

參、研究設計

3.1 研究架構及假設

利用 MVC Pattern(Model – View – Controller 設計樣式)說明模擬多層式架構時所需的準備工作，如圖 11。在 Client 端需要定義使用者的工作項目、每一項工作的大小(job size)及工作發生的時間(arrival time)；而 Controller 則扮演控制每項工作所需執行的順序及其需要的元件，它也可以部署多台網頁伺服器而形成叢集(cluster)；在 Model 則表達儲存所有元件的儲存器，並且將其表達為一個應用伺服器叢集；後端則為負責資料存取的資料庫。其中二個叢集的部分，就需要派工法則以達到平衡負載的功能。

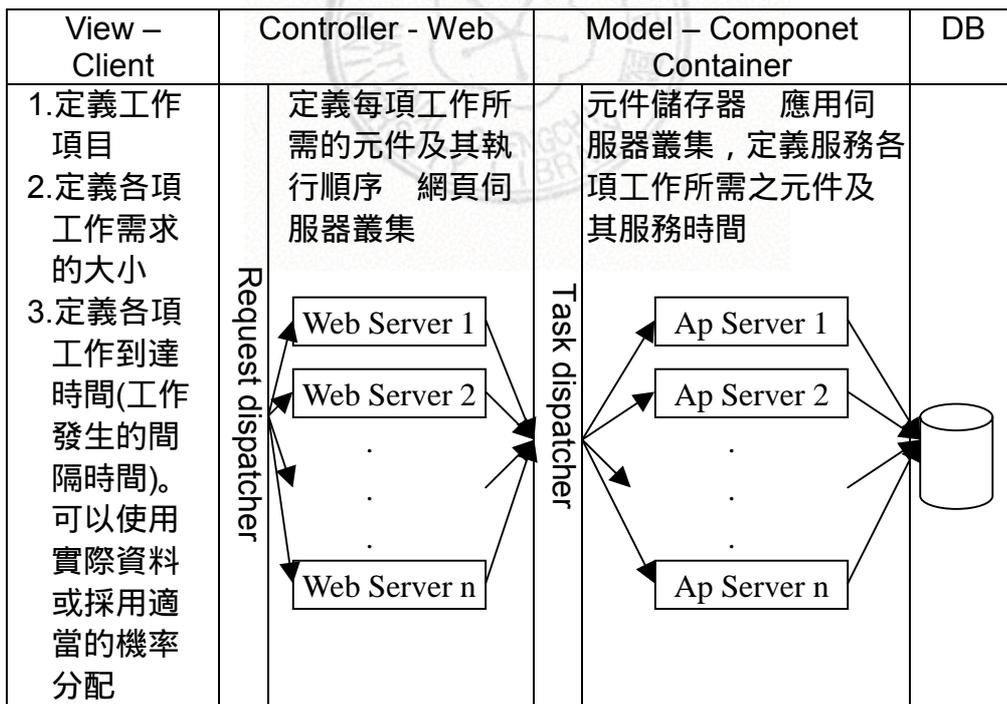


圖 11：以 MVC 表達之多層式架構模擬概念圖

而本研究著重之主題在應用伺服器叢集的 task dispatcher 所使用之派工法則對各項工作平均等待時間的影響，故先排除未涉及研究主題的部份。簡化後之研究架構如圖 12。

