

Procesamiento Distribuido y Paralelo de Bajo Costo Basado en Cloud&Movil

David Petrocelli¹, Armando De Giusti^{2,3}, Marcelo Naiouf²

¹ Accesibilidad de la información en los sitios web Argentinos (AISWA)
Universidad Nacional de Luján – UNLu

Ruta 5 y 7 – Luján – Buenos Aires – Argentina

² Instituto de Investigación en Informática LIDI (III-LIDI), Facultad de Informática,
Universidad Nacional de La Plata – CIC-PBA
50 y 120 – La Plata – Buenos Aires - Argentina

³ CONICET – Consejo Nacional de Investigaciones Científicas y Técnicas
dpetrocelli@unlu.edu.ar, {degiusti,mnaiouf}@lidi.info.unlu.edu.ar

Abstract. Actualmente el procesamiento intensivo se realiza a través de estructuras HPC híbridas (Grid, Cluster, Cloud) utilizando procesadores de arquitectura x86 y GPUs Nvidia, AMD o Intel, incurriendo en altísimos costos económicos y energéticos. Sin embargo, gracias a la evolución constante del hardware y con el advenimiento de los dispositivos móviles/microcomputadores con CPUs/GPUs ARM acompañado de la masividad de los mismos es posible pensar en una solución de bajo costo y consumo energético para solventar este tipo de problemas. Estos dispositivos incrementan su capacidad, eficiencia, estabilidad y potencia a diario, mientras ganan mercado, conservando un bajo costo, tamaño y consumo energético. A su vez, presentan lapsos de ociosidad, lo que representa una gran capacidad de recursos desaprovechados. Por tal motivo, el objetivo de este trabajo es presentar un prototipo de arquitectura distribuida dinámica, escalable y redundante geográficamente para explotar esta disponibilidad y realizar procesamiento intensivo aprovechando recursos y reduciendo costos.

Keywords: Computación en la nube, Computación móvil, Android, ARM, x86, GPU, Sistemas distribuidos, eficiencia energética.

1 Introducción

En el desarrollo inicial de los procesadores x86 (Intel/AMD) y sus correspondientes GPUs (Intel/AMD/Nvidia) las preocupaciones ambientales fueron débiles, y la atención al consumo energético fueron desvalorizadas. Si analizamos el contexto actual, en los centros de datos y procesamiento de cualquier organización el consumo eléctrico se está convirtiendo cada vez más en uno de los recursos más afectados y que contribuye al costo total de funcionamiento de una sala de procesamiento. Inclusive, dependiendo de la región y del uso de los recursos, el costo total de la energía puede exceder el costo de la adquisición de infraestructura de hardware. Entonces hoy, la energía y la eficiencia son ya una preocupación primordial para el diseño de cualquier sistema informático y se reconoce unánimemente que para alcanzar los retos

computacionales no sólo implica pensar en los objetivos matemáticos, sino también tener en cuenta el consumo energético, el desgaste al ambiente y los costos asociados.

Por tal motivo, desde otra rama de la computación, se desarrolló una línea de procesadores y chips donde por el contrario, el consumo energético es un factor clave del diseño y construcción de estos productos ya que los mismos están orientados a un mercado de dispositivos móviles/microdispositivos los cuales pueden estar alimentados por baterías con energía limitada. Si bien el rendimiento de estos procesadores y sus GPUs integradas es, naturalmente, menor y rezagada en comparación a los procesadores de propósito general, es interesante analizar su potencial y explotar sus capacidades. Al mismo tiempo, además de la potencia de hardware disponible, existe una fuerte masividad de estos dispositivos con grandes períodos de ociosidad mientras se encuentran sin tareas asignadas o recargando su batería. Este tiempo en desuso, en el pasado, fue aprovechado por arquitecturas colaborativas basadas en arquitecturas x86, por ejemplo, en proyectos de SETI@Home [1] para utilizar esa potencia de cálculo y construir un gran centro de datos para analizar señales, entre otras, que requerían gran poder computacional. En la actualidad, y siguiendo el uso de arquitecturas móviles, algunos autores [2,3,4] llevaron, aprovechando características similares, este modelo colaborativo a los dispositivos móviles.

Habiendo analizado la metodología, estructuras y resultados obtenidos de estos trabajos, el objetivo de ésta investigación es presentar ajustes y mejoras respecto a los estudios anteriores.

