

INSTITUTO MEXICANO DE TECNOLOGÍA DEL AGUA
COORDINACIÓN DE HIDROLOGÍA
SUBCOORDINACIÓN DE HIDROMETEOROLOGÍA

IMPLEMENTACIÓN Y ADAPTACIÓN DEL MODELO
HURRICANE WEATHER RESEARCH AND FORECASTING
PARA EL PRONÓSTICO DE TRAYECTORIAS DE
HURACANES CON AFECTACIÓN EN MÉXICO

TH1611.1

INFORME FINAL

INDALECIO MENDOZA URIBE

NOVIEMBRE 2016

CONTENIDO

I.	ANTECEDENTES	3
II.	INTRODUCCIÓN	4
1.	REVISIÓN DE DOCUMENTACIÓN Y ARTÍCULOS CIENTÍFICOS RELACIONADOS CON EL MODELO HWRF.....	6
1.1.	CARACTERÍSTICAS DEL MODELO HWRF	6
1.2.	PRUEBAS DE FUNCIONALIDAD DEL MODELO HWRF	9
1.3.	EL MODELO HWRF EN MÉXICO	11
2.	DESCARGA, INSTALACIÓN, CONFIGURACIÓN Y PRUEBAS DEL MODELO HWRF	14
2.1.	ASPECTOS GENERALES DEL MODELO HWRF.....	14
2.2.	ESTRUCTURA DEL DIRECTORIO DEL CÓDIGO FUENTE	17
2.3.	ESTRUCTURA DEL DIRECTORIO DE DATOS DE ENTRADA	21
2.4.	ESTRUCTURA DEL DIRECTORIO DE PRODUCCIÓN.....	25
2.5.	COMPONENTES DEL MODELO HWRF	26
2.6.	OBTENCIÓN DEL CÓDIGO FUENTE	26
2.7.	DEFINICIÓN DEL DIRECTORIO PRINCIPAL DEL HWRF.....	27
2.8.	REQUERIMIENTOS DEL SISTEMA, LIBRERÍAS Y HERRAMIENTAS.....	28
2.9.	LIBRERÍAS INCLUIDAS.....	29
2.10.	COMPILACIÓN DE LOS MODULOS DEL HWRF	30
2.10.1.	COMPILANDO WRF-NMM.....	31
2.10.2.	COMPILANDO HWRF-UTILITIES.....	33
2.10.3.	COMPILANDO POM-TC.....	36
2.10.4.	COMPILANDO GFDL VORTEX TRACKER	38
2.10.5.	COMPILANDO NCEP COUPLER	39
2.10.6.	COMPILANDO WPS.....	40
2.10.7.	COMPILANDO UPP	41
2.10.8.	COMPILANDO GSI.....	43
2.11.	CONFIGURACIÓN Y PRUEBAS DEL MODELO HWRF	45
2.11.1.	CONFIGURAR EL DOMINIO HWRF.....	50
2.11.2.	EJECUTAR WPS	50
2.11.3.	EJECUCIÓN DE PREP_HYBRID	51

2.11.4.	EJECUCIÓN DE REAL	51
2.11.5.	EJECUCIÓN DE GSI: GSI_WRFINPUT	52
2.11.6.	EJECUCIÓN DE WRF ANALISYS	52
2.11.7.	EJECUCIÓN DE WRF GHOST.....	53
2.11.8.	EJECUCIÓN DE TRACKER ANALISYS	53
2.11.9.	PRUEBAS DEL MODELO HWRF	54
3.	DEFINICIÓN DE PRODUCTOS GRAFICOS.....	56
4.	CASO DE ESTUDIO: HURACAN PATRICIA (2015).....	59
4.1.	PRONÓSTICO DE HWRF INICIALIZADO EL 19 DE OCTUBRE A LAS 00 UTC	60
4.2.	PRONÓSTICO DE HWRF INICIALIZADO EL 19 DE OCTUBRE A LAS 12 UTC	60
4.3.	PRONÓSTICO DE HWRF INICIALIZADO EL 20 DE OCTUBRE A LAS 00 UTC	61
4.4.	PRONÓSTICO DE HWRF INICIALIZADO EL 20 DE OCTUBRE A LAS 12 UTC	61
4.5.	PRONÓSTICO DE HWRF INICIALIZADO EL 21 DE OCTUBRE A LAS 00 UTC	62
4.6.	PRONÓSTICO DE HWRF INICIALIZADO EL 21 DE OCTUBRE A LAS 12 UTC	62
4.7.	PRONÓSTICO DE HWRF INICIALIZADO EL 22 DE OCTUBRE A LAS 00 UTC	63
4.8.	PRONÓSTICO DE HWRF INICIALIZADO EL 22 DE OCTUBRE A LAS 12 UTC	63
4.9.	PRONÓSTICO DE HWRF INICIALIZADO EL 23 DE OCTUBRE A LAS 00 UTC	64
4.10.	PRONÓSTICO DE HWRF INICIALIZADO EL 23 DE OCTUBRE A LAS 12 UTC.....	64
	CONCLUSIONES	65
	BIBLIOGRAFIA.....	67
	APENDICE A. MANUAL DE INSTALACIÓN Y OPERACIÓN DEL MODELO HWRF	68
	APENDICE B. CONTENIDO DEL DVD	84

I. ANTECEDENTES

Por sus condiciones geográficas, México es vulnerable a recibir afectación cada año de ciclones tropicales, también conocidos como huracanes. La formación de estos sistemas meteorológicos, de forma indistinta, se dan tanto en el océano pacífico como en el Golfo de México, Mar Caribe y Océano Atlántico. Entre los desastres naturales asociados a los huracanes se encuentran: inundaciones, deslaves, pérdidas de cultivos, obstrucción de caminos, miles de damnificados y pérdidas económicas millonarias.

El Centro Nacional de Huracanes de Estados Unidos (NHC), ha sido nombrado por la Organización Meteorológica Mundial (OMM) como Centro Meteorológico Regional para el Atlántico Norte y Pacífico Este. El NHC emite advertencias, avisos y da seguimiento a la evolución de todas las tormentas tropicales de la región, pero una vez que estas se disipan y pierden su categoría, se dejan monitorear y no se da seguimiento a los remanentes derivados de dichas tormentas en días posteriores.

En México, la Coordinación General del Servicio Meteorológico Nacional (CNSMN), es el organismo oficial encargado de proporcionar información sobre el estado del tiempo a escala nacional y local en nuestro país. La CGSMN, depende de la Comisión Nacional del Agua (CONAGUA), la cual forma parte de la Secretaría de Medio Ambiente y Recursos Naturales (SEMARNAT). Para la emisión de avisos y seguimiento de tormentas tropicales en territorio y aguas nacionales, la CGSMN depende la información proporcionada por el NHC y no cuenta actualmente con un producto alternativo para el pronóstico de trayectorias de huracanes.

En la Subcoordinación de Hidrometeorología del Instituto Mexicano de Tecnología del Agua, se cuenta con experiencia en la implementación y operación de modelos numéricos de la atmósfera y oceánicos, como es el caso de los modelos MM5, WAM, SWAM, WRF, entre otros. Además, una de las fortalezas del grupo corresponde a la infraestructura computacional que permite la ejecución de forma operativa de este tipo de modelos.

II. INTRODUCCIÓN

El modelo Hurricane Weather Research and Forecasting (HWRF) es un modelo acoplado oceano-atmosfera para el pronóstico de huracanes que se ejecuta operacionalmente por la National Oceanographic and Atmospheric Administration (NOAA) y la National Centers for Environmental Prediction (NCEP) para proporcionar orientación al National Hurricane Center (NHC).

Una versión revisada del modelo HWRF típicamente es implementada de forma operativa en los meses de junio, al comienzo de la temporada de huracanes del Atlántico Norte. La versión actualizada resulta de la transición de las operaciones y desarrollos conducidos por el Environmental Modeling Center (EMC) y otros laboratorios gubernamentales e instituciones académicas, así como de pruebas controladas realizadas por el Developmental Tested Center (DTC) y el Environmental Modeling Center (EMC).

