國立屏東大學 即時與嵌入式系統實驗室

Mixed-Criticality Systems

混合關鍵性系統

混合關鍵性系統(Mixed Criticality Systems，MCS)[1]研究領域源自於發展已逾半世紀的即時系統，其所考慮的工作不但具有嚴格時效性要求，還必須滿足工作在不同關鍵層級(Criticality Levels)下的各項要求，目前已進一步拓展至鐵路運輸、航空及車用電子(含自駕車與非自駕車)與醫療照護等安全攸關(Safety-Critical)應用領域，為具備不同關鍵層級(CriticalLevels)的即時系統工作提供具有可預測性與效能保證的學理分析、技術與工具。MCS的相關研究始於Steve Vestal在2007年IEEE RTSS會議[2]上所發表的首篇針對MCS即時工作排程方法論文[Vestal2007]。其後旋即吸引了眾多即時系統領域的學者專家投入相關研究，並在接下來的十餘年間逐步地成長為資訊領域備受重視的研究領域之一。關於MCS的相關研究成果可參考英國約克大學Burns教授與Davis教授合著的Survey論文[Burns2022]。

若從系統架構的角度來看，MCS通常被設計為是由多個軟體元件所構成的一個嵌入式系統(Embedded System)，其中每個軟體元件都會依據不同關鍵層級的功能性或非功能性需求，提供不同的實作版本；當然，也可以是單一的實作版本，但可依關鍵層級執行不同的功能。一般而言，當關鍵層級愈高時，軟體元件執行的結果對於人身安全的相關性就愈高，其所執行的工作內容自然就不同於低關鍵層級時的要求。例如一個用於計算自駕車路徑角度修正的工作，其要求的精確度在低關鍵層級時，只需要計算到小數點後5位；但在高關鍵層級時，為了得到更安全的結果，必須要計算到小數點後10位，並且還需要進行重複的驗算以確保其正確性。這些軟體元件的關鍵層級，是由MCS的設計者先行決定，在執行階段時必須在相同的硬體環境中運行，且還必須滿足不同關鍵層級的時效性要求。舉例來說，由負責執行煞車控制、倒車感應、胎壓偵測以及影音娛樂等不同工作的軟體元件所組成的車用電腦就是MCS。這些元件所負責的工作在執行時的安全性與時效性要求不盡相同，但必須在相同的嵌入式機板(Embedded Board)上執行[3]。

這些軟體元件的關鍵層級，是由MCS的設計者先行決定，在執行階段時必須在相同的硬體環境中運行，且還必須滿足不同關鍵層級的時效性要求。舉例來說，由負責執行煞車控制、倒車感應、胎壓偵測以及影音娛樂等不同工作的軟體元件所組成的車用電腦就是MCS。這些元件所負責的工作在執行時的安全性與時效性要求不盡相同，但必須在相同的嵌入式機板(Embedded Board)上執行[4]。

早期MCS的研究主要針對航空領域的整合模組化航電(Integrated Modular Avionics，IMA)系統[DiVito1999]，探討如何部署不同的應用工作於不同的航電模組裝置中，以避免其相互間的影響。從較高的抽象層次來看，此問題也就是工作切割(Task Partition)的問題 – 將多個工作切割部署到一個以上的同質性或異質性運算平台。自2007年Steve Vestal博士於IEEE RTSS會議發表了混合關鍵性工作的即時排程方法[Vestal2007]後，相關研究旋即開始聚焦於混合關鍵性工作在即時系統上的效能表現，眾多學者們陸續針對具備單處理器、多處理器或多核心處理器的即時系統，發表了為數眾多的混合關鍵性工作即時排程方法、同步方法以及相關的可排程性分析等研究成果，相關研究成果可參考英國約克大學Burns教授與Davis教授合著的Survey論文[Burns2017]。

特徵

混合關鍵性系統(MCS)的最大特徵就是其工作具有「時效性(Timeliness)」與「關鍵性(Criticality)」，其中時效性承襲自即時系統研究領域 — 工作的執行結果不但必須正確，同時還必須滿足嚴格 的時間要求;至於關鍵性則代表工作對於安全性的要求 — 高安全性要求的工作通常意味著需要得到 更為精確的運算結果，或是需要額外的處理器時間進行更為保守的驗證程序。後續我們將就時效性與關鍵性，以及其相關工作模型加以介紹:

時效性(Timeliness)

時效性是指工作的執行必須滿足特定的時間限制(Timing Constraint)，否則將有可能帶來不可彌補的後果，例如一個在車用MCS裡使用雷達偵測前車距離，並控制車速以保持安全間距的車距調節工作，它被設定為每秒必須執行5次，且每次都必須在200ms 內完成其執行，否則將可能造成車禍事故的發生。時效性是指工作的執行必須滿足特定的時間限制(Timing Constraint)，否則將有可能帶來不可彌補的後果，例如一個在車用MCS裡使用雷達偵測前車距離，並控制車速以保持安全間距的車距調節工作，它被設定為每秒必須執行5次，且每次都必須在200ms內完成其執行，否則將可能造成車禍事故的發生。與時效性最為相關且最常被使用的工作模型，有週期性即時工作模型[Liu1973]與偶發性工作模型(Sporadic Task Model)[Mok1983]

關鍵性(Criticality)

MCS在執行階段，可以視特定條件切換具有不同安全性要求的多個「關鍵性層級(Criticality Levels)」— 當關關鍵層級愈高時，工作的執行結果與人身安全的相關性就愈高。為了因應不同關鍵層級的安全性要求，工作通常必須設計並實作多個不同的版本，例如用以維持車輛行駛安全的循跡防滑控制工作，可以針對低安全性與高安全性要求分別實作兩個不同的版本供低關鍵層級與高關鍵層級運行時使用。當車輛處於低速與高速行駛時，車用MCS將會分別切換至對應的低與高關鍵層級並切換使用不同的工作實作版本。當MCS運行在低速行駛的低關鍵層級時，低安全性的工作實作版本在計算車輛行進軌跡時，只需要精確至小數點後2位；但是當車輛以高速行駛時，我們對於安全性的要求將會大幅提升，MCS將會切換至高關鍵層級運行，並且改為使用高安全性的工作版本計算行進軌跡至小數點後5位，同時還需要進行冗餘計算以確保計算結果的正確性。

上述概念首見於Steve Vestal於2007年 IEEE RTSS會議(以即時系統的週期性工作模型為基礎)所發表的首篇針對MCS即時工作排程方法論文[Vestal2007]，其假設工作的執行時間與其關鍵性成正比 — 意即工作的關鍵層級愈高、其執行時間愈長，我們將其稱為「關鍵性相依的執行時間工作模型(Criticality-Dependent-Execution-Time Task Model)」(由於Vestal率先提出此概念，所以此模型又常被稱為Vestal工作模型)。換句話說，Vestal認為工作的執行時間和其關鍵性是相依的 — 當工作的關鍵性愈高時，考慮到高關鍵性所代表的是「高安全性」需求，因此必須增加執行時間來計算更為精確的結果(或是反覆進行多次運算以確保結果的正確性)。為了因應不同關鍵層級的安全性要求，工作通常必須設計多個不同的實作版本，例如用以維持車輛行駛安全的循跡防滑控制工作，可以針對低安全性與高安全性要求分別實作兩個不同的版本供低關鍵層級與高關鍵層級運行時使用。當車輛處於低速與高速行駛時，車用MCS將會分別切換至對應的低與高關鍵層級並切換使用不同的工作實作版本。當MCS運行在低速行駛的低關鍵層級時，低安全性的工作實作版本在計算車輛行進軌跡時，只需要精確至小數點後2位；但是當車輛以高速行駛時，我們對於安全性的要求將會大幅提升，MCS將會切換至高關鍵層級運行，並且改為使用高安全性的工作版本計算行進軌跡至小數點後5位，同時還需要進行冗餘計算以確保計算結果的正確性。

