

具有吞吐量及时延约束的软件无线电波形部署决策算法*

吴尚峰¹, 唐麒¹, 魏急波^{1,2}, 李卫^{1,3}

(1. 国防科技大学 电子科学与工程学院, 湖南 长沙 410073;

2. 中国电子科技集团公司第五十四研究所 通信网信息传输与分发技术重点实验室, 河北 石家庄 050081;

3. 中国人民解放军 75753 部队, 广东 广州 510600)

摘要:针对当前具有吞吐量和时延约束的软件无线电波形部署决策算法在内存开销方面考虑的不足, 提出一个更为通用的波形部署决策系统模型, 将波形的处理开销、内存开销和通信带宽开销都纳入到波形部署决策过程中, 并设计了一种基于动态规划的波形部署决策算法以最小化波形对平台各种计算资源的占用总和。仿真结果表明, 内存对波形部署决策具有重要影响, 与未考虑内存开销的解决方案相比, 平台支持的平均最大波形数目和平均处理资源利用率均下降40%左右。

关键词:软件无线电; 波形部署决策; 吞吐量和时延约束; 流水线调度; 动态规划

中图分类号: TP924 文献标志码: A 文章编号: 1001-2486(2015)01-136-06

A deployment decision-making algorithm for software defined radio waveforms with throughput and latency constraints

WU Shangfeng¹, TANG Qi¹, WEI Jibo^{1,2}, LI Wei^{1,3}

(1. College of Electronic Science and Engineering, National University of Defense Technology, Changsha 410073, China;

2. Science and Technology on Information Transmission and Dissemination in Communication Networks Laboratory,

The 54th Research Institute of China Electronics Technology Group Corporation, Shijiazhuang 050081, China;

3. The PLA Unit 75753, Guangzhou 510600, China)

Abstract: Aimed at the problem that current deployment decision-making algorithms for software defined radio waveforms with throughput and latency constraints were paid little attention to the memory overhead, a more general system model for waveform deployment decision-making was proposed, by which the processing, memory and communication bandwidth were all incorporated in the process of waveform deployment decision-making. Besides, a dynamic programming-based algorithm for waveform deployment decision-making was devised to minimize the sum of various kinds of computing resource occupancies of waveform. Simulation results show that the memory has an important effect on the waveform deployment decision-making, while comparing with the algorithm that does not consider the memory overhead, the average maximum waveform number and the average processing resource usage supported by platform are both decreased about 40%.

Key words: software defined radio; waveform deployment decision-making; throughput and latency constraints; pipelined scheduling; dynamic programming

近年来,软件无线电(Software Defined Radio, SDR)凭借高度灵活性、易升级及易维护等优势获得广泛的关注,成为无线通信领域的研究热点^[1-3]。

SDR波形是指实现特定无线协议的软件组件集合^[3],通常具有明确的实时性约束^[4-5]。由于波形需要在任意时刻启动或停止,SDR系统需要支持多个实时性约束波形的动态运行。为了实现灵活的计算资源管理^[6-7],系统需要提供在线

的实时调度服务。采用流水线调度^[8],实时任务调度问题可简化为具有实时性约束的波形部署决策问题,即在满足波形实时性约束的同时确定波形组件在平台中运行的目标处理元件。

现有研究表明波形的实时性约束可描述为最小吞吐量和最大时延约束^[4-5]。为满足波形应用的实时性约束,文献[6-7]中提出了一种面向SDR的计算资源管理框架,并提出实时性约束下的应用映射算法,算法中考虑了平台中处理资源

* 收稿日期:2014-04-15

基金项目:新技术研究高校合作项目(KX142600035);通信网信息传输与分发技术重点实验室资助项目(9140C170301140C17087)

作者简介:吴尚峰(1986—),男,重庆涪陵人,博士研究生,E-mail:sfwu1104@126.com;