El componente atmosférico del HWRF es una configuración del modelo WRF que ha sido diseñado para simular y predecir ciclones tropicales (Bao et al. 2012, Gopalakrishnan et al. 2012). Esto incluye el Non-hydrostatic Mesoscale Model (NMM) núcleo dinámico con soporte telescópico, vortex para dar seguimiento al movimiento de los dominios. La suite de física incluye las parametrizaciones Simplified Arakawa-Schubert (SAS) esquema de cúmulos, el modelo de capa superficial Geophysical Fluid Dynamics Laboratory (GFDL) modelo de capa superficial y parametrización de radiación, el Global Forecasting System (GFS) parametrización de capa limite, y el esquema Tropical Ferrier microphysics.

El componente oceánico de HWRF es una versión del Princeton Ocean Model adaptado para ciclones tropicales (POM-TC), que fue desarrollado en la University of Rhode Island (URI). Los componentes atmosféricos y oceánicos se comunican a través de un acoplador desarrollado en NCEP/EMC.

El post-procesamiento de HWRF hace uso del procesador post-procesador unificado (UPP) NCEP y del GFDL seguidor de vórtice externo, que puede extraer la ubicación, intensidad y estructura del ciclón tropical desde la salida del modelo.

Cuando una perturbación o tormenta tropical está presente, el HWRF se inicializa cada seis horas y se agota a 126 horas. Los campos atmosféricos a gran escala se inicializan utilizando el Sistema de Asimilación de Datos HWRF (HDAS), que emplea el Sistema de Interpolación

Estadística Gridpoint (GSI) para asimilar observaciones convencionales usando la predicción de 6 h del Sistema Global de Asimilación de Datos (GDAS).

Se utiliza una técnica de inicialización de vórtice para mejorar su punto de inicio. Cuando la intensidad de tormenta observada es mayor o igual a 16 ms^{-1} , el vórtice HDAS se elimina y el vórtice de las previsiones HWRF anteriores de 6 h es Insertado, después de su ubicación, intensidad y estructura se corrigen según observaciones. Cuando la tormenta observada es débil (intensidad inferior a 16 ms^{-1}), el vórtice HDAS en sí es reubicado y corregido, por lo que el vórtice de la previsión HWRF anterior no se utiliza. Como un caso especial, el primer ciclo de una tormenta se inicializa utilizando un vórtice falso cuando la intensidad observada es mayor o igual a 30 ms^{-1} , o utilizando el HDAS corregido Vórtice en caso contrario.

Un proceso de inicialización oceánica basado en características genera condiciones iniciales para el componente oceánico POM-TC. Los componentes atmosféricos y oceánicos del HWRF se ejecutan en paralelo e intercambian información a través del acoplador: el modelo atmosférico calcula y envía el impulso y los flujos sensibles y térmicos al océano, mientras que el modelo oceánico envía la temperatura superficial del mar a la atmósfera.

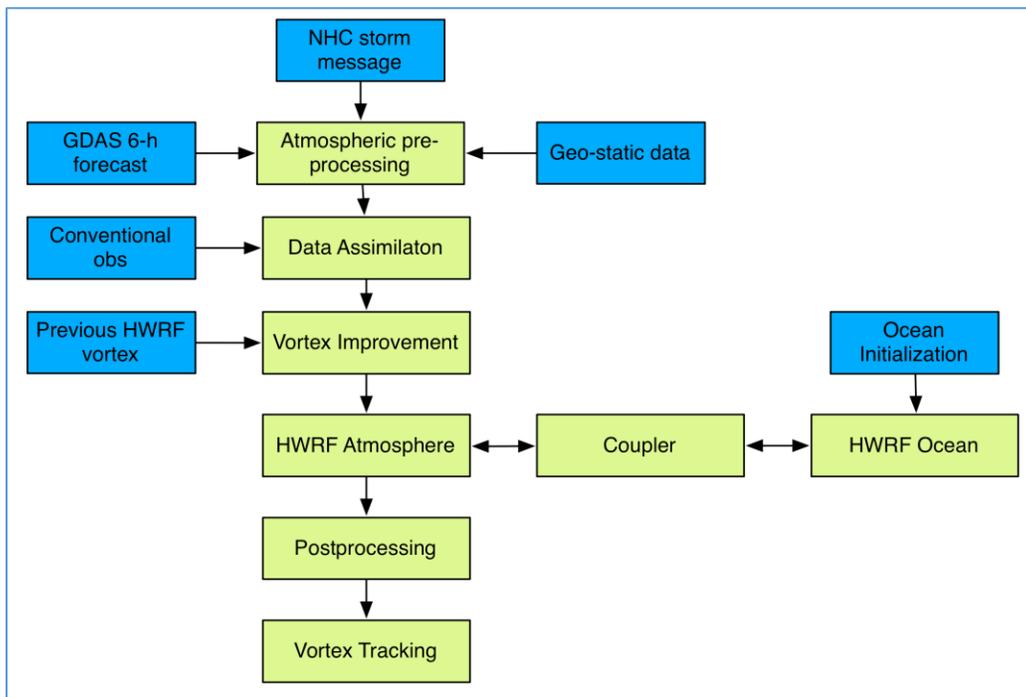


Figura II.1. Diagrama esquemático del HWRF componentes. Las casillas verdes representan los ocho componentes HWRF.

1. REVISIÓN DE DOCUMENTACIÓN Y ARTÍCULOS CIENTÍFICOS RELACIONADOS CON EL MODELO HWRF

Como principal fuente de información acerca del modelo WRF para Huracanes, el DTC mantiene un sitio web (<http://dtcenter.org/HurrWRF/users>) donde los usuarios pueden obtener las diferentes versiones del código HWRF, una guía de usuarios, documentación científica y conjuntos de datos de prueba. Las emisiones del código estable probado están planificadas en un calendario anual, que corresponde a la configuración utilizada en las operaciones de ese año temporada de huracanes. Los usuarios tienen acceso completo a puntos de referencia extensos del código de la comunidad. Adicionalmente, se puede consultar una mesa de ayuda contactando al correo electrónico wrfhelp@ucar.edu.

Si bien el acceso al código HWRF por medio de la página web satisface las necesidades de la mayoría de los usuarios, no aborda todos los requisitos de los desarrolladores del HWRF, especialmente aquellos que están trabajando en el entorno operativo, aquellos que trabajan en el modo de desarrollo rápido, orientado a la implementación operacional en un período de tiempo de 1-5 años, necesitan trabajar en estrecha colaboración entre sí y el EMC. Necesitan acceso a versiones experimentales de última generación del código, y no al modelo de estaciones fijas. Para apoyar esa actividad, el DTC ahora está proporcionando acceso directo al repositorio de código a expertos, desarrolladores colaboradores. Pueden obtener las últimas versiones del código y usar “ramas” en el repositorio de código para agregar sus contribuciones, que pueden ser vistas y compartidas con otros desarrolladores.

1.1. CARACTERÍSTICAS DEL MODELO HWRF

El componente atmosférico del HWRF es una configuración del modelo Weather Research and Forecasting (WRF) que ha sido diseñado para predecir ciclones tropicales, e incluye el núcleo dinámico del modelo Nonhydrostatic Meso-scale Model con módulo de seguimiento del dominio del vórtice, el esquema de cúmulos Simplificado de Arakawa-Schubert, la parametrización de la capa superficial Geophysical Fluid Dynamics Laboratory (GFDL), la parametrización de la capa límite planetaria del Sistema de Pronóstico Global (GFS) y el esquema de microfísica Tropical Ferrier. El componente oceánico de HWRF es una versión del modelo oceánico de Princeton adaptado para ciclones tropicales (POM-TC), que fue desarrollado en la Universidad de Rhode Island (URI). El componente oceánico de HWRF es una versión del Princeton Ocean Model adaptado para ciclones tropicales (POM-TC), que fue desarrollado en la University of Rhode Island (URI). Los componentes atmosféricos y

oceánicos se comunican a través de un acoplador desarrollado en NCEP/EMC. El post-procesamiento de HWRF hace uso del procesador post-procesador unificado (UPP) NCEP y del GFDL seguidor de vórtice externo, que puede extraer la ubicación, intensidad y estructura del ciclón tropical desde la salida del modelo.

