

۱- تولید وظایف

- هر وظیفه τ_i در قالب یک گراف می‌باشد.
 - هر گره از گراف نشان‌دهنده بخشی از وظیفه است که باید اجرا شود.
 - هر یال نشان‌دهنده وابستگی بین بخش‌های مختلف اجرا است.
- گراف وظیفه بر اساس روش Erdős-Rényi به صورت $G(|V_i|, p)$ ایجاد می‌شود. به این صورت که در این پروژه $p = 0.1$ در نظر گرفته می‌شود. p نشان‌دهنده احتمال وجود یال بین گره‌ها می‌باشد.
 - بعد از ایجاد گراف، یک گره source به عنوان ریشه به ابتدای گراف اضافه کنید و یک گره sink نیز به انتهای گراف اضافه کنید. بنابراین گراف مورد نظر تنها یک ریشه و یک فرزند دارد.
 - زمان اجرای گره source و گره sink برابر با صفر می‌باشد.
- تعداد گره‌های هر وظیفه عددی تصادفی از بازه $[5, 20]$ می‌باشد
- بدترین زمان اجرای هر گره عددی تصادفی بین $[13, 30]$ می‌باشد.
- گره‌های گراف ترکیبی از گره‌های soft real-time و hard real-time هستند. به این معنی که اجرای گره‌های hard حیاتی می‌باشد و در صورتی که deadline خود را miss کنند فاجعه رخ می‌دهد. در حالی که، اجرای گره‌های soft ضروری نمی‌باشد و در صورت اجرای هر بخش از این گره‌ها کیفیت خدمات افزایش می‌یابد.
 - گره‌های hard real-time نمی‌توانند فرزند گره‌های soft real-time باشند.
- بعد از انتساب زمان اجرا، به هر یک از گره‌ها، مجموع زمان اجرای تمامی گره‌ها، معادل با بدترین زمان اجرای کل وظیفه می‌شود.

$$C_i = \sum_{v \in V_i} c(v)$$

- وظایف به صورت پریودیک می‌باشند.
- طول مسیر بحرانی \mathcal{L}_i برابر با طولانی‌ترین مسیر در گراف مربوط به وظیفه τ_i می‌باشد.

$$\mathcal{L}_i = \max_{\pi \in G_i} \{len(\pi)\}$$

- مهلت نسبی D_i هر وظیفه τ_i به این صورت تولید می‌شود: بعد از اینکه طول مسیر بحرانی \mathcal{L}_i محاسبه شد، نسبت \mathcal{L}_i/D_i عددی تصادفی بین $[0.125, 0.25]$ می‌باشد. به همین صورت D_i محاسبه می‌گردد.
 - هر گراف دارای یک D_i می‌باشد و مهلت نسبی هر یک از گره‌های گراف نیز برابر با این مقدار می‌باشد.
- مقدار دوره تناوب T_i برابر با D_i می‌باشد.
- بهره‌وری هر وظیفه τ_i بر اساس $U_i = \frac{C_i}{T_i}$ محاسبه می‌شود.

۲- تولید منابع

- هر یک از وظایف، در طول اجرای خود، یا در حال اجرای بخش عادی و یا در حال اجرای بخش بحرانی می‌باشد. در بخش بحرانی، وظیفه به یک منبع مشترک دسترسی انحصاری پیدا می‌کند و یا به بیانی دیگر بخش بحرانی، به بخشی از کد اشاره دارد که باید به صورت انحصاری به آن دسترسی پیدا کرد. به همین ترتیب هر یک از وظایف در طول اجرای خود می‌تواند به تعداد n_r منابع مشترک $\Theta = \{l_1, \dots, l_{n_r}\}$ دسترسی پیدا می‌کند.
- تعداد منابع موجود n_r در سامانه عددی تصادفی بین $[1,6]$ انتخاب می‌شود.

۳- تخصیص منابع به هر وظیفه

- هر $v \in V_i$ به صورت توالی از بخش عادی و بخش بحرانی نمایش داده می‌شود. به این صورت که هر گره v به صورت $v = \{c_{i,1}, c'_{i,1}, \dots, c'_{i,s-1}, c_{i,s}\}$ توصیف می‌شود که $c_{i,k}$ نشان‌دهنده k امین بخش عادی و $c'_{i,k}$ نشان‌دهنده k امین بخش بحرانی می‌باشد. به طور کلی هر بخش بحرانی بین دو بخش عادی از اجرای یک گره قرار می‌گیرد.
- تعداد کل دسترسی به هر منبع توسط تمام وظایف $\sum_{\tau_i \in T} N_{i,q}$ عددی تصادفی بین $[1,16]$ می‌باشد و بعد از اینکه این مقدار انتخاب شد، با توجه به حداکثر تعداد دسترسی به هر منبع، هر منبع به صورت تصادفی بین وظایف مختلف توزیع می‌شود. برای مثال با توجه به این موضوع یک وظیفه به تنهایی ممکن است ۳ مرتبه به منبع l_1 دسترسی پیدا کند.
- حداکثر طول دسترسی به هر منبع توسط تمام وظایف عددی تصادفی بین $[5,100]$ می‌باشد که هر $L_{i,q}$ خود به صورت تصادفی عددی بین $[1, \max_{\tau_i} \{L_{i,q}\}]$ می‌باشد. به بیانی دیگر اگر حداکثر طول دسترسی به منبع l_q برابر با ۵ باشد آنگاه طول دسترسی به منبع l_q در هر بخش بحرانی از هر وظیفه عددی تصادفی بین $[1,5]$ انتخاب خواهد شد و لزوماً طول دسترسی‌ها با هم برابر نیستند.
- دسترسی به منابع به صورت non-nested می‌باشد یعنی در صورتی که یک گره منبعی را در اختیار گرفت تا زمانی که آن منبع را رها نکند نمی‌تواند به منبع دیگری دسترسی پیدا کند.

۴- فرمول تعداد پردازنده کل

- تعداد کل پردازنده‌ها را می‌توان با توجه به فرمول $m = \left\lceil \frac{U_{\Sigma}}{U_{norm}} \right\rceil$ محاسبه کرد.
- U_{Σ} برابر با بهره‌وری کل وظایف موجود در یک مجموعه وظیفه می‌باشد.
- بهره‌وری به هنجار شده U_{norm} (نسبت بهره‌وری کل وظایف موجود در هر مجموعه وظیفه به تعداد کل پردازنده‌ها) هر مجموعه وظیفه از قبل مشخص می‌باشد و مقداری بین $[0.1,1]$ می‌باشد.

۵- Federated scheduling

- Federated scheduling روشی از زمان‌بندی است که بین global scheduling و partitioned scheduling قرار دارد.

- این روش بین وظایف با $U_i > 1$ و $U_i \leq 1$ تمایز قائل می‌شود ($U_i = \frac{C_i}{T_i}$) به این صورت که:
 - با توجه به فرمول $m_i = \left\lceil \frac{C_i - L_i}{D_i - L_i} \right\rceil$ ، به هر وظیفه τ_i با $U_i > 1$ به صورت انحصاری m_i پردازنده تعلق می‌گیرد.
 - به هر وظیفه با $U_i \leq 1$ به صورت انحصاری نیز یک پردازنده اختصاص داده می‌شود.
- مجموع این پردازنده‌ها نباید از تعداد کل پردازنده‌های موجود در سیستم بیش‌تر شود در غیر این صورت آن مجموعه وظیفه زمان‌بند پذیر نخواهد بود.

۶- خروجی

برای فاز اول، نتیجه هر کاری که انجام دادید باید نمایش داده شود.