魏急波(通信作者),男,教授,博士,博士生导师,E-mail:wjhbhw@nudt.edu.cn

和处理器间的通信带宽资源对应用映射的影响。然而,该研究存在两点不足:第一是未考虑波形组件的内存开销;第二是未考虑波形组件间通信的处理开销和内存开销。由于流水线调度对组件排序和定时工作的简化是以大量缓存开销为代价的,因而考虑波形的内存开销是很有必要的。此外,SDR系统中可能不存在专用的通信子系统,处理元件之间的通信需要处理元件的参与和大量缓存资源,因此,考虑处理元件间通信的处理开销和内存开销也是非常必要的。

1 基于系统时隙的流水线调度

1.1 流水线调度

本文采用流水线调度作为系统运行时的任务执行模式。流水线调度是指周期性任务的执行时间存在一定形式的重叠,是一种非严格周期性的任务执行模式^[8-9]。图1为一个简单的流水线调度示例。

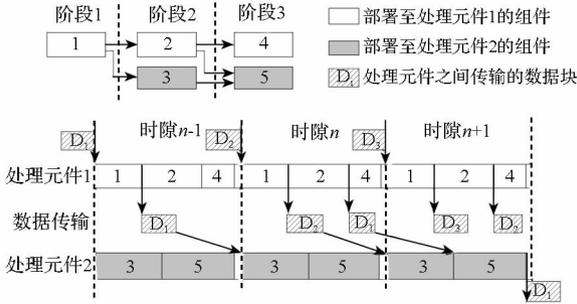


图1 流水线调度示例

Fig. 1 Example of pipelined scheduling

在流水线调度中,一个波形根据其内部组件间的优先关系被划分为多个流水线阶段。任意两个具有优先关系的组件被划分到前后不同的阶段中,因此,同一个阶段中的所有组件都是相互独立的。系统时隙是指系统中任务执行所采用的统一执行周期。在任意一个执行时隙内,每个组件最多被执行一次,而且组件执行所依赖的输入数据是由该组件的前驱组件在上一个时隙中执行后产生的。

本文采用流水线调度具有两方面优势:一是将任务调度中复杂的组件排序和定时工作替换为简单的参量检测,有效地降低了系统运行时的调度开销;二是利用组件与组件间通信的并发特性,消除了处理元件间的通信开销,因此可提供更高的波形吞吐量^[9]。

1.2 确定系统时隙

由于单个时隙内组件最多执行一遍,数据块

经过整个波形处理后产生的处理时延为波形阶段数目与系统时隙的乘积。为了保证波形的处理时延约束 L_w , 波形时隙 T_w 即波形的最大可行时隙应为

$$T_w = \frac{L_w}{N_w} \quad (1)$$

其中, N_w 为波形的流水线阶段数目。为了保证所有并发执行波形的处理时延约束,系统时隙 T_s 应为

$$T_s = \min\{T_w(i), i \in \mathbf{W}\} \quad (2)$$

其中, $T_w(i)$ 为波形 i 的波形时隙, \mathbf{W} 为并发执行的波形集合。

2 波形部署决策的系统模型

鉴于现有研究在内存开销方面考虑的不足,本文提出一个更为通用的系统模型,将波形的处理、内存以及通信带宽开销都纳入到波形部署决策过程之中。

2.1 统一计量单位

为构建统一的资源评估标准和资源管理模型,本文采用统一的计量单位。具体地,采用 MOPS 作为处理资源的计量单位,一些常用计量单位如时钟频率和 MIPS 均可等效地转换为 MOPS^[10]。此外,分别采用 MbPS(即 MbitPS)和 KB(即 KByte)作为通信带宽资源和内存资源的计量单位。

2.2 平台模型

假设 SDR 平台由多种处理元件构成且具有全互连网络架构,可以是板卡级或芯片级的嵌入式异构系统。此外,假定每个处理元件都有各自独立的本地内存和网络接口。平台模型的资源定义如下:

定义1 一个 SDR 平台表示为一个五元组 $(\mathbf{P}, pt, R, M, L)$ 。其中, \mathbf{P} 表示平台中的处理元件集合, $pt \in PT$ 指定处理元件的类型,如 GPP 或 DSP 等,函数 $R: \mathbf{P} \rightarrow \mathbb{N}_0$ 和函数 $M: \mathbf{P} \rightarrow \mathbb{N}_0$ 分别指定处理元件的处理资源大小和内存资源大小,函数 $L: \mathbf{P} \times \mathbf{P} \rightarrow \mathbb{N}_0$ 指定处理元件之间的通信带宽大小。

为便于数学运算,将平台的处理资源序列和内存资源序列分别建模为向量 \mathbf{P}_r 和向量 \mathbf{P}_m , 将各处理元件之间的通信带宽建模为通信阵列 \mathbf{L} 。此外,假设处理元件的内部通信都基于共享内存方式实现,不需要进行数据搬移,因此,处理元件内部通信带宽视为无穷大,相应地,通信矩阵 \mathbf{L} 的

对角线元素均视为无穷大。

图 2 和表 1 分别给出一个 SDR 示例平台及其各种属性信息。

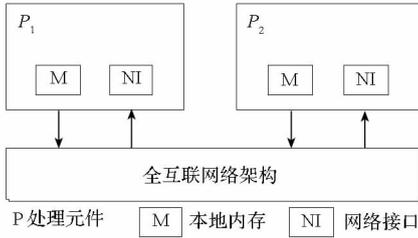


图 2 SDR 示例平台

Fig. 2 SDR example platform

表 1 SDR 示例平台属性

Tab. 1 Properties of the SDR example platform

	pt	R	M	$L(i, j)$	
P_1	GPP	1000	256	L_{12}	300
P_2	DSP	2000	512	L_{21}	500

2.3 应用模型

本文利用有向无环图 (Directed Acyclic Graph, DAG)^[11] 描述一个 SDR 波形, 图中节点和有向边分别代表波形组件和组件间的通信边。波形的资源需求定义如下:

定义 2 一个 SDR 波形表示为一个六元组 $(T, E, \Gamma, \Theta, Th_w, L_w)$ 。其中, T 和 E 分别表示波形的组件集合和通信边集合, 函数 $\Gamma: P \rightarrow \mathbb{N}_0^2$ 和函数 $\Theta: P \rightarrow \mathbb{N}_0^6$ 分别指定组件和通信边的各种资源需求, 而 $Th_w \in \mathbb{N}_+$ 和 $L_w \in \mathbb{N}_+$ 则分别指明波形的吞吐量约束和时延约束。进一步地, 函数 Γ 又表示为一个二元组 (τ, μ) , τ 和 μ 分别代表每个组件对所在处理元件的处理资源需求和内存资源需求; 函数 Θ 表示为一个六元组 $(\tau_{src}, \tau_{dst}, \mu_p, \mu_{src}, \mu_{dst}, \beta)$, 其中, μ_p 代表通信边位于单个处理元件内部时对所在处理元件的内存资源需求, τ_{src} , τ_{dst} , μ_{src} 和 μ_{dst} 分别代表通信边跨越不同处理元件时对其源端和目的端处理元件的处理资源需求和内存资源需求, β 则代表通信边的通信带宽需求。

与平台模型类似, 将波形组件的处理资源需求序列和内存资源需求序列分别建模为向量 T_τ 和向量 T_μ , 且将波形的各种资源需求分别建模为矩阵 $E_{\tau_{src}}, E_{\tau_{dst}}, E_{\mu_p}, E_{\mu_{src}}, E_{\mu_{dst}}$ 和 E_β 。由 DAG 图的内在优先关系可知, 任意组件只能接收其前驱组件的输出数据, 因而通信边的所有资源需求阵列均为上三角矩阵。