La figura 1.1 presenta un diagrama de flujo esquemático de los componentes del modelo HWRF de la comunidad. Los mensajes de tormenta Emitidos por el Centro Nacional de Huracanes, incluyendo la localización e intensidad de las tormentas, se usan para definir el dominio HWRF. El preprocesador WRF (WPS, por sus siglas en inglés) se utiliza para generar condiciones preliminares iniciales y de contorno, las cuales se usan en el proceso de inicialización de vórtices para mejorar la representación del vórtice inicial. Si se dispone de una previsión previa de HWRF de 6 h, se utiliza en el proceso de ciclismo de vórtice; De lo contrario, un vórtice falso basado en la climatología del HWRF se utiliza. El sistema de asimilación de datos variacionales en 3-D de la interpolación estadística Gridpoint (GSI) se utiliza para ingerir Datos en los campos iniciales. Un proceso de inicialización del océano genera las condiciones iniciales para el componente oceánico POM-TC. Los componentes atmosféricos y oceánicos del HWRF funcionan entonces en paralelo e intercambiar información a través de un acoplador: el modelo atmosférico calcula y envía los flujos de impulso y calor al océano, mientras que el modelo oceánico envía la temperatura de la superficie del mar a la atmósfera. El Post-Procesador Unificado se utiliza para de-escalonar el pronóstico, para interpolarlo a una cuadrícula de latitud-longitud regular sobre los niveles de presión, y para calcular las cantidades derivadas. Por último, el rastreador de vórtices GFDL se utiliza para obtener propiedades del ciclón tropical previsto, como la ubicación y la intensidad.

El modelo HWRF fue desarrollado en el NCEP Environmental Modeling Center basado en la versión 2 del WRF y comenzó sus operaciones en la temporada de huracanes de 2007. Durante su desarrollo, numerosas características fueron implementadas para mejorar el rendimiento. Sin embargo, la mayoría de estos avances no estaban disponibles para la comunidad de investigadores. Mientras tanto, el modelo general del WRF fue evolucionando con las contribuciones de la comunidad de investigadores, pero estas contribuciones no tenían un camino claro para beneficiar las operaciones de NCEP.

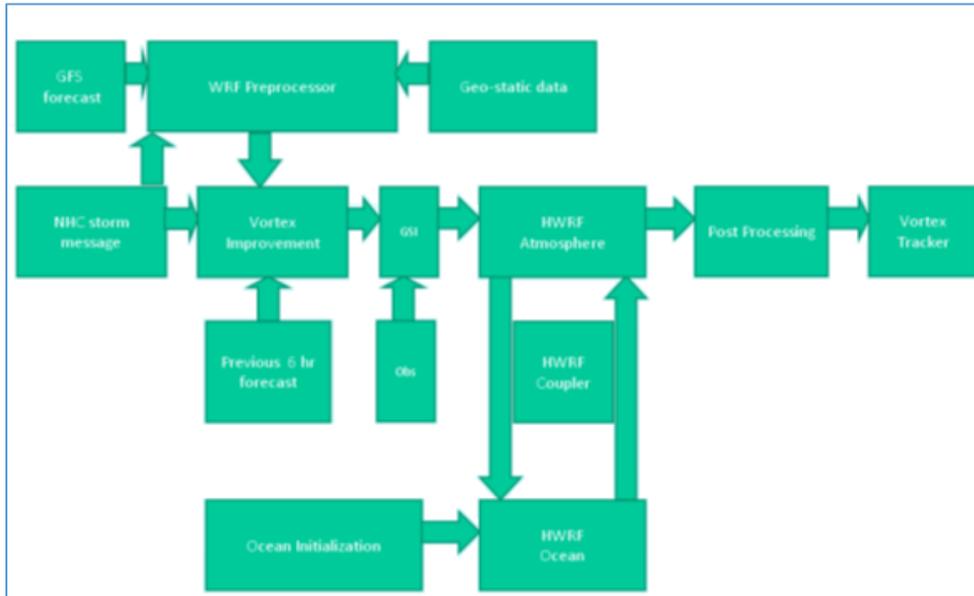


Figura 1.1. Diagrama esquemático de los componentes del sistema HWRP de la comunidad.

El DTC, en colaboración con la URI and NOAA NCEP/EMC, GFDL, y Atlantic Oceanographic and Meteorological Laboratory (AOML), portar al modelo operativo HWRP de las capacidades del WRF y WPS del código del repositorio de la comunidad. Y estableció un nuevo repositorio de código de la comunidad para inicialización del vórtice, vortex tracker, POM-TC y el acoplador basado sobre el código operacional, estableciendo efectivamente un código único para que todos los componentes del HWRP puedan ser compartidas entre las comunidades de investigación.

Una parte importante de la transición de los códigos HWRP a la comunidad fue la generalización de Los componentes para que pudieran ejecutarse en plataformas Linux además de la plataforma operacional, que en su momento era una IBM.

Una versión beta del HWRP de la comunidad, utilizando la versión WRF v3.1.1, fue lanzada en marzo de 2010, que contenía la capacidad del modelo operativo 2009.

Un plan de gestión del código para el modelo HWRP ha sido redactado por DTC y EMC. De acuerdo con este plan, el repositorio de códigos comunitarios se utilizará para el desarrollo y apoyo del sistema HWRP. HWRP estará totalmente integrado con los códigos de la comunidad, y su componente atmosférico Será una de las muchas configuraciones posibles del modelo WRF.

1.2. PRUEBAS DE FUNCIONALIDAD DEL MODELO HWRF

DTC lleva a cabo varios tipos de pruebas del HWRF. Una clase de pruebas tiene como objetivo asegurar la integridad del código de la comunidad. Dado que el componente atmosférico de HWRF es uno de los muchos posibles configuraciones del modelo WRF, es importante comprobar que la respuesta HWRF permanece sin cambios cuando WRF se actualiza en áreas no relacionadas con la configuración HWRF (por ejemplo, química). DTC también prueba el HWRF para asegurarse de que cualquier nuevo desarrollo hecho en la comunidad, laboratorios de investigación o en EMC han sido adecuadamente soportados. Otra clase de pruebas tiene como objetivo el benchmarking del código de la comunidad para Crear configuraciones de referencia de DTC (Wolff et al., 2010). Finalmente, DTC prueba el HWRF para evaluar nuevos desarrollos Implementación operativa.

La última actividad de pruebas de huracanes en el DTC tenía como objetivo asegurarse de que las capacidades de la Línea de Base Operativa HWRF de 2011 habían sido migradas correctamente al código de la comunidad antes del lanzamiento planeado de julio de 2011 y para comparar ese código comunitario para crear un DTC RC. El Programa Operacional HWRF 2011 es una configuración HWRF que utiliza WRF v3.2 con correcciones de errores adicionales y desarrollos utilizados para las pruebas iniciales hacia el modelo operativo 2011.

Es importante entender que no se espera una coincidencia exacta entre HNR2 y HR20 debido a pequeñas diferencias entre las dos configuraciones. En particular, HNR2 se ejecutó en el clúster HFIP Linux utilizando el formato NetCDF E/S, mientras que HR20 se ejecutó en el NCEP IBM utilizando formato binario de E/S binario. Por lo tanto, sólo se busca una coincidencia de los resultados estadísticos globales.

Las Figuras 1.2 y 1.3 muestran el error medio de intensidad de pista y de intensidad absoluta, respectivamente, con intervalos de confianza del 95%, para la cuenca del Atlántico Norte para las agregaciones de todos los casos. Los errores de la pista aumentan linealmente con el tiempo de pronóstico y las predicciones HNR2 y HR20 son estadísticamente indistinguibles. Los errores de intensidad absoluta crecen rápidamente en las primeras 6 h de pronóstico, seguidos por un período de crecimiento lineal en 48 h. En los últimos tres días de pronóstico los errores permanecen relativamente constante. No se encontraron diferencias estadísticamente significativas entre las intensidades absolutas medias de HNR2 y HR20 más allá del tiempo inicial, en Que las velocidades máximas del viento difieren en 2 kt. La razón de esta discrepancia de inicialización está siendo investigada.

Esta prueba demostró que el DTC funcionalmente similar de pruebas y la suite de evaluación se han establecido y que la versión base operativa del 2011 fue portado con éxito al código de la comunidad. La ejecución HNR2 se utilizará para designar un DTC RC.