En primer lugar, la arquitectura diseñada se basa en un prototipo puramente distribuido, flexible y tolerante a fallos para dar soporte a las necesidades de los usuarios a través de un modelo para la gestión de la entrada y salida de dispositivos así como también de las tareas de división, ejecución de trabajos por nodo y recomposición basado en un modelo de recursos en la nube que se ajusta automáticamente en tamaño y capacidad dependiendo de la carga de trabajo, apartado no estudiado anteriormente. En segundo lugar, y siendo una de las motivaciones principales de este trabajo, el escenario de prueba elegido ataca directamente uno de los puntos que se considera de más vulnerabilidad para los procesadores ARM: las tareas de compresión y transcodificación de video, aspecto que fue solo mencionado en investigaciones anteriores. Además, se realiza un análisis comparativo de tiempo de procesamiento y consumo eléctrico implicado entre arquitecturas x86 y ARM.

El resto de este artículo se organiza de la siguiente manera. La sección 2 ofrece una visión general del estado del arte de los procesadores ARM y de los dispositivos móviles. La sección 3, presenta un análisis de factibilidad que justifica la realización de pruebas de compresión de video sobre arquitecturas ARM. Derivado del apartado anterior, la sección 4 define la arquitectura del modelo distribuido planteado. En el capítulo final se presentan las conclusiones y los trabajos futuros.

2 Estado del arte: ARM y los dispositivos móviles

La arquitectura de procesadores más utilizada en términos de dispositivos móviles corresponde a la empresa ARM (por sus siglas en inglés “Advanced RISC Machine”). Sus chips se basan en la arquitectura RISC (por sus siglas en inglés “Reduced

Instruction Set Computing”), la cual se compone por un conjunto de instrucciones reducidas. Una arquitectura RISC típica consta de un único archivo de registro uniforme, arquitectura de carga y almacenamiento y dispone de un modo de direccionamiento simple e instrucciones de longitud fija. Debido a estas características, la arquitectura consigue un alto rendimiento respetando un bajo consumo de energía [5] y un tamaño reducido de chip [6]. Por el contrario, los procesadores de arquitectura x86 tienen el problema de que poseen TDPs (por sus siglas en inglés “Thermal Design Power”) muy por encima de los que se manejan en los desarrollos de fabricantes basados en ARM como Qualcomm, Samsung o MediaTek. Según informes Qualcomm, el TDP para un smartphone está por debajo de los 2,5 W, mientras que para una tablet esa cifra debe ser inferior a los 5W [7]. Recién éstos últimos años, Intel ha podido crear procesadores de bajo consumo con la gama de los Intel Core M [8], con un TDP promedio de 4,5 W pero aún a rendimientos menores [9] o con algunos Intel Atom de bajo consumo, rondando en un TDP de 5 W [10].

Tomando el último producto lanzado por Samsung en 2017, el Samsung Galaxy 8, puede destacarse que se equipa con un procesador ARM de 64 bits, el Snapdragon 835, el cual ha alcanzado 8 núcleos a una frecuencia de 2.45 GHz cada uno con un tamaño de 10 nm, su GPU, la Adreno 540, da soporte a cálculos de punto flotante (en simple y doble precisión) y dispone de 256 cores con un promedio de 645 Mhz por cada uno de ellos, con soporte a OpenCL 3 [11,12] a una tasa muy baja de consumo energético (TDP < 3W) [13].

Por otra parte, además del aumento de la capacidad de hardware, los dispositivos móviles cuentan con otro aspecto interesante que diversas instituciones han investigado: la gran masividad y penetración en el mercado. Según la Unión Internacional de Telecomunicaciones (ITU), desde finales de 2016 existen aproximadamente 7.300 millones de usuarios móviles [15]. A su vez, en el mismo artículo, predicen que el número de dispositivos móviles per cápita se espera que llegue a 1,5 o 2 en 2020 [14]. Por su parte, Adam Lella [16] anunció estadísticas que se aproximan al 80% de adopción de teléfonos inteligentes en las personas a lo largo del mundo desde la misma fecha. Otras fuentes como Wikipedia [17] y WeAreSocial [18] estiman una penetración de estos dispositivos entre el 60 y 70 % a lo largo del planeta. Según IDC [19] se estima para 2017, se establecerán 1534 millones de ventas y entregas de dispositivos móviles. A través de los reportes de TechMalak y Statista [20], se estima que habrá un total de 2.3 billones de dispositivos inteligentes para finales de 2017

