی که طول می‌کشد یک دستگاه IoT ورودی را دریافت و سپس آن را پردازش و هندل کند.

۳.۱ تاخیر سرویس‌دهی یا Service latency

مسئله service latency یا service execution time ($T_{exe.}$) مدت زمانی است که طول می‌کشد سیستم IoT تمام درخواست‌های پردازشی و ارتباطی را اجرا کند (the total application exe time). مدت زمان کل مصرف شده از، مدت زمان سرویس یک ریکوئست به مدت زمان تمام تسک‌هایی که با موفقیت پردازش شده‌اند. بنابراین، این فاصله زمانی بین درخواست برنامه و بدست آوردن نتایج می‌باشد.

۱.۳.۱ بخش‌هایی که زمان سرویس‌دهی دارند

- مدت زمان انتقال داده از دستگاه IoT به زیر ساخت فاگ
- مدت زمان انتقال داده از دستگاه IoT به سرورهای ابری
- مدت زمان انتقال داده از فاگ به کلاد
- مدت زمان انتقال اعلانات از کلاد به فاگ
- مدت زمان انتقال اعلانات از کلاد به دستگاه‌های IoT
- مدت زمان انتقال اعلانات از فاگ به دستگاه IoT
- مدت زمان محاسبات در دستگاه IoT
- مدت زمان محاسبات در لایه فاگ
- مدت زمان محاسبات در سرورهای ابری

نکته: مدت زمانی که برای هر کاری در سیستم‌های IoT سپری می‌شود به نوع و شیوه پیاده‌سازی معماری دستگاه‌ها و نرم‌افزار بخش‌ها بستگی دارد و می‌تواند کاملاً متفاوت باشند. عموماً سرویس لیتنسی بین المان‌های سیستم IoT توزیع شده هستش و شامل دستگاه‌های اینترنت اشیا، شبکه‌ها و سیستم‌های پردازشی می‌شود.
تاخیر سرویس‌دهی:

$$T_{exe.} = T_{cm} + T_{cp} \tag{۳}$$

- T_{cm}: مدت زمان تاخیر در ارتباطات
- T_{cp}:مدت زمان تخیر در پردازش

مدت زمان اجرا بایستی کمتر از زمان‌بندی تسک‌ها در فاگ یا کلاد باشد. یعنی سرویس تایم باید کمتر از نیازمندی‌های IoT application (T_req) باشد.
برای کاهش service latency از فرمول زیر بایستی پیروی کند:

$$Objective : \min(T_{exe.}) = T_{cm} + T_{cp} \leq T_{req} \tag{۴}$$

موقعی که داری در مورد Execution time می‌نویسی فرمول communication latency رو باید داشته باشی که بگی از کجا بدست میاد. و همچنین فرمول computation latency رو هم بعدش. یعنی ثابت‌ها خودشون از زیر ثابت‌های اصلی بدست میان که جمع میشن و میشه T_{exe}.

۲.۳.۱ زمان ارتباطی

$$T_{cm} = \sum_{i=1}^{N}(d_{proc} + d_{queue} + d_{trans} + d_{prop}) \quad (۵)$$

- d_{prop}: تاخیر پردازشی
- d_{queue}: تاخیر در صف
- d_{trans}: تاخیر انتقال
- d_{prop}: تاخیر توزیع

تاخیر مربوط به Propagation مجموع زمان مورد نیاز برای داده جهت ارسال از منبع به مقصد که مبتنی بر طول لینک فیزیکی و سرعت رسانا می‌باشد.

$$d_{trans} = \frac{P_s}{R_L} \quad (۶)$$

که در آن:

- P_s: اندازه بسته در واحد `bits`
- R_L: سرعت لینک ارتباطی `bps`

$$d_{prop} = \frac{l_{ij}}{c} \quad (۷)$$

- l_ij: لینک فیزیکی
- c: سرعت توزیع `media`

۳.۳.۱ زمان پردازشی

$$T_{cp} = T_L + \sum_{i=1}^{k} t_{offi} \quad (۸)$$

که در آن:

- t_L: اجرا و پردازش‌های داخلی
- t_offi: اجرا و پردازش‌های خارج از دستگاه IoT مانند برنامه‌هایی که در سیستم‌های ابری یا Fog مستقر شده‌اند که وظیفه پردازش تسک‌های Offloading را دارند.