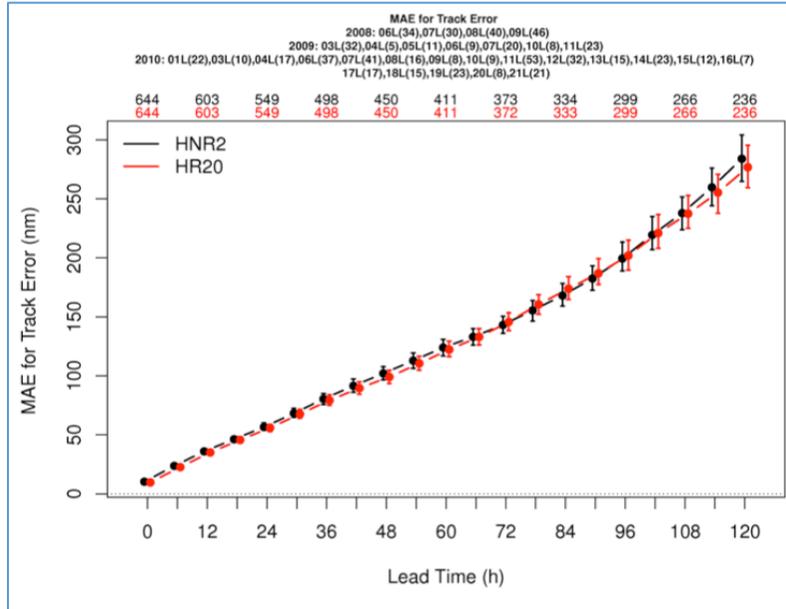


Figura 1.2. Intervalos de confianza media y 95% para los errores de pista (nm) en función del tiempo de pronóstico (h) en la cuenca atlántica de las configuraciones HNR2 (negro) y HR20 (rojo) HWRF.

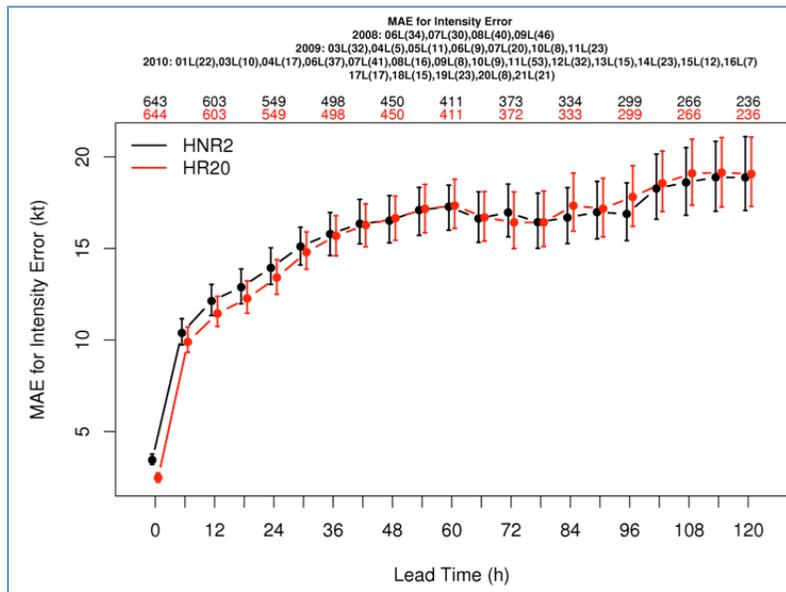


Figura 1.3. Lo mismo que la figura 1.2, excepto para la intensidad absoluta (kt).

1.3. EL MODELO HWRF EN MÉXICO

En la última década, gracias a la disponibilidad de datos provenientes de complejos modelos numéricos y avances tecnológicos para su monitoreo continuo, existen herramientas más confiables para el estudio de los ciclones tropicales así como una mejor capacidad de predecir cambios en su trayectoria e intensidad.

Como parte de las actividades de vinculación, en la unidad foránea del CICESE en La Paz se realizan actividades de observación y análisis de datos meteorológicos que incluyen imágenes de alta resolución de satélites geoestacionarios así como elaboración de los pronósticos a través de modelos numéricos que se aplican en equipo de supercómputo del CICESE.

En los últimos 66 años, más de 40 ciclones tropicales han tocado tierra en el sur de la península de Baja California, principalmente en septiembre, por lo que representan un importante grupo de sistemas que se deben de analizar. El Dr. Luis Manuel Farfán Molina, investigador titular del CICESE en La Paz, por más de una década se ha dedicado al estudio de fenómenos meteorológicos que ocurren sobre la península de Baja California, Golfo de California y océano Pacífico.

Se proporciona apoyo a tomadores de decisiones, grupos de respuesta y personal de protección civil para entender mejor las observaciones de la estructura y estimación de cambios de intensidad, señaló el científico. Esto incluye explicar los beneficios, limitaciones y precauciones a seguir con la información en los pronósticos oficiales, así como revisar las alternativas derivadas de modelos regionales y globales.

En el CICESE se realiza el análisis de observaciones de superficie, altura y satélite geoestacionario GOES. Las imágenes de los satélites se reciben cada 15 minutos en equipo del Departamento de Cómputo en Ensenada, y son útiles para determinar la posición, estructura y movimiento de sistemas atmosféricos que pueden afectar al continente o zonas marítimas.

La aplicación de información del modelo HWRF (Hurricane Weather Research and Forecasting) para estudio de las trayectorias de ciclones inició esta temporada de 2014. Se aplica una vez por día, con el fin de crear productos gráficos que muestren la distribución espacial de campos de viento y la evolución de la precipitación sobre la parte sur de la península. Ello ayuda a estudiar fenómenos como el reciente huracán “Odile”, que inició

como una depresión tropical 15-E el 10 de septiembre pasado y en menos de cinco días había alcanzado la categoría 4 como huracán.

En general, la predicción de trayectorias de ciclones tropicales es buena, aunque requiere de revisar muchas fuentes de datos. Los modelos numéricos son muy útiles y una parte fundamental en la elaboración de los pronósticos de 1 a 5 días. Hay limitaciones y deficiencias, pero es posible utilizar las predicciones para identificar zonas en las que sería necesario evacuar a la población, aunque esto es algo en la que otras ramas como la geología e hidrología son incorporadas.

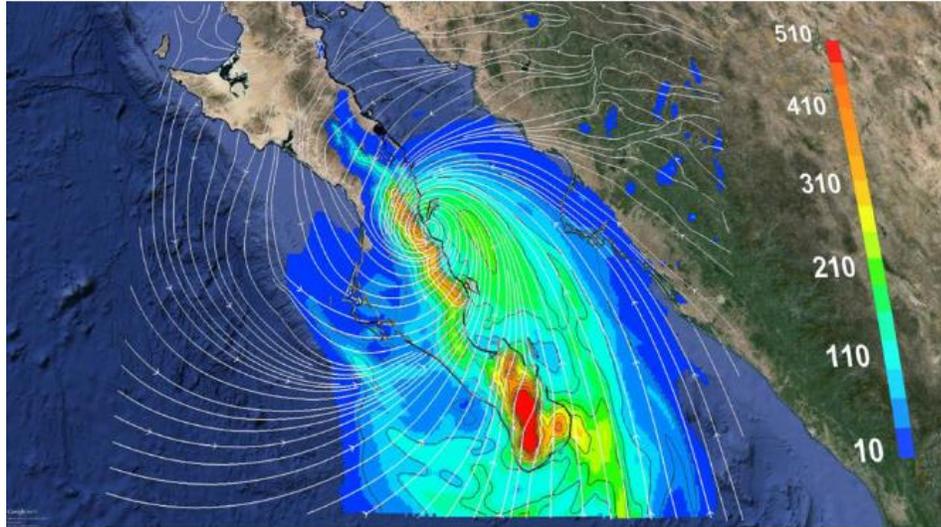
Los productos gráficos del CICESE, sobre ciclones tropicales, pueden consultarse en el sitio <http://met-hwrf.cicese.mx>.

A pesar de los avances tecnológicos y de los modelos numéricos, estos no son perfectos por lo que las investigaciones en este tema no se detienen y la mayoría están enfocadas en mejorar los modelos de predicción existentes para hacerlos más precisos pues todos tienen algún grado de incertidumbre y error.



Figura 1.4. Ojo del huracán Odile el 15 de septiembre de 2014.
(Fuente <http://www.excelsior.com.mx/media/pictures/2014/09/15/1043103.jpg>).