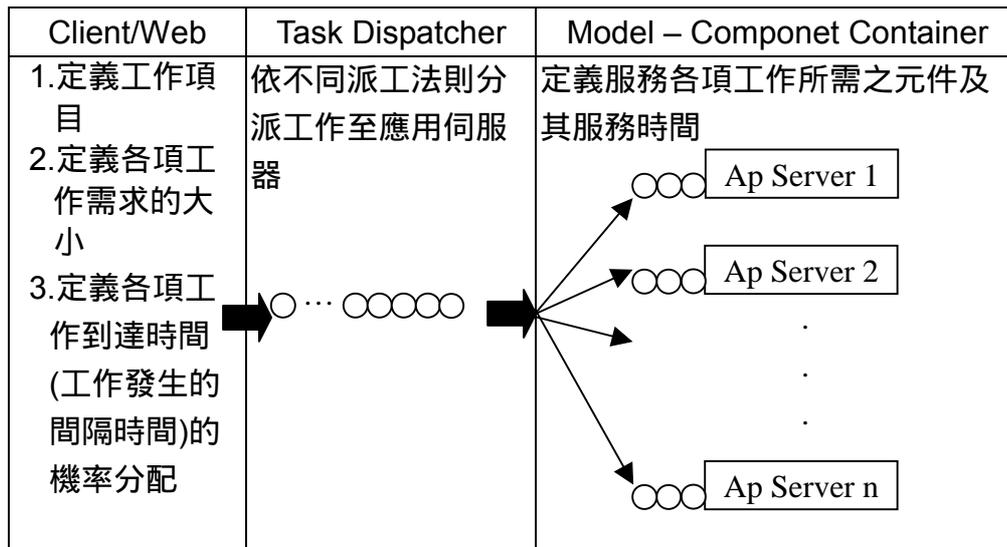


圖 12：研究架構概念圖

依據上述的研究架構，本研究做以下的假設：

1. Client 端的假設：

(1)12 種工作項目：現行的中央政府預算編製作業的工作項目視工作切割的方式，可達數百種、甚至上千種。本研究只挑選預算書表中較具代表性之二張書表 歲出計畫提要及分支計畫概況表及歲出機關別預算表，並將此二張表區分成十二項工作。如表 1。

(2)包含 832 家機關單位預算及分預算機關的作業量。

(3)假設每一項工作發生的間隔時間呈現指數分配，並在相同的平均數之下，考慮變異數的大小，故取 CV(Coefficient of Variation)值為 1/2、1 及 2，以獲得間隔時間的變異在大、中、小三種情況之下的結果。

(4)假設不同機關的各項工作需求大小(job size)呈現常態分配。

2. Task Dispatcher 的假設

(1)不限制工作到達的數量及時間。

(2)採用先進先出法(FIFO, First In First Out)，先到之工作先分派，一次分派一項工作。

(3)工作到達時，馬上透過派工法則進行分派，並不考慮分派工作所需時間。

(4)測試以下四種派工法則：

- 輪流法(Round Robin, RR)：當工作到達時，Task Dispatcher 將工作輪流分派給不同的應用伺服器服務。
- 最少連線法 (Least Connections, LC)：當工作到達時，Task Dispatcher 會視各應用伺服器的連線數，將工作分派給最少連線數的應用伺服器服務。
- 估計負載法(Estimated Load, EL)：本研究提出的派工法則。由事前分析各項工作的需求大小及其所需之服務時間來估算每一台應用伺服器的負載量。每當一項工作到達時，Task Dispatcher 會依之前工作分派的記錄，估算負載最輕的伺服器，便將該項工作分派至該應用伺服器。
- 最小負載法(Least Load, LL)：當 client 端的工作到達時，視當時所有應用伺服器的負載狀況，挑選負載最輕的應用伺服器服務此工作。前提是可以正確掌握每台應用伺服器的負載情況，但在實務上並不可行。所以，此項派工法則在本研究中是擔任比較其他各種派工法則的基準。

3. 應用伺服器的假設

- (1)每一個工作種類，由不同的元件(component)對應處理，如表 1。每當一個工作到達時，即建立一個新的元件實例(instance)處理之，處理結束後立即將該元件實例毀滅。暫不考慮 Instance Pooling 等技術。
- (2)採用先進先出法(FIFO, First In First Out)，一次服務一項工作，每一個工作進來，即佔用一個連線；等待佇列無限制大小。暫不考慮 Connection Pooling 等技術。
- (3)處理每一個工作的時間(service time)可以用簡單迴歸方程式表示[4]。
- (4)只考慮分派工作進入(incoming request)處理，不考慮回應工作的處理。