图 3 和表 2 分别描述了一个 SDR 示例波形

的 DAG 图表示及其相应的波形属性信息。

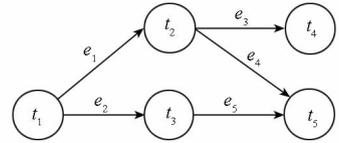


图 3 SDR 示例波形的 DAG 图表示

Fig. 3 The DAG presentation of the SDR example waveform

表 2 SDR 示例波形属性

Tab. 2 Properties of the SDR example waveform

	τ	μ		τ_{src}/τ_{dst}	$\mu_p/\mu_{src}/\mu_{dst}$	β
t_1	120	30	e_1	20/20	20/10/10	40
t_2	30	10	e_2	10/10	10/5/5	20
t_3	100	20	e_3	15/15	15/8/8	30
t_4	50	10	e_4	20/20	20/10/10	40
t_5	80	20	e_5	25/25	25/13/13	50

3 基于动态规划的波形部署决策算法设计

波形部署决策可分解为一系列的波形组件部署决策, 且顺序进行的组件部署决策满足“无后效性”, 可利用动态规划方法^[12]进行求解。

3.1 算法概述

基于动态规划的波形部署决策过程可描述为图 4 中从左至右依次进行的节点选路过程。

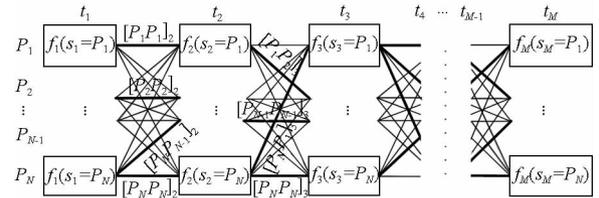


图 4 基于动态规划的波形部署决策过程示意图

Fig. 4 A sketch of the process of waveform deployment decision-making based on dynamic programming

图 4 中, 横向和纵向分别表示组件和处理元件。节点表示各个阶段可能的状态, 记录有该状态的最小可能代价值。节点的输入边对应数值代表在前一状态下部署当前组件到目标处理元件产生的新增代价值。节点选路, 即从节点所有输入边中选择最佳输入边, 策略是最小化节点累积代价值。选路完成后, 从最后一个阶段中具有最小代价值的节点出发, 从右向左沿已选定的路径回溯, 可获得最终的波形部署结果。

3.2 建立动态规划模型

给定 SDR 波形和平台, 假设波形的组件数目

为 M , 平台的处理元件数目为 N 。可根据如下定义建立波形部署决策的动态规划模型。

1) 阶段变量 $i \in [1, M]$;

2) 状态变量 $s_i \in \mathbf{S}_i = \{P_j | j=1, \dots, N\}$, \mathbf{S}_i 为可能状态集;

3) 决策变量 $u_i(s_i) \in \mathbf{U}_i(s_i) = \{[s_{i-1} \ s_i]\}$, $\mathbf{U}_i(s_i)$ 为允许决策集;

4) 部分子策略 $pl_i = \{u_1, u_2, \dots, u_i\}$ 和全过程策略 $pl_M = \{u_1, u_2, \dots, u_M\}$;

5) 状态转移: 当 $i=1$ 时, $s_1 = P_j, j=1, \dots, N$; 当 $2 \leq i \leq M$ 时, $s_i = T_i(s_{i-1}^*, u_i^*(s_i))$, 其中, $u_i^*(s_i)$ 和 s_{i-1}^* 分别表示当前状态作出的决策及其对应的前一状态;

6) 代价函数: 当 $i=1$ 时, 状态对应的代价为 $f_1(s_1) = g_1(s_1)$; 当 $2 \leq i \leq M$ 时, 状态输入边对应的新增代价为 $g_i(s_i, u_i)$, 状态的累积代价为 $f_i(s_i) = \min_{u_i \in \mathbf{U}_i(s_i)} \{f_{i-1}(s_{i-1}) + g_i(s_i, u_i)\}$;