HWRf (CICESE) model initialized at 00 UTC, September 15, 2014
 3-km grid, total rainfall (mm) from 00 through 18 hours
 White line are streamlines at the 10-m level



- 950 mm maximum (red) over southern mountains at elevations above 1,000 meters
- San José del Cabo: HWRf=325, observation=265; Cabo San Lucas: HWRf=190, obs=N/A
- La Paz: HWRf=225, obs=108; Ciudad Constitución: HWRf=45, obs=85.

Figura 1.5. Inicialización del modelo HWRf a las 00 UTC del 15 de septiembre de 2014
 (Fuente: <http://todos.cicese.mx/imagenes/internas/p193j7848o14m4otfpr19h517t45.jpg>)

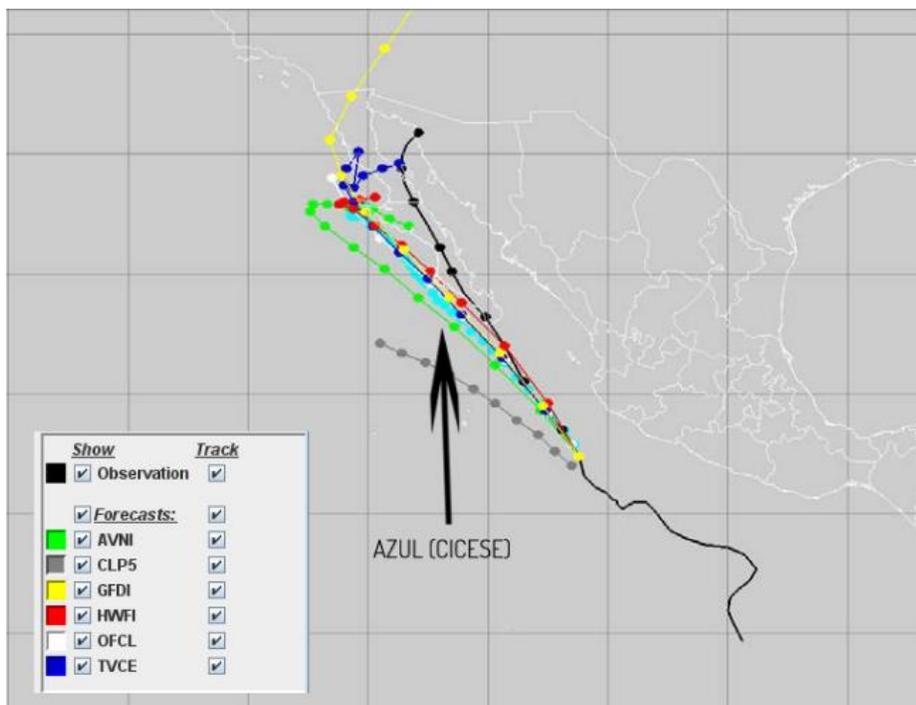


Figura 1.5. Pronóstico de trayectoria del huracán Odile emitido por el CICESE (color azul)
 (Fuente: <http://todos.cicese.mx/imagenes/internas/p193j7848o9rm8cq2qp6p11v46.jpg>)

2. DESCARGA, INSTALACIÓN, CONFIGURACIÓN Y PRUEBAS DEL MODELO HWRF

2.1. ASPECTOS GENERALES DEL MODELO HWRF

El Sistema de Previsión y Previsión del Tiempo (WRF) para huracanes (HWRF) es un modelo operativo implementado en los Centros Nacionales de Predicción Ambiental (NCEP) del Servicio Meteorológico Nacional (NWS) para proporcionar orientación numérica al Centro Nacional de Huracanes de la trayectoria, intensidad y estructura de los ciclones tropicales. El presente capítulo está basado en la versión v3.5b.

El modelo HWRF es una ecuación primitiva no hidrostática acoplado al modelo océano-atmósfera con el componente atmosférico formulado con 42 niveles en la vertical. El modelo utiliza el núcleo dinámico del modelo WRF (WRF-NMM) del modelo NMR (Non-hydrostatic Mesoscale Model), con un dominio padre y dos dominios de anidados. El dominio padre cubre aproximadamente $80^\circ \times 80^\circ$ en una rejilla E-escalonada rotacional de latitud/longitud. La ubicación del dominio padre se determina en base a la posición inicial de la tormenta y en el pronóstico del Centro Nacional de Huracanes (NHC) de la posición de 72 h, si está disponible. El dominio del anidado medio, de alrededor de $11^\circ \times 10^\circ$, y el dominio del dominio interior, de aproximadamente $7.2^\circ \times 6.5^\circ$, se mueven junto con la tormenta utilizando anidamiento interactivo bidireccional. El dominio matriz estacionario tiene un espaciamiento de la malla de 0.18° (aproximadamente 27 km), mientras que el espaciamiento del dominio medio es de 0.06° (unos 9 km) y el espaciamiento del dominio interno es de 0.02° (aproximadamente 3 km). Los pasos de tiempo son de 45, 15 y 5 segundos, respectivamente, para los dominios padre, anidado central y anidado interno.

La física del modelo se originó principalmente del modelo de huracán del Laboratorio de Dinámica de Fluidos Geofísicos (GFDL), e incluye un esquema simplificado de Arakawa-Schubert para la parametrización del cúmulo y un paquete de microfísica de la nube de Ferrier para la física húmeda explícita. El esquema de difusión vertical se basa en el esquema no-local de Troen y Mahrt. El esquema de Monin-Obukhov se utiliza para los cálculos del flujo superficial, que también emplean una parametrización mejorada del flujo momentum aire-mar en condiciones de fuerte viento, y un modelo de suelo de una capa de losa. Los efectos de la radiación se evalúan mediante el esquema GFDL, que incluye las variaciones diurnas y los efectos interactivos de las nubes. La física HWRF incluye la parametrización del calentamiento disipativo.

La predicción de 6 horas del Sistema Global de Asimilación de Datos (GDAS) del NCEP, inicializada 6 horas antes de una inicialización HWRF dada, se utiliza para generar las condiciones iniciales para el modelo de huracán en la configuración operativa. Cuando no se dispone de datos de GDAS, se puede utilizar el análisis del Sistema de predicción global del NCEP (GFS). La primera suposición se modifica utilizando el sistema de asimilación de datos HWRF (HDAS) mediante la ingesta de observaciones en un sistema de asimilación híbrido conjunto-variacional de datos llamado Interpolación Estadística Gridpoint (GSI). La información del conjunto se obtiene del Global Ensemble Forecast System (GEFS). HWRF tiene la capacidad de asimilar las observaciones del núcleo interno del huracán, como el Doppler Radar (TDR) de la aeronave P3 de NOAA. Cuando se asimilan los datos del TDR, se utiliza First Guess en el tiempo apropiado (FGAT) para asegurarse de que GSI utiliza innovaciones calculadas comparando las observaciones con los correspondientes campos de análisis del modelo válidos en el momento en que se recogió la observación. Los campos pronosticados de GFS cada 6 horas se utilizan para proporcionar condiciones de contorno lateral durante cada pronóstico.

El análisis también se modifica eliminando el vórtice presente en los primeros campos de suposición e insertando un nuevo vórtice. Dependiendo de la intensidad observada de la tormenta y en la carrera de frenado o ciclo, el nuevo vórtice puede derivarse de un procedimiento falso, a partir de la predicción de 6 horas del modelo HWRF inicializado 6 horas antes o de HDAS. En cualquier caso, el vórtice se modifica para que la posición inicial de la tormenta, la estructura y la intensidad se ajusten al mensaje de la tormenta NHC.

La integración temporal se realiza con un esquema de retroceso hacia atrás para ondas rápidas, un esquema implícito para propagación vertical de ondas sonoras y el esquema de Adams-Bashforth para la advección horizontal y para la fuerza de Coriolis. En la vertical, se usa la coordenada híbrida de presión-sigma (Arakawa y Lamb 1977). Difusión horizontal basada en un tipo de Smagorinsky de segundo orden (Janjic 1990).