表 1：研究架構概念表

Client		Dispatcher	Application Servers		
工作種類	工作描述	派工法則	應用伺服器	元件種類	元件內容
job1	查詢歲出計畫提要及分支計畫概況表及說明	RR、LC、EL 及 LL 四項派工法則	Server1..4 共 4 台	compoent 1	處理 job1
job2	新增歲出計畫提要及分支計畫概況表預算資料及說明			compoent 2	處理 job2
job3	修改歲出計畫提要及分支計畫概況表預算資料及說明			compoent 3	處理 job3
job4	刪除歲出計畫提要及分支計畫概況表預算資料及說明			compoent 4	處理 job4
job5	查詢歲出歷史資料			compoent 5	處理 job5
job6	新增歲出歷史資料			compoent 6	處理 job6
job7	修改歲出歷史資料			compoent 7	處理 job7
job8	刪除歲出歷史資料			compoent 8	處理 job8
job9	查詢歲出機關別預算表			compoent 9	處理 job9
job10	新增歲出機關別預算表預算說明			compoent 10	處理 job10
job11	修改歲出機關別預算表預算說明			compoent 11	處理 job11
job12	刪除歲出機關別預算表預算說明			compoent 12	處理 job12

3.2 資料收集與分析

依照前面章節定義之研究架構及假設，蒐集模擬所需之各項數據及計算公式：

1. 使用者端所需數據：

透過問卷及訪談二十二家機關的預算編製承辦人員，獲得以下數據：

(1)每家機關處理一個預算階段時，使用資訊系統處理的時間大約是五天；而一天的作業時間大約是 10 小時。

(2)每家機關在一個預算階段必需處理此十二項工作的作業次數，如表 2。經

過計算每項工作在五天內(每天工作 10 小時)的平均作業次數及 832 家機關的作業量，並知其呈現 Poisson 分配；將單位時間內每項工作的發生次數轉換為每項工作的發生間隔時間，則為指數分配。計算出其工作發生間隔時間之平均數；並且考慮間隔時間在不同變異程度情況下的影響為何，故計算 CV(Coefficient of Variance)值為 1、1/2 及 2 時之三種 Gamma 分配來表達工作發生間隔時間之機率分配。原始資料如表 2、計算結果如表

3。計算過程如下：

$$cv = \sqrt{\text{var}} / \text{mean}$$

$$cv = 1, \quad 1 = \frac{\sqrt{\alpha} \beta}{\alpha \beta}, \quad 1 = \frac{1}{\sqrt{\alpha}}, \quad \alpha = 1$$

$$cv = 2, \quad \alpha = 1/4$$

$$cv = 1/2, \quad \alpha = 4$$

表 2：22 家機關執行各項作業次數統計表

	job1	job2	job3	job4	job5	job6	job7	job8	job9	job10	job11	job12
1	42	20	5	3	35	20	10	20	30	20	10	8
2	30	30	10	30	5	5	5	5	30	25	10	15
3	26	28	9	19	26	28	26	14	10	24	28	40
4	15	10	5	5	20	10	5	10	10	10	5	8
5	5	5	3	3	5	5	5	5	10	10	3	10
6	5	30	10	30	5	5	3	3	3	20	3	23
7	15	15	5	30	15	15	5	10	15	15	5	30
8	20	10	3	6	3	2	2	2	20	10	5	14
9	30	15	5	20	15	20	5	5	25	25	10	20
10	1	1	1	1	5	5	5	5	5	5	1	9
11	10	5	3	2	20	20	20	5	10	10	3	6
12	60	12	30	30	15	10	5	6	15	20	10	55
13	20	60	10	15	10	5	5	10	20	50	20	40
14	10	5	5	5	1	1	1	1	2	2	2	4
15	25	25	5	35	30	25	5	15	10	30	25	45
16	15	12	3	20	10	5	2	10	5	15	2	45
17	20	10	5	30	10	5	3	3	5	20	10	45
18	30	25	5	10	20	10	10	10	30	10	5	15
19	10	40	5	20	2	5	2	10	2	5	2	20
20	20	20	10	50	10	10	5	5	5	20	5	40
21	10	20	3	30	5	5	3	10	10	15	2	40
22	15	16	3	20	15	15	5	20	5	10	10	40