3 Análisis de Factibilidad: Compresión de video sobre arquitecturas ARM

Habiendo realizado un análisis del estado actual de los dispositivos móviles, la capacidad y evolución del hardware de los procesadores ARM, en esta sección se presenta como este trabajo contribuye a manejar los recursos disponibles para realizar tareas de compresión de video, teniendo en cuenta un bajo costo y consumo energético. Por lo tanto, se examina (a) cantidad y calidad de videos fuentes acordes para realizar la tarea de compresión, tamaño de fragmentos de video y parámetros básicos para la

compresión del video y (b) las exigencias individuales para realizar dicha tarea en segmentos acotados de las fuentes comparando dispositivos ARM y x86.

(a) Para generar una prueba que produzca una saturación en las arquitecturas de los procesadores, se analizaron trabajos y configuraciones previas [21, 22]. Por tal motivo, para este trabajo, se recopilaron 4 fuentes de video con propiedades características de ambientes de pruebas [23,24,25] las cuales contienen una amplia gama de características para realizar las pruebas de factibilidad. El resumen de las propiedades más importantes de cada video se detalla en la figura a continuación:

Propiedades de Videos elegidos para realizar las pruebas									
Nombre	Tamaño	Proporciones	Contenedor	Codec	Perfil	Nivel de Compresión	Tasa de bits	FPS	BPP
512.mp4	36 MB	416x240	MP4	AVC x264	BaseLine	@L1.3	512 kbps	24	0.21
2500.avi	210 MB	854x480	AVI	Lavc	Simple	@L1	2500 kbps	24	0.25
4000.mp4	339 MB	1920x1080	MP4	AVC x264	High	@L4.2	4000 kbps	60	0.03
8000.mp4	642 MB	3840x2160	MP4	AVC x264	High	@L5.1	8000 kbps	60	0.01

Fig. 1. Características principales de videos utilizados para las pruebas de compresión

Por su parte, se trabaja con 2 tamaños de fragmentos, medido en segundos, para el procesamiento en un nodo trabajador: 3 y 10 segundos, valores que se encuentran dentro del tamaño recomendado para el servicio de streaming de video [26, 27]. Respecto del proceso de transcodificación y re-empaquetamiento a realizar, se describen dos tareas a continuación. En ambos casos se empaqueta a archivos MP4 y se re-comprime con el códec AVC H.264, siendo x264 la librería de código abierto utilizada por la herramienta FFMpeg, siguiendo para la misma las referencias recomendadas [28].

El resumen de las propiedades de los perfiles de compresión se presenta a continuación:

Tabla de Perfiles de compresión de Video											
Perfil	Códec Video	Int. KeyFrames	Kbps Video	Perfil de Comp.	Niv. de Comp.	FPS	Preset	Códec Audio	Kbps Audio	Muestreo	Canales
Perfil 1	libx264	50	220	baseline	@L3.0	15	Rápido	aac	64	44100	2
Perfil 2	libx264	250	8000	high	@L5.1	30	Lento	aac	256	48000	6

Fig. 2. Características principales de los videos utilizados para las pruebas de compresión

(b) Para evaluar las exigencias individuales implicadas en las tareas de compresión se utilizó un conjunto de recursos x86 y ARM. La selección de dicho equipamiento busca presentar escenarios que permitan contrastar rendimiento y consumo energético, con el objetivo de justificar el modelo de este trabajo. Los dispositivos y características más relevantes se detallan a continuación:

Tabla de recursos utilizados para experimentación					
Nombre dispositivo	Tipo Procesador	Procesador	Características	TDP Max	Memoria
Samsung Galaxy J7 (2017)	ARM	Exynos 7870 (Cortex-A53)	8x1.6 Ghz	2.5 W	3 GB
Samsung Galaxy S7	ARM	Exynos 8890 (Cortex-A53)	4x2.3 GHz + 4x1.6 GHz	2.5 W	4 GB
BGH Positivo SchoolMate 15 TV	x86	Intel Celeron N2806	2x2 Ghz	4.5 W	4 GB
BGH Positivo A-497	x86	Intel i7 2630QM	4x2.00 GHz	45 W	4 GB
Pc de Escritorio Genérica	x86	Intel i7 6770k	8x4.0 Ghz	91 W	16 GB