La versión de la comunidad del HWRF puede utilizarse para las siguientes cuencas: Atlántico Norte, Pacífico Oriental del Pacífico, Pacífico Central, Pacífico Occidental e India (incluido el Mar Árabe y la Bahía de Bengala). En las cuencas del Atlántico Norte y del Pacífico Norte Oriental, de las cuales NHC es responsable, el modelo atmosférico se combina con el Modelo Oceanográfico de Princeton (POM) para Ciclones Tropicales (POM-TC). El POM se desarrolló en la Universidad de Princeton. En la Universidad de Rhode Island (URI), el POM se acopló a los modelos GFDL y WRF. En el Pacífico Norte Oriental, se emplea una configuración unidimensional (columna) del POM-TC, mientras que en la cuenca atlántica, el POM-TC se

ejecuta en tres dimensiones. En ambas cuencas el espaciamiento horizontal de la rejilla es de $1/6^\circ$ (aproximadamente 18 km). En el Atlántico, el POM-TC está configurado con 23 niveles verticales, mientras que 16 niveles se utilizan en el Pacífico norte oriental. En las otras cuencas, HWRF está configurado para funcionar solamente con su componente atmosférico.

El POM-TC se inicializa mediante un giro diagnóstico y pronóstico de las circulaciones oceánicas utilizando datos oceanográficos climatológicos. Para las tormentas situadas en la parte occidental de la cuenca del Atlántico, las condiciones iniciales se mejoran con la temperatura de la superficie del mar en tiempo real (SST), los datos de altura de la superficie del mar y la asimilación de "características" oceánicas. Durante la subida del océano, se incorporan representaciones realistas de la estructura y posiciones de la Corriente de Bucle, Corriente del Golfo y remolinos de núcleo caliente y frío usando una técnica de asimilación de datos basada en características desarrollada en URI.

HWRF es adecuado para su uso en aplicaciones tropicales, incluyendo NWP en tiempo real, investigación de pronósticos, investigación de parametrización física, investigación y enseñanza de acoplamiento aire-mar. Además, se ha agregado una capacidad idealizada de simulación de ciclones tropicales a HWRF V3.5a. El apoyo del sistema HWRF a la comunidad por el Developmental Testbed Centre (DTC) incluye los siguientes tres módulos principales.

- HWRF atmospheric components
 - WRF-NMM (which has tropical physics schemes and a vortex- following moving nest)
 - ✓ ○ WRF Preprocessing System (WPS)
 - ✓ ○ Vortex initialization
 - ✓ ○ Gridpoint Statistical Interpolation (GSI)
 - ✓ ○ Unified Post-Processor (UPP)
 - ✓ ○ GFDL vortex tracker

- HWRF oceanic components
 - ✓ ○ POM-TC model
 - ✓ ○ Ocean initialization

- Atmosphere-Ocean Coupler

Los componentes atmosféricos y oceánicos están acoplados de forma interactiva con un acoplador basado en MPI (Message Passing Interface), desarrollado en el Centro de Modelación Ambiental (EMC) de NCEP. Los componentes atmosféricos y oceánicos

intercambian información a través del acoplador; El océano envía la temperatura superficial del mar (SST) a la atmósfera; La atmósfera recibe la SST y envía los flujos superficiales, incluyendo el flujo de calor sensible, el flujo de calor latente y la radiación de onda corta al océano, y así sucesivamente. La frecuencia de intercambio de información es de 9 minutos.

2.2. ESTRUCTURA DEL DIRECTORIO DEL CÓDIGO FUENTE

El código fuente del sistema HWRF tiene los siguientes componentes:

- WRF Atmospheric Model
- WPS
- UPP
- GSI
- HWRF Utilities
- POM-TC
- GFDL Vortex Tracker
- NCEP Atmosphere-Ocean Coupler

El código para todos los componentes se pueden obtener descargandolos en archivos con formato “tar.gz” desde la página web de DTC.

- hwrfv3.5a_utilities.tar.gz
- hwrfv3.5a_pomtc.tar.gz
- hwrfv3.5a_gfdl_vortextracker.tar.gz
- hwrfv3.5a_ncep-coupler.tar.gz
- hwrfv3.5a_wrf.tar.gz
- hwrfv3.5a_wps.tar.gz
- hwrfv3.5a_upp.tar.gz
- hwrfv3.5a_gsi.tar.gz

Después de copiar estos archivos tar.gz en un directorio superior HWRF definido por el usuario y expandirlos, el usuario debería ver los siguientes directorios.

- WRFV3. Modelo Weather Research and Forecasting (WRF)
- WPSV3. WRF Pre-Procesador
- UPP. Preprocesador Unificado.
- GSI. Asimilacion de datos con Gridpoint Statistical Interpolation 3D-VAR (GSI)
- hwrf-utilities. Inicialización del vortices, utilerías, herramientas, y librerías suplementarias.
- gfdl-vortextracker. Vortex tracker

- ncep-coupler. Acoplador océano - atmósfera.
- Pomtc. Versión de POM para ciclones tropicales.

La estructura de directorios para estos ocho componentes se enumera a continuación. Debemos tener en cuenta que estos son los directorios después de que se compila el código fuente. Antes de la compilación, no todos estos directorios están presentes.

1. hwrp-utilities (HWRP Utilities programs and scripts)

```

|__arch/          (compile options)
|__clean          script to clean created files and executables
|__compile        script to compile the HWRP-Utilities code
|__configure      script to create the configure.hwrp file for compile
|__exec/          (executables)
|__libs/          (libraries including blas, sp, sfcio, bacio, w3 and bufr)
|__makefile       top level makefile
|__parm/          (various namelists including those for WPS, WRF, GSI, and UPP;
|                 relevant WRF lookup tables from run subdirectory, some of which are
|                 modified for HWRP)
|__scripts/      (scripts used to run the HWRP system)
|                 |__funcs (shell functions used by the scripts)
|__tools/         (source code for tools to run the HWRP system)
|                 Makefile makefile for tools code
|                 |__grbindex
|                 |__hwrp_data_flag
|                 |__hwrp_prep_hybrid
|                 |__hwrp_readtdrstmid
|                 |__hwrp_readtdrtime
|                 |__hwrp_wrfout_newtime
|                 |__wgrib
|__vortex_init/   (source code for the HWRP vortex initialization)
|                 Makefile makefile for vortex_init code
|                 |__hwrp_anl_bogus
|                 |__hwrp_anl_cs
|                 |__hwrp_anl_step2
|                 |__hwrp_create_nest
|                 |__hwrp_create_trak_fnl
|                 |__hwrp_create_trak_guess
|                 |__hwrp_diffwrf_3dvar
|                 |__hwrp_guess
|                 |__hwrp_pert_ct
|                 |__hwrp_set_ijstart
|                 |__hwrp_split
|                 |__interpolate
|                 |__merge_nest
|__wrapper_scripts/ (top-level wrapper scripts to run the HWRP system)

```

2. pomtc (POM-TC Ocean model)

__arch/	(compile options)
__clean	script to clean created files and executables
__compile	script to compile the POM-TC code
__configure	script to create the configure.pom file for compile
__fix	fixed data files needed to run POM-TC
__makefile	top level makefile
__ocean_exec/	(ocean model executables)
__ocean_init/	(source code for generating ocean model initial condition)
	<i>Makefile</i> makefile for the ocean initialization code
	__date2day
	__day2date
	__eastatl
	__eastpac
	__ext_eastat
	__getsst
	__gfdl_find_region
	__sharp_mcs_rf_l2m_rmy5
	__united
__ocean_main/	(source code for the ocean forecast model)
	<i>Makefile</i> makefile for the ocean model code
	__ocean_eastatl
	__ocean_eastatl_ext
	__ocean_eastpac
	__ocean_united
__ocean_parm/	(namelists for ocean model)
__ocean_plot/	(sample GrADS scripts used to plot ocean output)

3. ncep-coupler (NCEP Coupler)

__arch/	(compile options)
__clean	script to clean created files and executables
__compile	script to compile the coupler code
__configure	script to create the configure.cpl file for compile
__cpl_exec/	(coupler executables)
__hwrj_wm3c/	(source code)
__makefile	top level makefile

4. gfdl-vortextracker (GFDL Vortex Tracker)

__arch/	(compile options)
__clean	script to clean created files and executables
__compile	script to compile the tracker code
__configure	script to create the configure.trk file for compile
__makefile	top level makefile
__trk_exec/	(GFDL vortex tracker executables)
__trk_plot/	(GFDL vortex tracker plot scripts and data)
__trk_src /	(GFDL vortex tracker source codes)