表 3：各項工作發生間隔時間機率分配表

各項工作發生間隔時間機率分配表			
	Gamma(α, β)		
	CV=1	CV=2	CV=1/2
Job1	(1,10.967)	(1/4,43.867)	(4,2.742)
Job2	(1,11.497)	(1/4,45.987)	(4,2.874)
Job3	(1,33.284)	(1/4,133.136)	(4,8.321)
Job4	(1,11.497)	(1/4,45.987)	(4,2.874)
Job5	(1,16.878)	(1/4,67.512)	(4,4.220)
Job6	(1,20.604)	(1/4,82.418)	(4,5.151)
Job7	(1,34.742)	(1/4,138.967)	(4,8.685)
Job8	(1,25.867)	(1/4,103.470)	(4,6.467)
Job9	(1,17.183)	(1/4,68.731)	(4,4.296)
Job10	(1,12.829)	(1/4,51.317)	(4,3.207)
Job11	(1,27.043)	(1/4,108.173)	(4,6.761)
Job12	(1,8.321)	(1/4,33.284)	(4,2.080)

(3)從 30 家機關現行資料庫中查詢影響本研究定義之 12 項工作大小的資料表。假設 832 家機關之工作大小呈現常態分配，整理後如表 4。

表 4：工作大小機率分配表

各機關需求大小(Request Size)資料表			
工作種類	平均數 (Kbytes)	標準差 (Kbytes)	常態分配
Job1 ~ Job4	21.324	39.714	N(21.324,39.741)
Job5 ~ Job8	1.066	2.126	N(1.066,2.126)
Job9 ~ Job12	7.475	12.164	N(7.475,12.164)

2. 應用伺服器端所需的數據

應用伺服器處理每一種工作所需的處理時間，參考 Teo[21]的作法，將其視為線性函數。根據實際測試現行系統對於每一項工作的處理時間，獲得以下數

據，如表 5、6、7。獲得此項數據的測試平台如下：

(1)硬體：

- CPU：Pentium 4 -1.4Ghz
- RAM：512MB RDRAM
- HD：7200RPM、ATA100

(2)軟體：

- 作業系統：Windows 2000 Professional
- 資料庫軟體：Microsoft SQL Server 7.0
- 應用系統：GBA V1.4(中央政府歲計會計系統)
- 數據監控軟體：Microsoft SQL Server 7.0 – Profiler

表 5：工作項目 1 至 4 之服務時間迴歸分析表

序號	Job Size (單位：Kbytes)	Service Time (單位：秒)			
		Job1	Job2	Job3	Job4
1	1.960	0.540	1.273	1.220	0.500
2	6.272	0.660	1.690	1.433	1.040
3	10.584	1.124	2.024	1.543	1.260
4	14.896	1.250	2.634	1.627	1.453
5	19.208	1.473	2.964	1.750	1.534
6	23.520	1.720	2.984	1.763	1.813
7	27.832	2.013	3.130	1.877	2.474
8	32.144	2.364	4.664	2.244	2.567
9	36.456	2.653	4.874	2.913	2.754
10	40.768	3.167	5.713	3.573	3.406
迴歸分析	截距	0.305	0.528	0.919	0.437
	斜率	0.065	0.195	0.050	0.068
	標準誤	0.116	0.595	0.334	0.169

表 6：工作項目 5 至 8 之服務時間迴歸分析表

序號	Job Size (單位：Kbytes)	Service Time (單位：秒)			
		Job5	Job6	Job7	Job8
1	0.576	0.110	0.010	0.010	0.010
2	0.864	0.140	0.010	0.010	0.010
3	1.152	0.150	0.010	0.010	0.010
4	1.440	0.170	0.010	0.010	0.010
5	1.728	0.170	0.010	0.010	0.010
6	2.016	0.180	0.010	0.010	0.010
7	2.304	0.180	0.010	0.010	0.010
8	2.592	0.200	0.010	0.010	0.010
9	2.880	0.230	0.010	0.010	0.010
10	3.168	0.230	0.010	0.010	0.010
迴歸分析	截距	0.097	0.010	0.010	0.010
	斜率	0.042	0.000	0.000	0.000
	標準誤	0.009	0.000	0.000	0.000