به بیان دیگر می‌توان آن را به صورت مدل زیر محاسبه کرد:

$$T_{cp} = t_L + t_F + t_C \quad (۹)$$

$$T_{cp} = t_L + \max_{i=1,\dots,k} t_{F_i} + \max_{j=1,\dots,n} t_{C_i} \quad (۱۰)$$

که در آن:

- t_L: مدت زمان پردازش‌های داخلی
- t_{F_i}: مدت زمان پردازش در نود i^{th} در Fog
- t_{C_i}: مدت زمان پردازش در سرور i^{th} ابری

عموماً مصرف پردازشی بستگی به سرعت و معماری پردازنده مرکزی (CPU)، حافظه رم (RAM)، سرعت حافظه ذخیره‌ساز (HDD) یا (SSD)، سرعت پردازنده گرافیکی یا (GPU) و غیره. دارد.

۴.۳.۱ زمان پردازش محلی t_L یا زمان پردازش در هر زیر سیستم t_{pi}

برای بدست آوردن زمان پردازش در هر زیر سیستم از فرمول زیر استفاده می‌شود:

$$t_{pi} = \frac{I_{CC_i}}{f_{cpu,i}} \tag{۱۱}$$

- t_{pi}: زمان پردازشی در زیر سیستم i
- I_{CC_i}: تعداد سایکل‌های CPU که برای اجرای یک برنامه نیاز است.
- $f_{cpu,i}$: نرخ کلاک (فرکانس کاری CPU) زیر سیستم i

$$t_{Pi} = t_{CPU_i} + t_{I/O_i} \tag{۱۲}$$

۵.۳.۱ تابع محاسبه CPU time

مدت زمانی که در CPU برای اجرا برنامه در نظر گرفته می‌شود به دو دسته تقسیم می‌شود:

۱. User CPU time

۲. System CPU time: t_{OS}

محاسباتی که در CPU time انجام می‌شود خالصانه در قسمت پردازشگر مرکزی صورت می‌گیرد و هیچ محاسبه جانبی مانند مدت زمان I/O و مدت زمان اجرای دیگر برنامه‌ها در نظر گرفته نمی‌شود.

$$t_{cpu_i} = \frac{I_{CC_i}}{f_{cpu_i}} + t_{OS} = I_{CC_i} \times t_{cc_i} + t_{OS} \tag{۱۳}$$

حاصل این تابع معمولاً بسیار کوچک است و می‌تواند نادیده گرفته شود زیرا به سمت صفر میل می‌کند ($t_{OS} \to 0$). به همین خاطر بیشتر روی User CPU time تمرکز می‌کند که توسعه‌دهنده بر روی آن کدهای خود را اجرا می‌کند و سیستم IoT را راه‌اندازی می‌کند.

تابع مطرح شده بر اساس قدرت محاسباتی دستگاه $i(f_{cpu_i})$ و تعداد کلاک CPU برای اجرای یک برنامه (I_{CC_i}) می‌باشد. مقدار f_{cpu_i} بر واحد (Hz) می‌باشد و t_{cc_i} زمان چرخه کلاک است. لازم به ذکر است که I_{CC_i} به نوع دستورالعمل که شامل اندازه داده ورودی، زبان برنامه نویسی، میزان پیچیدگی الگوریتم نرم‌افزاری مورد استفاده، و دیگر موارد می‌باشد.