Fig. 3. Características principales de videos utilizados para las pruebas de compresión

Con objeto de realizar las pruebas se instala FFMpeg (versión 3.3.2) en cada uno de los recursos ARM (Android) y x86 (Windows) y se toma el tiempo en milisegundos para cada tarea. Luego de ejecutar los procesos definidos en el punto (a) sobre los fragmentos de 3 y 10 de cada fuente de video, la tabla resultante es la siguiente:

Tabla de tiempos (en milisegundos) de las tareas de compresión especificadas						
Video	Perfil	Intel Celeron N2806 (4,5 W)	Intel i7 2630QM (45w)	Intel i7 6700k (91 w)	Samsung Galaxy J7 (2,5 W)	Samsung Galaxy S7 (2,5 W)
512_3segs	Perfil Base	1030	270	149	1426	1102
512_10segs	Perfil Base	4664	714	336	2521	1740
512_3segs	Perfil High	3771	739	347	3192	2388
512_10segs	Perfil High	14927	2747	1293	6885	3852
2500_3segs	Perfil Base	3854	660	330	4451	2273
2500_10segs	Perfil Base	8314	1530	733	10887	4398
2500_3segs	Perfil High	13861	2216	1145	7954	3826
2500_10segs	Perfil High	56180	8990	3562	20219	10729
4000_3segs	Perfil Base	8653	1330	544	5203	2854
4000_10segs	Perfil Base	26577	4494	1448	14164	7400
4000_3segs	Perfil High	15869	2990	1115	10970	5440
4000_10segs	Perfil High	142327	21059	7930	49634	27170
8000_3segs	Perfil Base	28992	11515	1673	9429	6644
8000_10segs	Perfil Base	104252	15630	4799	31384	18892
8000_3segs	Perfil High	48117	17120	3260	32988	22287
8000_10segs	Perfil High	424868	105970	21802	242997	141483

Fig. 4. Características principales de videos utilizados para las pruebas de compresión

A partir de dicha experimentación, la evaluación (b) es la siguiente:

- Los procesadores Intel i7 de arquitectura x86 presentan los mejores resultados de compresión, pero incurriendo en muy alto consumo energético, tomando como base la información de sus TDPs.

- Tomando el i7 6700k y el Samsung Galaxy S7, como mejores candidatos de cada arquitectura, para las diferentes pruebas se deduce que:

- El i7 realiza el proceso de compresión, en promedio, 5 veces más rápido que el chip ARM. Sin embargo, el consumo del es 36 veces mayor.

- Las tareas de compresión sobre fragmentos de 10 segundos presentan tiempos más cortos respecto de los de 3 segundos (4,5 contra 5,7 veces más lento) en proporción.

- Las tareas de compresión sobre el perfil High muestra una relación más positiva respecto de la Baseline (4,73 contra 5,46 veces más lento)

- Si se compara el Intel Celeron N2806 (x86) y Exynos 7870 (ARM), ambos de poco consumo energético, el procesador ARM realiza la tarea asociada 3 veces más rápido.

4 Arquitectura distribuida para HPC sobre ARM

4.1 Introducción

A partir de los resultados obtenidos se desprende y resulta factible la propuesta de este trabajo: La construcción de una arquitectura distribuida sobre dispositivos ARM para realizar tareas de cálculo intensivo utilizando la capacidad de cálculo disponible, el tiempo ocioso y la masividad de los mismos.

Cómo punto de partida, es importante destacar que la arquitectura aquí planteada se basa en un modelo híbrido compuesto por (a) recursos Cloud, y (b) por una red P2P de dispositivos móviles.

Los recursos Cloud (a) se utilizan para construir el sistema de administración, gestión, clasificación y control de dispositivos P2P, siendo además el punto redundante de acceso al sistema, gestión y balanceo de carga, así como también fragmentación y recomposición de segmentos de recursos. Por su parte, la red P2P (b) se utiliza, como cliente, para realizar las solicitudes de trabajo de procesamiento intensivo y para procesar de manera distribuida las solicitudes cuando el nodo actúa como servidor.

Para continuar con las pruebas de factibilidad realizadas, el modelo plantea disponer en cada nodo la herramienta FFMpeg.