表 7：工作項目 9 至 12 之服務時間迴歸分析表

序號	Job Size (單位：Kbytes)	Service Time (單位：秒)			
		Job9	Job10	Job11	Job12
1	0.68	0.56	0.89	0.86	0.970
2	3.156	0.76	1.36	1.103	1.400
3	5.632	0.893	2.133	1.24	2.310
4	8.108	1.153	3.133	2.18	2.853
5	10.584	1.17	3.757	3.094	3.064
6	13.06	1.57	4.023	3.52	3.263
7	15.536	2.187	4.137	4.41	4.554
8	18.012	2.35	4.421	4.96	4.974
9	20.488	2.47	4.922	5.423	5.990
10	22.964	2.663	5.696	6.523	6.320
迴歸分析	截距	0.371	1.048	0.237	0.708
	斜率	0.102	0.203	0.262	0.242
	標準誤	0.157	0.349	0.273	0.282

3.3 建立模型

本研究採用三種不同的工作到達間隔時間的機率分配及四種不同的派工法則，故總共有十二個模型需要建立。如表 8。

表 8：模型建置表

組別	模型編號	間隔時間機率分配	派工法則
1	1	Gamma(cv=1)	RR
	2	Gamma(cv=1)	LC
	3	Gamma(cv=1)	EL
	4	Gamma(cv=1)	LL
2	5	Gamma(cv=2)	RR
	6	Gamma(cv=2)	LC
	7	Gamma(cv=2)	EL
	8	Gamma(cv=2)	LL
3	9	Gamma(cv=1/2)	RR
	10	Gamma(cv=1/2)	LC
	11	Gamma(cv=1/2)	EL
	12	Gamma(cv=1/2)	LL

以下便以 Promodel 模擬軟體來說明第一組中的四個模型的運作規則：

1. 根據研究架構，定義模型中各種角色的位置，包括發生 12 種工作需求的位置 1 台 Task Dispatcher 及 4 台應用伺服器 在 Promodel 中需定義 Location 資料，如表 9。

表 9：Location 定義表

Location 定義表					
Name	Capacity	Unit	Statistics	Rule	Cost
Job_Class1..12	1	1	Time Series	Oldest	
Dispatcher	1	1	Time Series	Oldest	
AP_Server1..4	Infinite	1	Time Series	Oldest	

2. 定義在 Job_Class1 至 12 位置產生各項工作間隔時間的機率分配，定義在 Promodel 中的 Arrival 項目中，由 Promodel 依照定義的機率分配去產生十二項工作，並送至 Task Dispatcher。Arrival 定義如表 10。

表 10：Arrival 定義表

Arrival 定義表				
Entity	Location	Quantity each	Occurrences	Frequency
Job1	Job Class1	1	Infinite	G(1,10.967)SEC
Job2	Job Class2	1	Infinite	G(1,11.497)SEC
Job3	Job Class3	1	Infinite	G(1,33.284)SEC
Job4	Job Class4	1	Infinite	G(1,11.497)SEC
Job5	Job Class5	1	Infinite	G(1,16.878)SEC
Job6	Job Class6	1	Infinite	G(1,20.604)SEC
Job7	Job Class7	1	Infinite	G(1,34.742)SEC
Job8	Job Class8	1	Infinite	G(1,25.867)SEC
Job9	Job Class9	1	Infinite	G(1,17.183)SEC
Job10	Job Class10	1	Infinite	G(1,12.829)SEC
Job11	Job Class11	1	Infinite	G(1,27.043)SEC
Job12	Job Class12	1	Infinite	G(1,8.321)SEC