5. WRFV3 (Atmospheric model)

<code>__Makefile</code>	makefile used to compile WRFV3
<code>__Registry/</code>	(WRFV3 Registry files)
<code>__arch/</code>	(compile options)
<code>__chem/</code>	(WRF-Chem, not used in HWRF)
<code>__clean</code>	script to clean created files and executables
<code>__compile</code>	script to compile the WRF code
<code>__configure</code>	script to create the <code>configure.wrf</code> file for compile
<code>__dyn_em/</code>	(Advanced Research WRF dynamic modules, not used by HWRF)
<code>__dyn_exp/</code>	('toy' dynamic core, not used by HWRF)
<code>__dyn_nmm/</code>	(WRF-NMM dynamic modules, used by HWRF)
<code>__external/</code>	(external packages including ocean coupler interface)
<code>__frame/</code>	(modules for WRF framework)
<code>__hydro/</code>	(hydrology module, not used by HWRF)
<code>__inc/</code>	(include files)
<code>__main/</code>	(WRF main routines, such as <code>wrf.F</code>)
<code>__phys/</code>	(physics modules)
<code>__run/</code>	(run directory, not used by HWRF)
<code>__share/</code>	(modules for WRF mediation layer and WRF I/O)
<code>__test/</code>	(sub-dirs for specific configurations of WRF, such as idealized HWRF)
<code>__tools/</code>	(tools directory)
<code>__var/</code>	(WRF-Var)

6. WPSV3 (WRF Pre-processor)

<code>__arch/</code>	(compile options)
<code>__clean</code>	script to clean created files and executables
<code>__compile</code>	script to compile the WPS code
<code>__configure</code>	script to create the <code>configure.wps</code> file for compile
<code>__geogrid/</code>	(source code for <code>geogrid.exe</code>)
<code>__link_grib.csh</code>	script used by <code>ungrib</code> to link input GRIB files, used in idealized simulations
<code>__metgrid/</code>	(source code for <code>metgrid.exe</code>)
<code>__test_suite/</code>	(WPS test cases)
<code>__ungrib/</code>	(source code for <code>ungrib.exe</code>)
<code>__util/</code>	(utility programs for WPSV3)

7. UPP (Unified Post-Processor)

<code>__arch/</code>	(compile options)
<code>__bin/</code>	(executables)
<code>__clean</code>	script to clean created files and executables
<code>__compile</code>	script to compile the UPP code
<code>__configure</code>	script to create the <code>configure.upp</code> file for compile
<code>__include/</code>	(include files)
<code>__lib/</code>	(libraries)
<code>__makefile</code>	makefile used to build UPP code
<code>__parm/</code>	(parameter files to control UPP performed, not used by HWRF)
<code>__scripts/</code>	(sample scripts running UPP, not used by HWRF)
<code>__src/</code>	(UPP source codes)

8. GSI (Gridpoint Statistical Interpolation)

<code>__arch/</code>	(compile options)
<code>__clean</code>	script to clean created files and executables
<code>__compile</code>	script to compile the GSI code
<code>__configure</code>	script to create the <code>configure.gsi</code> file for compile
<code>__fix/</code>	(fix files for GSI)
<code>__include/</code>	(include files)
<code>__lib/</code>	(libraries)
<code>__makefile</code>	makefile used to build GSI code
<code>__run/</code>	(executables)
<code>__scripts/</code>	(sample scripts running GSI, not used by HWRF)
<code>__src/</code>	(GSI source codes)
<code>__util/</code>	(GSI utilities, not used by HWRF)

2.3. ESTRUCTURA DEL DIRECTORIO DE DATOS DE ENTRADA

Los usuarios necesitarán los conjuntos de datos siguientes como entrada a los componentes del modelo HWRF. Para utilizar los scripts compatibles con DTC para ejecutar HWRF, estos conjuntos de datos deben almacenarse siguiendo la estructura de directorios de abajo en un disco accesible por los scripts y ejecutables HWRF.

1. *Tcvitals* (TC Vitals data)

`syndat_tcvitals.$YYYY`

2. *abdecks* (a-deck and b-deck files)

`a[al|ep]$MM}$YYYY}.dat` A deck file.

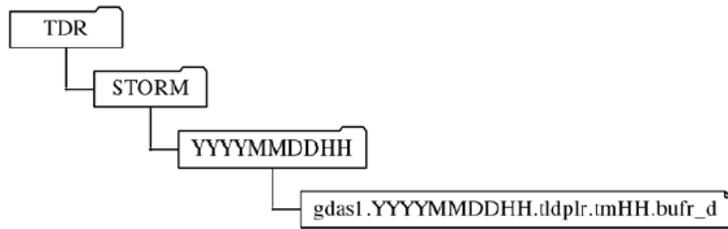
`b[al|ep]$MM}$YYYY}.dat` B deck file.

6. GEFS (input data from GEFS)

- *spectral (GEFS spectral data to provide ensemble information for data assimilation)*



7. TDR (tail doppler radar)



8. Fix (time-independent files)

8.1 *upp* (for UPP)

8.2 *gsi* (for GSI)

CRTM_Coefficients

anavinfo_hwrf
atms_beamwidth.txt
bufitab.012
global_ozinfo.txt
global_scaninfo.txt
hwrf_basinscale_satinfo.txt
hwrf_convinfo.txt
hwrf_hybens_d01_locinfo
nam_errtable.r3dv
nam_glb_berror.f77.gcv
nam_glb_berror.f77.gcv_Little_Endian
nam_global_ozinfo.txt
nam_global_pcpinfo.txt
nam_global_satangbias.txt
nam_regional_convinfo.txt
nam_regional_satinfo.txt
prepobs_errtable.hwrf
prepobs_prep.bufitable

8.3 *ocean* (for ocean initialization)

gfdl_ocean_topo_and_mask.eastatl
gfdl_ocean_topo_and_mask.eastatl_extn
gfdl_Hdeepgsu.eastatl
gfdl_gdem.[00-13].ascii
gfdl_initdata.eastatl.[01-12]
gfdl_initdata.gdem.united.[01-12]
gfdl_initdata.united.[01-12]
gfdl_ocean_readu.dat.[01-12]
gfdl_ocean_spinup_gdem3.dat.[01-12]
gfdl_ocean_spinup_gspath.[01-12]
gfdl_ocean_spinup.BAYuf
gfdl_ocean_spinup.FSGsuf
gfdl_ocean_spinup.SGYREuf
gfdl_ocean_topo_and_mask.united
gfdl_pctwat
gfdl_raw_temp_salin.eastpac.[01-12]

2.4. ESTRUCTURA DEL DIRECTORIO DE PRODUCCIÓN

Al utilizar los scripts incluidos en los archivos tar publicados para ejecutar el sistema HWRF, se crearán y rellenarán los siguientes directorios de producción:

El directorio superior de producción es $\${HWRF_OUTPUT_DIR}/\${SID}/\${YYYYMMDDHH}$, donde $\${SID}$ es el identificador de la tormenta, ejemplo 09L, y $\${YYYYMMDDHH}$ es el tiempo inicial del pronóstico.

The directory $\${HWRF_OUTPUT_DIR}/\${SID}/\${YYYYMMDDHH}$ podrá tener los siguientes subdirectorios:

messages (created by *hwrfdomain.ksh*)
geoprd (created by *geogrid.ksh*)
ungribprd (created by *ungrib.ksh*)
metgridprd (created by *metgrid.ksh*)
prep_hybrid (created by *prep_hybrid.sh*)
prep_hybrid_GFS (created by *prep_hybrid.sh*)
realprd (created by *real.ksh*)
realprd_GFS (created by *real.ksh*)
wrfghostprd (created by *wrf.ksh* run in “ghost” mode)
wrfanalysisprd (created by *wrf.ksh* run in “analysis” mode)
trkanalysisprd (created by *track_analysis.ksh*)
relocateprd (created by the vortex initialization scripts)
gsiprd (created by the GSI scripts, if GSI is run)
mergeprd (created by *merge.ksh*)
oceanprd (created by the ocean initialization scripts)

- *sharpn* (created by the sharpening program, present only for the “united” ocean domain)
- *getsst* (created by the procedure to extract the SST from GFS)
- *phase3* (created by the 48-hr spin-up procedure to generate geostrophically-balanced currents)
- *phase4* (created by the 72-hr spin-up procedure using the wind stress extracted from the NHC hurricane message file)

wrfprd (created by *wrf.ksh* in “main” mode)
postprd (created by *run_unipost*)
gvtpd (created by *tracker.ksh*)

2.5. COMPONENTES DEL MODELO HWRF

El sistema HWRF de la comunidad DTC, en el cual está basado el modelo HWRF operativo de la NOAA, consiste de ocho componentes:

- WRF Atmospheric Model
- WPS
- UPP
- GSI
- HWRF Utilities
- POM-TC
- GFDL Vortex Tracker
- NCEP Atmosphere-Ocean Coupler

Cada uno de estos componentes está disponible en el DTC como software de la comunidad. Los tres primeros de estos componentes son los componentes WRF tradicionales: WRF, WPS y UPP. GSI es un código de asimilación de datos variacional en 3D utilizado para la asimilación de datos, y los cuatro componentes restantes son específicos del propio sistema de huracanes, y como tales se denominan los componentes de huracán del sistema HWRF.