4.2 Arquitectura del prototipo distribuido

El prototipo aquí presentado se compone de tres grandes partes: (a) Punto de Acceso al sistema; (b) Nodos de Administración y (c) Nodos Móviles

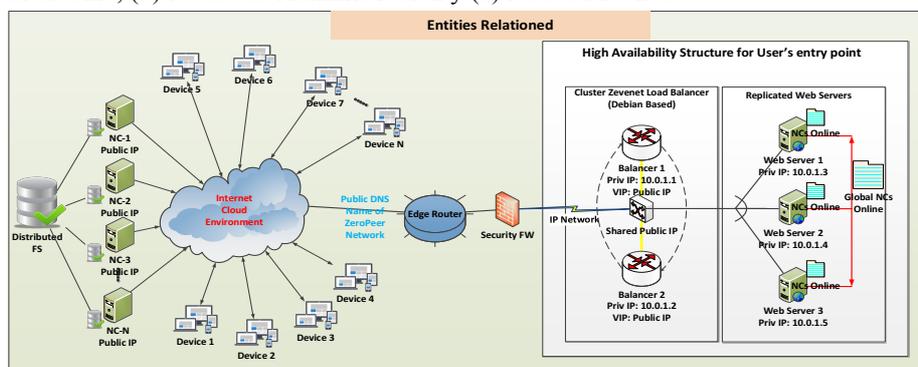


Fig. 5. Entidades básicas que se relacionan en el sistema

- Punto de Acceso al sistema (Nodos Cloud): La estructura de acceso al sistema evita tener puntos únicos de errores y se encarga de mantener replicado en al menos 2 nodos los datos necesarios para mantener el sistema disponible. Para ello, a través de la plataforma Zevenet (gratuita), se construye un balanceador en cluster que se encarga de manejar dos equipos para una única dirección IP virtual o dirección DNS. A su vez, el balanceador reparte la carga entre ambos nodos y mantiene el servicio disponible ante posibles fallas. También, detrás de esta capa de balanceo, se encuentra un conjunto de servidores Web replicados (2 o más) los cuales registran y administran cuáles nodos se encuentran disponibles para atender a los clientes. A su vez, estos servidores Web son los que los usuarios consultan, a través de la dirección DNS, para ingresar al sistema y obtener el listado actualizado de los nodos de administración que se encuentran actualmente activos.

- Nodos de Administración (Nodos Cloud): Los recursos Cloud se organizan para construir una red de administración flexible. Para ello, definen un conjunto de características:

En primer lugar, tienen la función de administrar su propia estructura flexible. Esto significa que la red de administración conoce el estado cada nodo y de la red en general, de manera de poder decidir si es necesario aumentar o reducir la cantidad de nodos que actualmente están atendiendo a los clientes, siempre respetando un mínimo de 3 nodos para garantizar tolerancia a fallos. En este primer modelo se plantea tener en cuenta, a nivel de conjunto, el análisis de las siguientes métricas: (a) Porcentaje de uso del procesador y memoria. (b) Cantidad de nodos clientes atendidos. (c) Cantidad de peticiones encoladas. En caso de disponer de valores de (a) superiores al del 80 %, de (b) por encima de 15 (por nodo) o (c) mayores a 10, se gestiona la creación de un nuevo nodo de administración. De igual manera, si la cantidad de (a) es menor al 40%, y la cantidad de nodos es superior al mínimo necesario (3), se apaga un nodo. Si algún recurso en la nube está apagado por más de una semana, se elimina con objeto de no incurrir en costos de reserva de almacenamiento.

En segundo lugar, estos nodos son el punto de acceso al desarrollo de actividades del sistema, motivo por el cual todos los dispositivos móviles, luego de haber obtenido la lista de los recursos Cloud, se conectan para solicitar tareas de cómputo intensivo.

En tercer lugar, relacionado con el punto anterior, la red de administración se encarga de (a) la gestión y (b) la clasificación de los dispositivos móviles que se conectan a la red. Respeto del punto (a), su tarea es administrar la entrada y salida de los recursos con objeto de mantener una lista de recursos de procesamiento disponible; relacionado con el apartado (b), se encarga de clasificar los dispositivos que se conectan según métricas estáticas y métricas dinámicas. Las primeras se componen de: Nombre del dispositivo, Sistema Operativo (familia y versión), información de hardware (CPU, memoria total, GPU) y posición geográfica. Con esas métricas se realiza una clasificación inicial en uno de tres grupos: 1. Gran capacidad de Hardware; 2. Capacidad promedio de Hardware; 3. Baja capacidad de hardware. Por su parte, los controles dinámicos se componen por un ajuste de esa evaluación inicial, donde una tarea se ejecuta cada 10 segundos y permite reclasificar el orden que obtienen dentro de cada grupo según varíen dinámicamente a lo largo del tiempo, siendo los mismos los siguientes: % de uso de CPU, memoria, GPU y ancho de banda de bajada y subida disponible.