在Vestal提出此假設及工作模型後，許多學者與研究團隊陸續投入相關的研究工作，並提出一些相關的衍生模型[Burns2022]，例如假設工作的週期與其關鍵性成反比，也就是當工作的關鍵層級愈高、其週期時間愈短 。例如當車輛從低速改變為高速行駛時，用以保持與前方車輛安全間距的車距調節工作，其工作的執行頻率將會從原本的每秒5次提升為20次(意即將工作週期從200*ms* 縮短為50*ms*)，以反映不同關鍵層級時對於車輛行駛的安全性要求。又例如自駕車路徑角度修正程式，從原本在低關鍵層級時的每10秒執行一次，

在高關鍵層級時則遞增到每1秒執行一次。在此情況下，這些工作的高關鍵層級的版本將與低關鍵層級的版本完全相同，但其執行頻率較高。我們將此假設稱為「關鍵性相依的週期工作模型(Criticality-Dependent-Period Task Model)」。儘管此衍生的假設與模型與Vestal原始的版本已不盡相同，但考量Vestal在此研究領域的創新貢獻，我們仍將其稱為Vestal工作模型。

分類

我們通常將只具有兩個關鍵層級的系統稱為Dual-Criticality MCS，並將具有兩個以上關鍵層級的系統稱為L-Criticality MCS。

關鍵層級標準

關於MCS系統的關鍵層級，目前已有許多安全攸關領域制定了相關的標準，例如:

- 美國聯邦航空總署(United States Federal Aviation Administration，FAA)就採用了航空無線電技術委員會(Radio Technical Commission for Aeronautics，RTCA)所制定的DO-178B標準[5]，要求在航電系統中所執行的工作，必須指定為Catastrophic，Hazardous，Major，Minor，No Effect等五個關鍵性層級之一。
- 廣泛應用於汽車產業的ISO 26262國際標準[6]，則定義了4個安全完整性等級(Safety Integrity Levels，SILs)。
- 核工領域的IEC 880標準(同時也是IEEE 603標準)[7)8)]
- 醫療領域的IEC 601-4標準[9)]
- 歐洲鐵路(European Railway)的EN 50128標準[10)]
- 歐洲航太標準化合作委員會(European Cooperation for Space Standardization)所制定的ECSS-E-ST40標準[11)]
- 歐洲標準化委員會(Comité Européen de Normalisation，CEN)發佈的EN 16602-80:2018標準[12)]。

MCS概念範例

以一個車用電腦為例，其工作的執行可區分為Safety-Critical層級，Mission-Critical層級與Non-Critical層級。系統中負擇煞車控制與倒車感應元件可歸屬於Safety-Critical層級，其工作的執行不但要求正確性，同時還必須滿足嚴格的時效性要求，否則將會帶來不可彌補的後果(例如逾時完成的煞車動作，可能帶來的是車禍意外的發生)；從即時系統的角度來看，此種工作屬於硬式即時工作(Hard Real-Time Task)。至於胎壓偵測元件則可歸屬於Mission-Critical層級，執行上必須確保正確性，但在時效性上則並不特別重視(稍為延遲回報的胎壓偵測數據並不會造成傷害性的後果，但若能如期回報當然更為理想)；也就是屬於軟式即時工作(Soft Real-Time Task)要求[13)]。在優先確保Safety-Critical與Mission-Critical的工作需求能夠得到滿足的前題之下，若系統還有多餘的能力讓影音娛樂元件也能提供高品質的影音播放效果固然是一件好事，但若做不到也不會為駕駛或乘客造成任何生理上的傷害，我們可將其歸屬於Non-Critical層級。

當關鍵層級愈高時，軟體元件執行的結果對於人身安全的相關性就愈高，其所執行的工作內容自然就不同於低關鍵層級時的要求。例如一個用於計算自駕車路徑角度修正的工作，其要求的精確度在低關鍵層級時，只需要計算到小數點後5位；但在高關鍵層級時，為了得到更安全的結果，必須要計算到小數點後10位，並且還需要進行重複的驗算以確保其正確性。