7) 最优解: 波形部署决策的最小代价值为 $\min_{s_M \in \mathbf{S}_M} \{f_M(s_M)\}$ 。

基于上述基本定义, 构建式(3)所示的动态规划基本方程。

$$\begin{cases} f_1(s_1) = g_1(s_1) & s_1 \in \mathbf{S}_1 \\ f_i(s_i) = \min_{u_i(s_i) \in \mathbf{U}_i(s_i)} \{f_{i-1}(s_{i-1}) + g_i(s_i, u_i(s_i))\} & s_i \in \mathbf{S}_i \\ \mathbf{U}_i(s_i) = \{[s_{i-1} \ s_i] | s_{i-1} \in \mathbf{S}_{i-1}\} & s_i \in \mathbf{S}_i \\ \mathbf{S}_1, \mathbf{S}_{i-1}, \mathbf{S}_i = \{P_j | j=1, 2, \dots, N\} \\ i=2, \dots, M \end{cases} \quad (3)$$

3.3 代价函数

本文提出式(4)~(9)所示的代价函数以最小化波形对平台各种资源的占用总和。式(4)和式(5)分别用于计算部署第一个组件和后续组件产生的新增代价, 式中的 C_r, C_m 和 C_l 分别表示处理代价、内存代价和通信代价, 而 α_r, α_m 和 α_l 则为各代价的归一化权重。部署第一个组件时, 由于没有组件间通信, 因而仅产生处理代价和内存代价。部署后续组件时将产生所有的处理代价、内存代价以及通信代价, 分别利用式(6)~(8)计算得到。式(6)~(8)中, 顶部带有短横的变量表示当前状态所作决策对应前一状态的平台剩余资源数量, 而 t_k^* 表示组件 t_k 所在处理元件的编号。

$$g_1(s_1 = P_j) = \alpha_r \times \frac{T_r(1)}{P_r(j)} + \alpha_m \times \frac{T_m(1)}{P_m(j)} \quad (j=1, \dots, N) \quad (4)$$

$$\begin{aligned} g_i(s_i = P_j, u_i) \\ = \alpha_r \times C_r(i, j) + \alpha_m \times C_m(i, j) + \alpha_l \times C_l(i, j) \\ j=1, \dots, N; i=2, \dots, M \end{aligned} \quad (5)$$

$$\begin{cases} C_r(i, j) = \frac{T_r(i)}{P_r(j)} + \sum_{k=1}^{i-1} \left[\frac{X_1(k)}{P_r(j)} + \frac{X_2(k)}{P_r(t_k^*)} \right] \\ X_1(k) = \begin{cases} E_{\tau_{dst}}(k, i) & t_k^* \neq j \\ 0 & t_k^* = j \end{cases} \\ X_2(k) = \begin{cases} E_{\tau_{src}}(k, i) & t_k^* \neq j \\ 0 & t_k^* = j \end{cases} \end{cases} \quad (6)$$

$$\begin{cases} C_m(i, j) = \frac{T_m(i)}{P_m(j)} + \sum_{k=1}^{i-1} \left[\frac{Y_1(k)}{P_m(j)} + \frac{Y_2(k)}{P_m(t_k^*)} \right] \\ Y_1(k) = \begin{cases} E_{\mu_{dst}}(k, i) & t_k^* \neq j \\ E_{\mu_{\beta}}(k, i) & t_k^* = j \end{cases} \\ Y_2(k) = \begin{cases} E_{\mu_{src}}(k, i) & t_k^* \neq j \\ 0 & t_k^* = j \end{cases} \end{cases} \quad (7)$$

$$\begin{cases} C_l(i, j) = \sum_{k=1}^{i-1} \frac{Z(k)}{L(t_k^*, j)} \\ Z(k) = \begin{cases} E_{\beta}(k, i) & t_k^* \neq j \\ 0 & t_k^* = j \end{cases} \end{cases} \quad (8)$$

3.4 复杂度分析

动态规划算法的复杂度由问题规模以及代价函数的复杂度共同决定。假定波形组件数目为 M , 平台处理元件数目为 N , 且代价函数的复杂度为 ccf 。由式(4)可知, 部署第一个组件产生的计算复杂度为 $N \cdot ccf$; 由式(5)~(8)可知, 部署后续组件产生的总的计算复杂度为 $(M-1) \cdot N^2 \cdot ccf$ 。因此, 动态规划算法的复杂度如式(9)所示。

$$N \cdot ccf + (M-1) \cdot N^2 \cdot ccf \approx M \cdot N^2 \cdot ccf \quad (9)$$