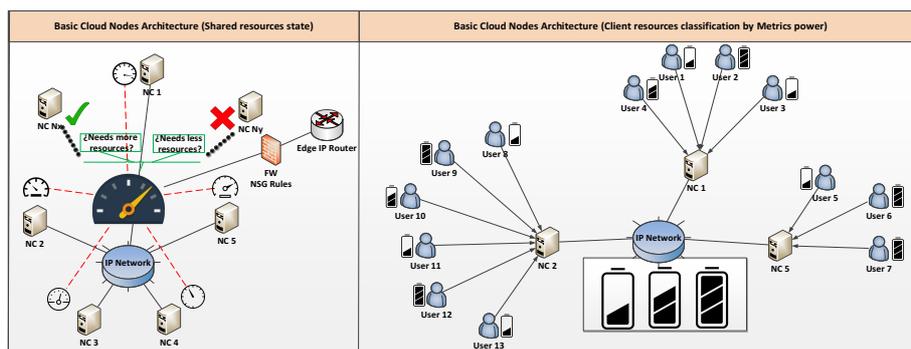


Fig. 6. A la izquierda representación de administración de recursos Cloud en base a carga del sistema; a la derecha ejemplificación de clasificación de dispositivos.

En cuarto lugar, a partir de las peticiones de los usuarios finales, el nodo de administración que lo atiende se encarga de gestionar la solicitud y derivar los fragmentos de tareas a procesar. Cuando la tarea se completa la devuelve al usuario final. Siguiendo la experimentación realizada en el análisis de factibilidad, cada nodo debe disponer la herramienta FFMpeg para realizar las tareas asociadas. Es importante destacar que el nodo Cloud dispone de la capacidad para gestionar los errores que pueden originarse en los nodos trabajadores. Para ello, mantiene para cada tarea una lista con la siguiente información {IDUsuario; IDTarea; IDSubtarea; EstadoTarea}, donde los estados pueden ser: {en cola, procesando, terminada, #errores}. Si una tarea por algún problema con el nodo trabajador llega corrupta, el dispositivo informa un error, o se vence el tiempo de espera, se marca el error y se valida si la cantidad de intentos fallidos es menor a 3. Si es mayor, se aborta la tarea, si es menor pasa a ser encolada nuevamente y se envía a un nuevo usuario para ser procesada.

- Nodos Móviles (Nodos P2P): Los dispositivos móviles se organizan como una red P2P para aportar su poder de cálculo en los tiempos que se encuentran ociosos. Para que un dispositivo sea considerado disponible deben darse un conjunto de condiciones: (a) se debe encontrar conectado a la red eléctrica con un tiempo mayor a 10 minutos y debe disponer de una carga de batería superior al 60%; (b) el equipo se debe encontrar con un uso de recursos menor al 10 % y (c) debe estar conectado a una red WiFi. Estos tres puntos buscan garantizar que, por un lado, no se afecte el uso de los dispositivos mientras los usuarios lo están utilizando, que no le produzca gasto al usuario y por otro, que estén conectados a la red desde un tiempo suficiente con objeto de no incluir demasiados equipos que entren y salgan constantemente a la red, evitando demasiadas tareas de gestión y problemas de reasignación de trabajos.

Como estos dispositivos construyen una red P2P, pueden cumplir dos roles: (a) Clientes o (b) Servidores de la red. Siguiendo la experimentación realizada, en ambos casos se debe incluir la herramienta FFMpeg para realizar las tareas asociadas.

En el caso que un nodo actúe como cliente, una vez conectado a la red de administradores solicita una tarea, en este caso de compresión, para la cual define un conjunto de perfiles y parámetros de compresión que desea obtener. Para cada perfil describe: códec de video y audio, cantidad de bits de audio y video, perfil de compresión, dimensión de pantalla, aspecto de pantalla, cuadros por segundo, entre otros.