بخش CPI (Clock cycles per instruction) به عنوان میانگین تعداد چرخه کلاک است که هر دستورالعمل به آن نیاز دارد. اگر I_{app_j} تعداد دستورالعمل‌ها برای یک برنامه باشد آن وقت I_{CC_i} از طریق معادله زیر بدست می‌آید.

$$I_{CC_i} = \sum_{j=1}^{k} I_{app_j} \times CPI_j \tag{۱۴}$$

۶.۳.۱ زمان پردازش محلی با توجه به اندازه داده (D)

$$I_{CC_i} = X \times D \tag{۱۵}$$

- D: اندازه ورودی داده بر حسب بیت
- X: شدت پردازش (تعداد چرخه‌های مورد نیاز برای هر بیت داده)

بنابراین خواهیم داشت:

$$t_{pi} = \frac{\beta_i \times D \times X}{f_{cpu,i}} \tag{۱۶}$$

که در آن β_i درصد داده‌ای است که در سیستم i پردازش می‌شود.

۴.۱ مصرف انرژی

۱.۴.۱ مجموع مصرف انرژی E_{dev}

برای محاسبه مصرف کل انرژی در سیستم‌های IoT می‌توان از فرمول زیر استفاده کرد:

$$E_{dev} = F(E_{cp}, E_{cm}, E_{idle}, E_{other}) \tag{۱۷}$$

- E_{cp}: انرژی مصرف شده طی محاسبات
- E_{cm}: انرژی مصرف شده در طی ارتباطات
- E_{idle}: انرژی مصرف شده در حالت نرمال و بیکار سیستم
- E_{other}: انرژی مصرف شده توسط بقیه فرایندها مانند سنسورها، صفحه نمایش، کارت گرافیک و غیره.

یکی از مهم‌ترین چالش‌های دستگاه‌های IoT مربوط به مصرف باتری آن‌ها می‌باشد. در بعضی مواقع دستگاه‌های IoT از باتری‌هایی استفاده می‌کنند که شرایط جایگزین کردن آن‌ها وجود ندارد. هر دستگاه IoT حتی در حالت بیکار انرژی بابت، پردازش داده‌ها، ارسال و دریافت داده‌ها مصرف می‌کنند. انرژی موجود $E_{dev}(t)$ در طی زمان کاهش پیدا می‌کند. به همین ترتیب انرژی باقی‌مانده $E_{dev}(r)$ یا مدت زمانی که سیستم می‌تواند روشن بماند از طریق فرمول زیر بدست می‌آید.

$$E_{dev}(r) = E_{dev}(i) - E_{dev}(t) \tag{۱۸}$$

- $E_{dev}(i)$: مقدار اولیه انرژی دستگاه

مقدار باتری باقی‌مانده $T(sys)$ به میزان ظرفیت باطری یا انرژی باقی‌مانده و انرژی مورد نیاز دستگاه برای انجام تمام سرویس‌های دستگاه، بستگی دارد. انرژی مصرفی وابسته به قدرت مورد نیاز برای پردازش‌های داخلی P_{cp} انتقال داده‌ها P_{cm} و بقیه فرایندها P_{other} می‌باشد.

$$T(sys) = \frac{E_{dev}(r)}{P_{cp} + P_{cm} + P_{other}} \tag{۱۹}$$

یکی دیگر از چالش‌های دستگاه‌های IoT مربوط به منبع‌تغذیه آن‌ها می‌باشد. در مواقعی که دستگاه‌های IoT از باتری استفاده می‌کنند و به دلیل محیط‌های مختلف شرایط به گونه‌ای است که امکان تعویض باتری وجود ندارد، محدودیت‌های باتری اغلب به عنوان شاخص طول عمر دستگاه‌های IoT استفاده می‌شود و می‌تواند به عنوان یکی از مهم‌ترین معیارهای QoS مورد استفاده قرار گیرد. هدف اصلی در این دستگاه‌ها این است که مصرف انرژی به حداقل برسد و طول عمر کلی سیستم به بیشترین حد ممکن.

۲ مدل‌های ارزیابی کارایی