4 仿真分析

通过仿真分析本文所提方案的性能, 并将结果与文献[6]中所采用的解决方案进行比较。

4.1 仿真环境设置

本文构建了如表3所示的5种仿真场景。其中, 场景1对应于文献[6]中采用的部署策略, 而后续4种场景是本文针对其不足而提出的改进策略。自适应权重是指根据波形各种资源需求总量与平台各种资源剩余总量的比值而获得的一种相对的归一化权重。

考虑实际波形存在的随机性, 将波形参数设置为表4所示的随机变量, 由波形生成程序自动生成实验波形。

表 3 5 种仿真场景

Tab. 3 Five types of simulation scenarios

仿真场景	是否考虑组件内存开销	是否考虑组件间通信的处理及内存开销	权重分配
场景 1	否	否	平均权重
场景 2	是	否	平均权重
场景 3	是	否	自适应权重
场景 4	是	是	平均权重
场景 5	是	是	自适应权重

表 4 波形随机变量及其分布函数

Tab. 4 Variables and distribution functions of waveforms

随机变量	分布函数
波形的组件数目	$U[5, 25]$
组件的处理资源需求	$U[1, 200]$
组件的内存资源需求	$U[1, 20]$
组件之间的通信资源需求	$U[1, 10]$

此外,设置了如图 5 所示的 4 套 SDR 平台,各平台具有相同的资源总量和不同的资源分布。图中,圆圈内数值表示处理元件的处理能力大小和内存大小,有向边旁数值表示处理器间通信带宽大小。

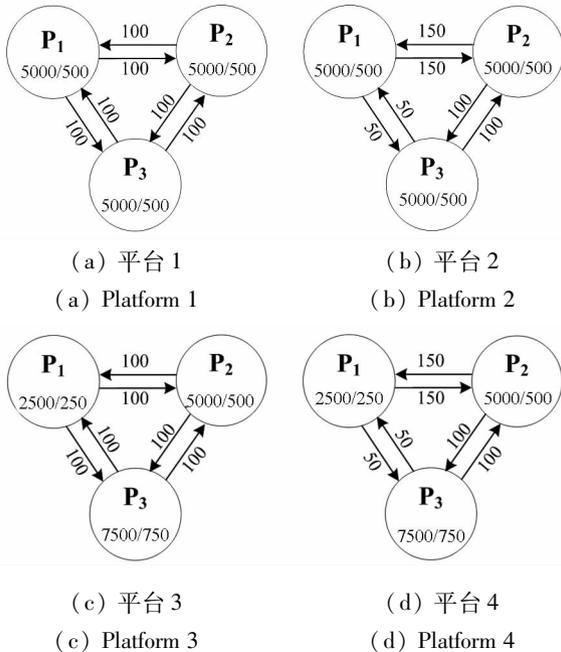


图 5 4 套 SDR 实例平台

Fig. 5 Four SDR example platforms

4.2 仿真结果分析

仿真测试选择的性能指标为平台可支持的平均最大波形数目和各种资源的平均资源占用率。利用表 4 所示随机变量产生实验波形,根据表 3

所示 5 种场景对应部署策略,将波形依次部署至图 5 所示 4 套平台,直至各平台均出现连续 5 次部署失败,表示平台不再支持更多波形。进行 10 000 次重复实验,获得各性能指标的统计值,如图 6 所示。

如图 6(a)所示,各平台支持的平均最大波形数目呈阶梯状下降,这表示组件的内存开销以及组件间通信的处理开销和内存开销均对波形部署决策构成影响,且组件间通信的处理开销和内存开销的影响尤为明显。