Finalmente, dependiendo de las capacidades del dispositivo móvil emisor pueden darse dos casos: El primero es que disponga de la suficiente potencia para fragmentar el video e ir enviando a la red de administración pequeños segmentos de tamaños variable, siempre entre 2 y 10 segundos, para realizar la tarea de compresión. A su vez esto implica que en la medida que la red comprima los fragmentos, los reciba el nodo de administración, éste último debe devolverlos directamente al cliente final para que lo reconstruya. El segundo caso, es que sea un dispositivo de baja potencia y obligatoriamente deba subir el video completo a la red para realizar el proceso. En este caso deriva no solo la tarea de compresión al sistema sino que también los procesos de fragmentación y unificación a la red de administración.

En el caso que el nodo actúe como servidor, es tarea del mismo llevar a cabo los procesos de cómputo intensivo a partir de los datos y parámetros que se reciben por parte de la red de administración.

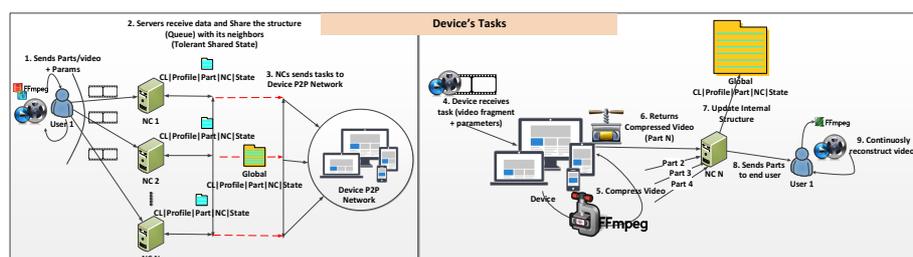


Fig. 7. A la izquierda representación del funcionamiento de un dispositivo móvil como cliente, a la derecha como servidor.

En este caso, a partir de los fragmentos de video y de los parámetros especificados, utilizará los mismos para ejecutar la tarea con la herramienta FFMpeg y se realice así el proceso de compresión. Una vez finalizada la tarea, se devuelven los resultados a la

red de administración. En caso que no pueda realizar la tarea, falle el proceso de compresión, o se desconecte de la red, debe avisar al nodo de gestión para que éste reasigne la tarea.

5 Conclusiones

Al inicio de este artículo, se investigaron y analizaron las características, evolución y masividad de los recursos ARM y se comprobó la factibilidad de realizar procesamiento intensivo y cómo es que estos procesadores presentan ahorros significativos respecto del consumo energético para dichas tareas. Finalmente, se propuso un modelo de arquitectura distribuida con dispositivos móviles para realizar tareas de cálculo intensivo de manera flexible y escalable explotando la capacidad de hardware ya disponible y ociosa. La propuesta se basa en varias observaciones, entre ellas las siguientes: a) la cantidad masiva de estos dispositivos; b) La ociosidad de estos recursos; y c) Los beneficios significativos que dicha infraestructura podría generar en términos de costos y procesamiento para diversos sectores de la comunidad. Finalmente, la misma se considera lo suficientemente robusta para afrontar diversas evaluaciones y afrontar los retos de este tipo de aplicaciones.

6 Trabajos Futuros

A partir de los aspectos investigados en este trabajo, se desencadenan un conjunto de actividades a realizar en etapas posteriores y que complementan el trabajo aquí desarrollado:

- Construir el prototipo del sistema, con el objetivo de poder evaluar la estructura de manera de obtener resultados concretos.
- Incluir dispositivos de diversas características y evaluar cómo responde la escalabilidad del sistema en la medida que existe mayor cantidad de recursos y cantidad de tareas a procesar.
- Incluir las capacidades de Hardware Accelerator de FFMpeg para evaluar rendimientos no solo con CPU sino también con las capacidades GPUs con OpenCL.
- Implementar otros servicio de algoritmia paralela que permitan realizar cálculos sobre las tarjetas gráficas. En ambos casos comprobar los resultados obtenidos entre arquitecturas x86 y ARM.