3. 接下來定義每項工作本身具有三項屬性：

(1) 工作大小(Job Size)：Att_Ent_Size，此項屬性在產生一個 Job 時，由 Promodel 依據表 4 的常態機率分配產生給值。

(2) 該項工作預估需要的服務時間：Att_Ent_MRT_Est，以表 5、6、7 的數據並套用以下公式，便可以獲得此項工作的預估需要服務時間。因為是估計的服務時間，所以未計算標準誤，這項數據也是 EL 法計算應用伺服器負載的計算基礎。

$$\text{Att_Ent_MRT_Est} = \text{截距} + (\text{斜率} * \text{Att_Ent_Size})$$

(3) 該項工作實際需要的服務時間：Att_Ent_MRT，此項屬性與前項不同之

處在於實際需要的服務時間通常無法準確預估，故本研究以迴歸分析中的標準誤來表示此項誤差。所以，本項屬性值的產生公式如下：

$$\text{Att_Ent_MRT} = \text{截距} + (\text{斜率} * \text{Att_Ent_Size}) + \text{Normal}(0, \text{標準誤})$$

4. 每當使用者端依據 Arrival 的定義，產生一項新的工作時，便給定前述三項屬性值；接下來便將此項工作傳至 Task Dispatcher 做分派的工作。以下說明四種派工法則的處理邏輯：

(1) RR：在 Promodel 中的 Route Rule 設定中，可以直接設定為 By turn，即是 RR 法。

(2) LC：在 Task Dispatcher 的處理邏輯(Process)定義中，需要計算以下公式，將此工作分派至負載量最輕的應用伺服器處理。

$$\text{Server_Load}(N) = \text{Var_Server_Conn}(N)$$

註：Server_Load(N)是指第 N 台應用伺服器的負載情形

Var_Server_Conn(N)是指第 N 台應用伺服器的連線數量

(3) EL：在 Task Dispatcher 的處理邏輯(Process)定義中，需要計算以下公式，將此工作分派至負載量最輕的應用伺服器處理。

$$\text{Server_Load}(N) = \text{Var_Server_TMRT_Est}(N)$$

註：Server_Load(N)是指第 N 台應用伺服器的負載情形

Var_Server_TMRT_Est(N)是指第 N 台應用伺服器等待佇列中尚待處理工作估計所需之時間總合

(4) LL：在 Task Dispatcher 的處理邏輯(Process)定義中，需要計算以下公式，將此工作分派至負載量最輕的應用伺服器處理。

$$\text{Server_Load}(N) = \text{Var_Server_TMRT}(N)$$

註：Server_Load(N)是指第 N 台應用伺服器的負載情形

Var_Server_TMRT(N)是指第 N 台應用伺服器等待佇列中尚待處理工作所需之時間總合

5.當工作被分派至任一應用伺服器時，需要做以下處理：

(1)將此工作置入等待佇列，並計算佇列中所有工作的實際需要的服務時間
及預估所需的服務時間

$$\text{Var_Server_TMRT}(N) = \text{Var_Server_TMRT}(N) + \text{Att_Ent_MRT}$$

$$\text{Var_Server_TMRT_Est}(N) = \text{Var_Server_TMRT_Est}(N) + \text{Att_Ent_MRT_Est}$$

(2)處理等待佇列中最前面的工作，並佔用該應用伺服器所需服務的時間

USE Res_Server(N) for Att_Ent_MRT

(3)待工作處理完畢後，此項工作便離開應用伺服器，此時必需扣除該項工
作的服務時間

$$\text{Var_Server_TMRT}(N) = \text{Var_Server_TMRT}(N) - \text{Att_Ent_MRT}$$

$$\text{Var_Server_TMRT_Est}(N) = \text{Var_Server_TMRT_Est}(N) - \text{Att_Ent_MRT_Est}$$

以上重點說明如何利用 Promodel 建立第一組的四個模型，另外第二、三組
的模型，僅需依據表 4 修改 Arrival 的定義即可完成。