由于场景 1 仅考虑了组件的处理开销和组件间通信的通信带宽开销,而并未考虑组件的内存开销以及组件间通信的处理开销和内存开销,因此,处理开销和通信开销成为影响波形部署决策的关键因素。如图 6(b)和图 6(c)所示,场景 1 的处理资源利用率超过 0.96,而通信资源利用率仅为 0.2,这表明处理开销成为制约波形部署的瓶颈。然而,由图 6(c)可知,场景 1 对应部署方案中波形实际的内存资源需求总量达到平台内存资源总量的 1.7 倍。这说明场景 1 对应的平均最大波形数目是一种假象,采用场景 1 对应的部署策略得到的部署方案存在重大隐患,极有可能因内存不足而导致波形无法稳定运行。

由图 6(b)和图 6(c)可知,当未考虑组件间通信的处理开销和内存开销时,场景 2 和场景 3 的处理资源利用率与内存资源利用率都超过 0.9;然而,当考虑组件间通信的处理开销和内存开销时,场景 4 与场景 5 的处理资源利用率仅为 0.5 左右,但内存资源利用率却达到 0.8 ~ 0.95。这表明相对于组件的内存开销,组件间通信的内存开销对波形部署决策的影响更大。

由图 6(d)的所示,场景 4 和场景 5 的通信资源利用率明显高于场景 1 至场景 3,这说明当考虑组件间通信的处理开销和内存开销以后,单个处理元件的资源能力变得更为有限,同一个波形的组件被更多地部署到多个处理元件,由此产生了更多的通信带宽开销。

由图 6 可知,资源分布差异与权重分配方式对波形部署决策的影响较小,但采用自适应权重能够实现更少总体资源占用。此外,与文献[6]中采用的解决方案相比,本方案得到的平均最大波形数目和平均处理资源利用率均下降约 40%。