單處理器MCS工作排程方法

由於MCS衍生自即時系統，其工作排程方法的研發亦受到即時工作排程方法的影響深遠，雖然既有的即時工作排程方法並不能直接套用到MCS[Vestal2007][14]，但在探討MCS工作排程方法前，仍必須先對傳統即時系統的工作排程方法有基礎的認知(請參閱即時系統主頁)。本節後續將分別針對採用固定優先權(Fixed Priority)與動態優先權(Dynamic Priority)的MCS工作排程方法彙整其相關研究成果。

此外，針對MCS工作的排程方法大部份都是針對偶發性工作進行探討，但由於在最差情況下偶發性工作將會等同於週期性工作[15]，因此本節後續內容除另行說明外，皆針對週期性工作加以討論，但同時也適用於偶發性工作。

固定優先權排程方法

Vestal的論文[Vestal2007]除了提出創新的工作模型與可排程檢測方法外，還提出了以下兩個固定優先權的排程方法：

- Period Transform(優先權轉換)[Vestal2007]
- Optimal Priority Assignment(最佳優先全給定)[Vestal2007]

Vestal的成果為MCS研究社群帶來了深遠的影響，吸引許多學者針對不同的系統架構投入相關的排程方法研究，例如在單處理器[Baruah2010b], [LiBaruah2010a], [Baruah2012], [Baruah2012b], [ParkKim2011], [Baruah2010], [LiBaruah2010b], [Socci2013a], [Gu2013], [Socci2015b], [Baruah2016]以及多處理器(或多核心處理器)[Kelly2011], [Rodriguez2013], [Gu2014], [Han2016]等，並有部份學者針對Vestal所提出模型進行修改，考慮更為真實的工作模型並提出相關的排程方法(例如Dorin進一步考慮了工作到達時間的延遲變化率(Release Jitter)[Dorin2010])□我們在此彙整固定優先權的排程方法如下：

- Own-Criticality-Based-Priority (OCBP)
- Static Mixed-Criticality (SMC) Scheduling (靜態混合關鍵性排程法)[BaruahVestal2008]
- Adaptive Mixed-Criticality (AMC) Scheduling (可調適混合關鍵性排程法)[Baruah2011b]

動態優先權排程方法

在動態優先權排程方法方面，幾乎所有的MCS排程研究成果都是採用著名的最早截限時間優先(Earliest Deadline First□EDF)[Liu1973][16]方法進行工作排程，

- Earliest Deadline First with Virtual Deadlines(EDF-VD)

其它相關排程方法

除了上述所介紹的方法以外，亦有一些學者使用不同的工作排程策略於混合關鍵性即時工作排程，例如UNC的研究團隊所提出的Time-Triggered排程方法[BaruahFohler2011], [Socci2013b]□就曾被他們自己以及其他研究團隊使用在混合關鍵性即時工作排程方法設計上[TheisFohler2013], [Baruah2012b], [Baruah2013a], [Yip2014], [Cohen2014]□另外□UNC研究團隊的Baruah與Guo也曾經設想一種處理器執行速度可能會降低的特殊情況 — 考慮處理器具有Normal與Degraded兩種運作速度，其中Normal為正常速度□Degraded則為當電力供應不足導致效能衰退時的較低速度。針對此種情況，他們考慮了例如當處理器效能開始衰退(也就是使用Degraded的低速度時)，限制僅允許高關鍵層級工作執行等策略，並發表了一連串的相關論文[BaruahGuo2013], [GuoBaruah2014], [BaruahGuo2014], [Guo2016], [GuoBaruah2015]□雖然與現在主流的動態電壓調整(Dynamic Voltage Scaling□DVS)[17]處理器技術無關，但其相關做法可做為

未來配合具DVS能力的處理器，設計具有節能效益的混合關鍵性即時工作排程方法參考。

多核心MCS工作排程方法

多核心處理器(Multi-Core Processor)可以允許多個工作同時在不同的處理器核心上執行，對於效能的提升相當有助[18)]。我們將一個擁有N^c個核心的多核心處理器定義為$\mathcal{MC}$其核心則以$core_1, core_2,$ $\cdots, core_{N^c}$表示。傳統即時系統的多核心處理器工作排程方法可概分為：Partitioned Scheduling、Global Scheduling與Clustered Scheduling等四類，相關的研究成果可參考英國約克大學的Davis教授與Burns教授所彙整的Survey論文(A Survey of Hard Real-Time Scheduling for Multiprocessor Systems)[DavisBurns2011]在此限於篇幅不予贅述。

