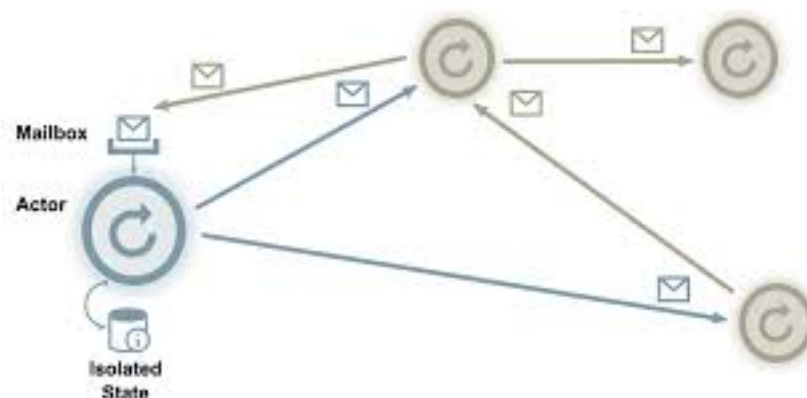


## مقدمه :

زمانیکه هدفمان تولید سامانه‌ی نرم افزاری باشد که تعداد بسیار زیادی از کاربران با آن سرو کار دارند و اتفاقاً این سامانه قرار است عملیات بسیار حساسی (نظیر عملیات بانکی و مالی، مخابراتی و ...) را انجام دهد و عدم سرویس دهی مناسب آن قابل تحمل نبوده و باعث خسارات مالی، نارضایتی و ... گردد می‌بایست از روش‌های خاصی برای توسعه‌ی این گونه سیستم‌ها استفاده نمود. این نرم افزارها برای اینکه بتوانند به تعداد درخواست‌های بسیار زیاد همزمان پاسخگو باشند و سرویس خود را با کیفیت مناسب ارائه دهند، می‌بایست دارای ویژگی‌های خاصی نظیر مقیاس پذیری (scalable) و تحمل پذیری در مقابل خطا (fault tolerance) باشند.

خوب حالا که صورت مسئله مشخص شد، یکی از راه حل‌های موجود را که مدل Actor Based است، بررسی می‌کنیم. مدل Actor Based یکی از مدل‌های استاندارد برای توسعه‌ی نرم افزارهایی با قابلیت اطمینان بسیار بالا، تحمل پذیر در مقابل خطا و پاسخ دهی بسیار سریع می‌باشد. در این مدل، وظایف نرم افزار به مجموعه‌ای از Actor ها تقسیم (توزیع) گردیده و هر یک از Actor ها به صورتی کاملاً ایزوله، در نخ (thread) خاص خودشان اجرا شده و بخشی از وظایف سیستم را انجام می‌دهند. سپس با اتصال Actor ها به یکدیگر، یک خط لوله (Pipeline) تشکیل شده و با استفاده از مکانیزم‌های ارسال و دریافت پیام، امکان همکاری و برقراری ارتباط بین Actor ها فراهم شده و در نتیجه وظیفه‌ی اصلی و کلی نرم افزار با حرکت در یک خط لوله و عبور از Actor های مختلف به صورت موازی و همزمان انجام خواهد شد. با توجه به اینکه هر یک از پیام‌های وارده به یک Actor در یک thread جداگانه اجرا می‌شود، امکان اینکه در یک لحظه چندین Thread در یک Actor در حال اجرا باشند و جود دارد و در نتیجه باید مکانیزم‌هایی وجود داشته باشد که تضمین کند پیام‌های وارد شده به خط لوله، به ترتیب معین شده، اجرا و از خط لوله خارج می‌شوند. در این مدل هر یک از Actor ها می‌توانند به صورت توزیع شده و بر روی سروری مجزا اجرا شوند. خوشبختانه فریمورک‌های متفاوت و بسیار قوی جهت توسعه به روش Actor Base وجود دارند؛ به عنوان مثال TPL DataFlow در .Net. یکی از نمونه‌های ساده آن بوده که در سال 2012 توسط Microsoft معرفی شد و Akka هم یک نمونه‌ی بسیار پخته‌تر و در بستر جاوا مطرح می‌باشد که پیاده سازی دات نت آن هم با نام Akka.net موجود است. Erlang نیز محصول Ericsson بوده و دنیای خاص خود را دارد.

در این روش وظیفه توسعه دهنده این است که اولاً یک خط لوله از اکتورها را تشکیل داده (کانفیگ) و یک عمل بزرگ را به چندین عمل کوچک‌تر تقسیم نموده و هر کدام را به یک اکتور جهت اجرا ارسال نماید. تصویر نمونه زیر یک خط لوله متشکل از 4 اکتور را نشان می‌دهد که از طریق ارسال پیام با یکدیگر در ارتباط هستند تا با همکاری یکدیگر عملی را انجام دهند. این ساختار، Pipeline یا خط لوله نامیده می‌شود.



در قسمت بعدی با جزئیات بیشتر و با نمونه‌های عملی این روش را بررسی می‌کنیم.