7 Referencias

1. David P. Anderson.: BOINC: A System for Public-Resource Computing and Storage. GRID '04 Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing. Pages 4-10, (2004)
2. Foster, I., Kesselman, C.: Computing while charging: building a distributed computing infrastructure using smartphones. CoNEXT '12 Proceedings of the 8th international conference on Emerging networking experiments and technologies pages 193-204, Nice, France. (2012)
3. Sriraman K R, Lead Architect.: Grid Computing on Mobile Devices - A Point of View. IEEE Xplore Conference: Grid Computing. Proceedings. Fifth IEEE/ACM International Workshop on (2004)

4. A New Purpose for Old Smartphones: Cluster Computing, <https://www.hpcwire.com/2015/01/27/new-purpose-old-smartphones-cluster-computing/>
5. ARM architecture, <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.subset.architecture.reference/index.html>
6. ARM's Cortex M: Even Smaller and Lower Power CPU Cores, <http://www.anandtech.com/show/8400/arms-cortex-m-even-smaller-and-lower-power-cpu-cores>
7. Qualcomm aims at 2.5 to 3W TDP for phones, <http://fudzilla.com/31532-qualcomm-aims-at-25-to-3w-tdp-for-phones>
8. Core M: paso atrás en potencia, paso adelante para el formato híbrido, <https://www.xataka.com/componentes/core-m-paso-atras-en-potencia-paso-adelante-para-el-formato-hibrido>
9. Versus: Intel Core M 5y10 vs Qualcomm Snapdragon 835, <https://versus.com/en/intel-core-m-5y10-vs-qualcomm-snapdragon-835>
10. Intel Atom z8000 Processor Series, <https://www.intel.com/content/dam/www/public/us/en/documents/datasheets/atom-z8000-datasheet-vol-1.pdf>
11. Qualcomm Details Snapdragon 835: Kryo 280 CPU, Adreno 540 GPU, X16 LTE, <http://www.anandtech.com/show/10948/qualcomm-snapdragon-835-kryo-280-adreno-540>
12. Samsung Galaxy S8 Review, <https://www.digitaltrends.com/cell-phone-reviews/samsung-galaxy-s8-review/>
13. Aaron Carroll, Gernot Heiser.: An analysis of power consumption in a smartphone USENIXATC'10 Proceedings of the USENIX conference on USENIX annual technical conference. Pages 21-21. Boston, MA — June 23 – 25. (2010)
14. Visual Networking Index, <http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/vni-infographic.html>
15. Statistics 2017. ITU. <http://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx>
16. U.S. Smartphone Penetration Surpassed 80 Percent in 2016, <https://www.comscore.com/Insights/Blog/US-Smartphone-Penetration-Surpassed-80-Percent-in-2016>
17. List of countries by smartphone penetration, https://en.wikipedia.org/wiki/List_of_countries_by_smartphone_penetration
18. DIGITAL IN 2017: GLOBAL OVERVIEW, <https://wearesocial.com/special-reports/digital-in-2017-global-overview>
19. Smartphone Volumes Expected to Rebound in 2017, <http://www.idc.com/getdoc.jsp?containerId=prUS42334717>
20. Number of smartphone users worldwide from 2014 to 2020 (in billions), <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>
21. Encoding. Lee, Byoung-Dai.: Empirical Analysis of Video Partitioning Methods for. International Journal of Multimedia and Ubiquitous Engineering. 10. 81-90. (2015).
22. Adriana Garcia, Hari Kalva, Borko Furht.: A study of transcoding on cloud environments for video content delivery. MCMC '10 Proceedings of the 2010 ACM multimedia workshop on Mobile cloud media computing Pages 13-18. Firenze, Italy — October 29 – 29. (2010)
23. Xiph.org Video Test Media [derf's collection], <https://media.xiph.org/video/derf/>
24. ITEC – Dynamic Adaptive Streaming over HTTP – Dataset, http://www-itec.uni-klu.ac.at/dash/?page_id=207
25. Download Guide: Big Buck Bunny, Sunflower versión, <http://bbb3d.renderfarming.net/download.html>
26. Optimal Adaptive Streaming Formats MPEG-DASH & HLS Segment Length. <https://bitmovin.com/mpeg-dash-hls-segment-length/>
27. Choosing the Optimal Segment Duration, <https://streaminglearningcenter.com/2016/08/04/choosing-the-optimal-segment-duration/>
28. Linux Encoding - x264 FFmpeg Options Guide, <https://sites.google.com/site/linuxencoding/x264-ffmpeg-mapping>