目前針對多核心處理器方面的混合關鍵性即時工作排程研究仍處於起步的階段，仍有待學者們投入相關研究。在現有的研究成果當中，大部份是針對採用Partitioned Scheduling方法時，探討該如何進行適當的工作切割的方法[Kelly2011], [Rodriguez2013], [Gu2014], [Han2016]例如Han等學者[Han2016]將EDF-VD方法[Baruah2015]延伸應用到多核心系統，並探討其相關的工作切割方法Kelly等學者[Kelly2011]則針對First-Fit與Best-Fit等常見的切割策略，再加上處理器使用率(Utilization)以及工作的關鍵層級，以Vestal回應時間分析式及其可排程性檢測方法[Vestal2007]進行了相關的比較，其結果顯示依工作關鍵層級遞減順序的First-Fit方法(First-Fit with Decreasing Criticality Order)有較佳的效能表現。另外Rodrigueze[Rodriguez2013]和Gu[Gu2014]則針對以EDF為基礎的混合關鍵性即時工作排程方法，進行了工作切割方法的比較，其研究結果一致指出讓高關鍵層級與低關鍵層級的工作，分別以Worst-Fit與First-Fit配置會有較好的系統效能。

除了Partitioned Scheduling的多核心混合關鍵性即時工作排程方法以外，其它的多核心排程方法學界著墨甚少，僅有部份研究是針對Global Scheduling或是Semi-Partitioned加以探討，例如Gratia等學者[Gratia2014], [Gratia2015]針對Dual-Criticality以及L-Criticality系統使用RUN排程法[Regnier2011]進行工作排程Li與Baruah則在允許工作遷移的前題之下，提出了EDF-VD方法[Baruah2015]的Global Scheduling版本[LiBaruah2012]除此之外Al與Bayati[Al-Bayati2015]以及Xu等[XuBurns2015]學者則針對Semi-Partitioned方法進行了初步的探討。

MCS工作同步方法

至於在混合關鍵性即時工作方面，共享資源的同步問題比起傳統即時系統更為困難，因為共享資源不但會被不同優先權的工作存取，還會被來自不同關鍵層級的工作存取。為了確保混合關鍵性即時工作的時間限制能夠得到滿足，不但需要考慮高優先權工作被低優先權工作阻擋的問題，還需要考慮高關鍵層級工作也可能因為共享資源的存取而被低關鍵層級工作阻擋的問題。目前在此方面的研究工作僅處於起步階段，大部份都是針對單處理器環境加以探討，後續仍亟待學者們積極投入。在目前少數的研究成果中，在固定優先權工作方面，有英國約克大學Burns的研究團隊結合PCP[Sha1990]與AMC[Baruah2011b]方法，提出了將共享資源放置於不同關鍵層級內，讓工作僅能被在同一個關鍵層級中的低優先權工作所阻擋的同步方法[Burns2013]而在動態優先權的工作方面，則有Zhao等學者結合EDF[Liu1973]與SRP[Baker1990]的方法被提出[Zhao2013a], [Zhao2015]此外Lakshmanan與Zhao等學者，仿效PIP方法[Sha1990]中的Priority Inheritance概念，提出了稱為Criticality Inheritance的概念，並將其結合到PIP與PCP方法[Lakshmanan2011], [Zhao2014]不同於前述的作法Brandenburg等學者提出了一個將所有共享資源交由Resource Server管理的方法[Brandenburg2014]

國際著名研究團隊

綜觀上述研究成果，國際上關於MCS工作排程研究的主要貢獻大部份來自於英國約克大學(University of York)以Alan Burns教授以及美國北卡羅來納大學教堂山分校(University of North Carolina at Chapel Hill，UNC)以Sanjoy Baruah教授為首的研究團隊所提出。