综上所述,本文提出的解决方案可更好地刻画组件及组件间通信的内存开销对波形部署决策的影响。

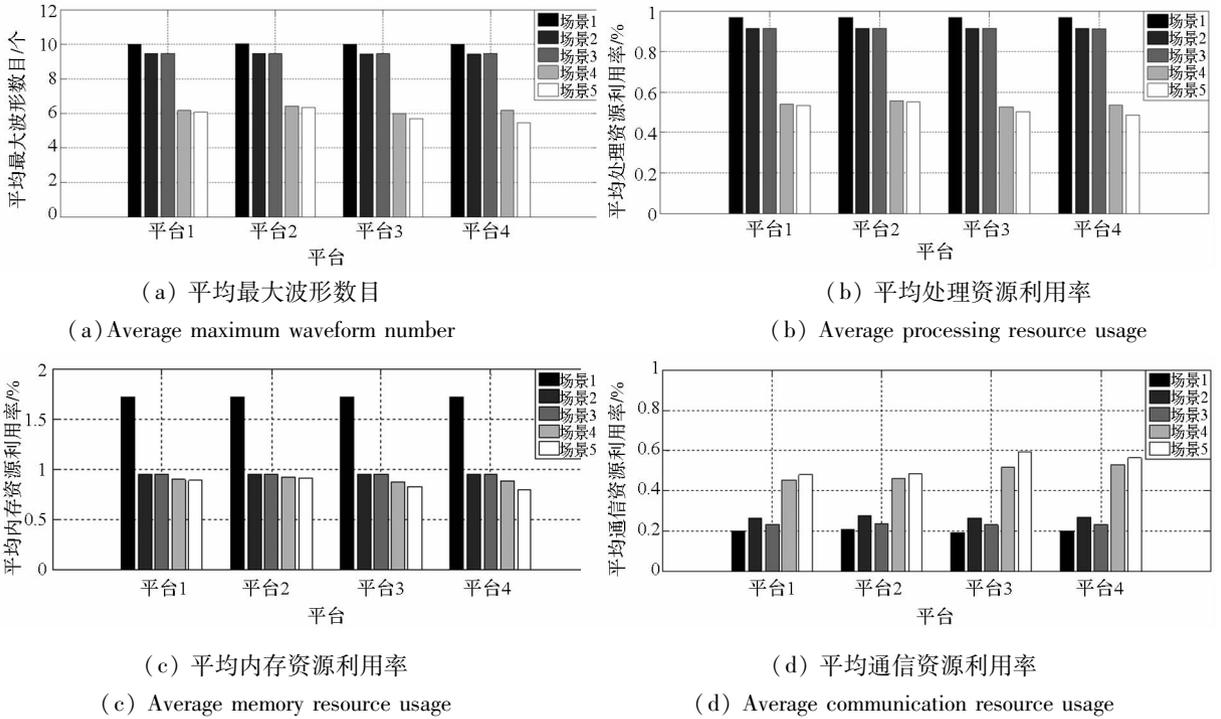


图 6 不同场景下各平台的平均最大波形数目和平均资源利用率

Fig. 6 Average maximum waveform number and various platform resource usages under different scenarios

5 结论

软件无线电系统需要支持多个具有吞吐量和时延约束波形的动态运行,通过采用流水线调度,将该问题转化为具有吞吐量和时延约束的波形部署决策问题。针对现有波形部署决策算法在内存开销方面考虑的不足,本文提出一个更为通用的波形部署决策系统模型,将波形的处理、内存以及通信带宽开销都纳入到波形部署决策过程之中;此外,设计一种基于动态规划的波形部署决策算法,算法目标为最小化波形对平台各种计算资源的占用总和。通过计算机仿真验证所提出模型和算法的性能,仿真结果表明内存对波形部署决策具有重要影响,与未考虑内存开销的方案相比,平台可支持的平均最大波形数目和平均处理资源利用率均下降约 40%。

参考文献 (References)

[1] Ulvers T. Software defined radio; challenges and opportunities [J]. IEEE Communications Surveys & Tutorials, 2010, 12 (4): 531 - 550.

[2] Ramacher U. Software-defined radio prospects for multi-standard mobile phones [J]. IEEE Computer, 2007, 40(10): 62 - 69.

[3] Jondral F K. Software-defined radio: basics and evolution to cognitive Radio [J]. EURASIP Journal on Wireless Communications and Networking, 2005, 3(1): 275 - 283.

[4] Geunx S J, Hausmansx J P H M, Bekooij M J G. Mapping of modal applications given throughput and latency constraints [C] // Proceeding of IEEE International Conference on Embedded Computer System, Samos, 2011: 373 - 379.

[5] Stuijk S, Basten T, Geilen M C W, et al. Multi-processor resource allocation for throughput? constrained synchronous dataflow graphs [C] // Proceedings of 44th ACM/IEEE Design Automation Conf., San Diego, 2007: 777 - 782.

[6] Marojevic V, Reves X, Gelonch A. A computing resource management framework for software-defined radios [J]. IEEE Transactions on Computers, 2008, 57(10): 1399 - 1412.

[7] Ismael G, Marojevic V, Gelonch A. ALOE: an open-source SDR execution environment with cognitive computing resource management capabilities [J]. IEEE Communications Magazine, 2011, 49(9): 76 - 83.

[8] Berthold U, Rhiemeier A R, Jondral F K. A pipelining approach to operating modular software defined radio [C] // Proceeding of IEEE Advances in Wired and Wireless Communication, Sarnoff, 2004: 201 - 204.

[9] Sriram S, Bhattacharyya S S. Embedded multi-processors scheduling and synchronization, second edition [M]. New York: CRC Press, 2009: 85 - 129.

[10] Mitola J. Software radio architecture: object-oriented approaches to wireless systems engineering [M]. New York: John Wiley & Sons, 2000: 437 - 438.

[11] Bajaj R, Agrawal D P. Improving scheduling of tasks in a heterogeneous environment [J]. IEEE Transactions on Parallel and Distribution Systems, 2004, (15) 2: 107 - 118.

[12] Bellman R. On the theory of dynamic programming [J]. Proceedings of the National Academy of Sciences of the USA, 1952, 38(8): 716 - 719.