論文關係圖

graph TD V[["Vestal2007"]] A(**mermaid**)-->B((plugin)) A-->C(((*for*))) B-->D[["Dokuwiki"]] C-->D

論文清單

bibtex file

A

- Adaptive Mixed-Criticality (AMC) Scheduling (可調適混合關鍵性排程法)

E

- Earliest Deadline First with Virtual Deadlines(EDF-VD)

O

- Own-Criticality-Based-Priority (OCBP)

S

- SCM - This a test for SCM
- Static Mixed-Criticality (SMC) Scheduling

V

- Vestal2007 - Preemptive Scheduling of Multi-criticality Systems with Varying Degrees of Execution Time Assurance

References

1)
早期有部份學者將此研究領域稱為Multi-Criticality(多重關鍵性)系統，後來已統一稱為Mixed Criticality(混合關鍵性)系統。
2)
RTSS會議全名為IEEE Real-Time Systems Symposium，是即時系統領域最頂尖的兩個國際會議之一；另一

個為IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)
[3] [4]
,
此處所討論的是較為單純的情況，在許多真實的應用中，一個車用電腦系統可以包含多個嵌入式機板，或是在機板上有多個可獨立運行的處理器核心。在這些情況下，除了要確保滿足不同關鍵層級的軟體元件需求外，還必須考慮元件在不同機板或處理器核心上的部署問題。
[5]

RTCA, Software Considerations in Airborne Systems and Equipment Certification, 1992.
[6]

ISO 26262-1:2018(en) Road vehicles — Functional safety.
https://www.iso.org/obp/ui/#iso:std:iso:26262:-1:ed-2:v1:en. Accessed: 2020-12-21.
[7]

IEC 880 Software for Computers in the Safety Systems of Nuclear Power Stations. 1986.
[8]

IEEE 603-1991 Standard Criteria for Safety Systems for Nuclear Power Generating Stations. 1991.
[9]

IEC 601-4, Safety Requirements for Programmable Electronic Medical Systems. 2011.
[10]

European Committee for Electrotechnical Standardization (CENELEC). Railway Applications - Communication, Signaling and Processing Systems - Software for Railway Control and Protection Systems. 2011.
[11]

European Cooperation for Space Standardization (ECSS), ECSS-E-ST-40C, Software.
https://ecss.nl/standard/ecss-e-st-40c-software-general-requirements/. Accessed: 2020-12-21.
[12]

European Committee for Standardization (CEN). Space Product Assurance - Software Product Assurance. https://www.cen.eu, Accessed: 2020-12-21.
[13]

在系統設計上，我們可以將煞車控制與倒車感應歸屬於同一關鍵層級(也就是safety-critical)或是進一步將它們分別分類為High Safety-Critical與Normal safety-critical兩類，畢竟煞車的故障與錯誤或延遲的胎壓偵測，所帶來的後果將會是完全不同等級的。
[14]

由於傳統的即時系統可視為系統內僅存在單一個關鍵層級的MCS所以對於擁有多個關鍵層級的MCS而言，無法直接使用傳統的即時工作排程方法。關於此論述更詳細的說明，亦可參考Steve Vestal論文[Vestal2007]裡的證明。
[15]

在最差情況下，每一個偶發性工作任務的任意兩個連續的工作，其到達時間的間隔都等於其最短間隔時間。因此若將最短間隔時間視為是工作的週期，則偶發性工作就等同於週期性即時工作。
[16]

Earliest Deadline First排程方法是在執行階段，依據工作距離其截限時間的遠近決定工作的優先權，較為急迫的工作將取得較高的優先層級；換句話說EDF方法永遠挑選距離截限時間最近的工作進行排程。
[17]

DVS處理器可透過供給電壓的改變來切換工作的執行速度，對於降低系統能耗或延長電池使用時間極有助益。
[18]

從抽象觀點來看，多核心處理器架構與早期所探討的多處理器架構，有極高的相似性；若將多核心處理器的每個核心視為是一個處理器，那麼適用於多核心處理器的排程方法也能夠應用在多處理器架構之上，因此本節將此兩類處理器架構相關的方法一併加以彙整。