2.6. OBTENCIÓN DEL CÓDIGO FUENTE

Todos los componentes del modelo HWRF están disponibles en el sitio web de HWRF. Si bien la mayoría de estos códigos también están disponibles en otros sitios web de la comunidad, las versiones necesarias para HWRF deben adquirirse en el sitio web de DTC HWRF para asegurarse de que son un conjunto coherente. Todos los componentes de HWRF se pueden obtener a través del sitio web de HWRF

<http://www.dtcenter.org/HurrWRF/users>

Seleccionando las fichas descargar y Sistema HWRF en el menú vertical izquierdo. Los nuevos usuarios deben primero registrarse antes de descargar el código fuente. Los usuarios que regresan sólo necesitan proporcionar su dirección de correo electrónico de registro. Una descarga exitosa produce los ocho archivos tar.gz.

- hwrfv3.5a_hwrf-utilities.tar.gz

- hwrfv3.5a_pomtc.tar.gz
- hwrfv3.5a_gfdl-vortextracker.tar.gz
- hwrfv3.5a_ncep-coupler.tar.gz
- hwrfv3.5a_WRFV3.tar.gz
- hwrfv3.5a_WPSV3.tar.gz
- hwrfv3.5a_UPP.tar.gz
- hwrfv3.5a_GSI.tar.gz

Después de descargar cada uno de los códigos de componente, el usuario debe comprobar los enlaces a problemas conocidos y correcciones de errores para ver si se requieren actualizaciones de código. Ahora tiene todos los componentes del sistema HWRF como archivos .tar.gz.

2.7. DEFINICIÓN DEL DIRECTORIO PRINCIPAL DEL HWRF

Los scripts de ejecución del modelo HWRF proporcionados por el DTC son razonablemente flexibles y con un esfuerzo mínimo pueden soportar casi cualquier diseño. Por simplicidad, se supone que el sistema HWRF se establecerá en una única estructura de directorio plana. Debido a los requisitos de almacenamiento necesarios para la configuración completa del sistema HWRF, normalmente tendrá que estar ubicado en la partición "scratch" o "work" de una computadora. Antes de desempaquetar los archivos tar.gz que acaba de descargar, cree un solo directorio de trabajo en su área de trabajo. Mueva los archivos tar en él y descomprímalos allí.

Los archivos tar.gz se pueden desempaquetar mediante el comando GNU gunzip,

```
gunzip * .tar.gz
```

Y los archivos tar extraídos ejecutando tar -xvf individualmente en cada uno de los archivos tar.

Una vez desempaquetado, debe haber los ocho directorios de origen.

- WRFV3
- WPSV3
- UPP

- GSI
- hwrp-utilities
- gfdl-vortextracker
- ncep-coupler
- pomtc

Un noveno directorio para la salida también se puede crear aquí también.

```
mkdir resultados
```

El usuario debe asegurarse de que el directorio de salida creado aquí es coherente con la variable de entorno HWRP_OUTPUT_DIR definida en el programa

```
hwrp-utilities/wrapper_scripts/global_vars.ksh
```

2.8. REQUERIMIENTOS DEL SISTEMA, LIBRERÍAS Y HERRAMIENTAS

En términos prácticos, el sistema HWRP consiste en una colección de scripts de shell, que ejecutan una secuencia de ejecutables de código serie y paralelo. El código fuente de estos ejecutables está en forma de programas escritos en FORTRAN, FORTRAN 90 y C. Además, los ejecutables paralelos requieren alguna variante de MPI/OpenMP para el paralelismo de la memoria distribuida, y la E/S depende de la librería de E/S netCDF. Más allá de los scripts de shell estándar, el sistema de compilación se basa en el uso del lenguaje de secuencias de programas en lenguaje Perl y los comandos make y date.

Los requisitos básicos para construir y ejecutar el sistema HWRP se enumeran a continuación.

- FORTRAN 90+ compiler
- C compiler
- MPI v1.2+
- Perl
- netCDF V3.6+
- LAPACK and BLAS
- GRIB1/2

Debido a que estas herramientas y librerías suelen ser el ámbito de los administradores de sistemas para instalar y mantener, se agrupan aquí como parte de los requisitos básicos del sistema.

2.9. LIBRERÍAS INCLUIDAS

Para mayor comodidad en la construcción de componentes HWRF-Utilities, POM-TC y GFDL Vortex Tracker, el componente HWRF-Utilities incluye varias bibliotecas en el directorio `hwrf-utilities/libs/src/`. Estas bibliotecas se crean automáticamente cuando se construye el componente HWRF-Utilities. Las bibliotecas incluidas se enumeran a continuación.

- BACIO
- BLAS
- BUFR
- SFCIO
- SIGIO
- SP
- W3

Los otros componentes, WPS, WRF, UPP y GSI, vienen con sus propias versiones de muchas de estas bibliotecas, pero normalmente se han personalizado para ese componente en particular y no deben ser utilizados por los otros componentes.

Cuando se compila el componente HWRF-Utilities, comienza construyendo primero todas las bibliotecas incluidas. El código de inicialización de vórtice contenido en el componente HWRF-Utilities requiere todas las bibliotecas anteriores, excepto la biblioteca SFCIO. Además, requiere las bibliotecas matemáticas BLAS y LAPACK cuando la biblioteca IBM ESSL no se incluye con la instalación del compilador.

El componente POMTC requiere las bibliotecas SFCIO, SP y W3. Además, la copia local de la biblioteca BLAS es necesaria cuando la biblioteca ESSL no se incluye con la instalación del compilador. Esto se debe a que las versiones suministradas por el vendedor de BLAS son típicamente incompletas, y la versión local complementa la versión del vendedor. Normalmente esto es para cualquier sistema que no sea IBM. El componente de seguimiento de vórtice GFDL requiere las bibliotecas BACIO y W3. El NCEP-Coupler no requiere bibliotecas adicionales.

Los ocho componentes del sistema HWRF tienen ciertas interdependencias. Muchos de los componentes dependen de bibliotecas producidas por otros componentes. Por ejemplo, cuatro de los componentes, WPS, UPP, GSI y HWRF-Utilities, requieren vincular a las bibliotecas de API de E / S de WRF para generar. Dado que estas bibliotecas de E / S se crean como parte de la compilación WRF, primero debe construirse el componente WRF. Una vez que WRF se construye, WPS, UPP, GSI, o el HWRF-Utilidades se pueden construir en cualquier orden. Dado que la construcción de HWRF-Utilities produce las bibliotecas suplementarias necesarias por POM-TC y por el GFDL Vortex Tracker, las utilidades HWRF deben ser construidas antes de cualquiera de estos componentes. El componente restante, el acoplador NCEP, se puede construir independientemente de cualquiera de los otros componentes.

La dependencia de los componentes es la siguiente.

- WRF
 - ✓ WPS
 - ✓ UPP
 - ✓ GSI
 - ✓ HWRF Utilities
 - POM-TC (BLAS sobre Linux, sfcio, sp, w3)
 - GFDL vortex tracker (w3 & bacio)
- NCEP Coupler

2.10. COMPILACIÓN DE LOS MODULOS DEL HWRF

El código WRF tiene un mecanismo de construcción bastante sofisticado. El paquete intenta determinar la máquina donde se está construyendo el código y, a continuación, presenta al usuario con opciones de generación compatibles en esa plataforma. Por ejemplo, en una máquina Linux, el mecanismo de compilación determina si la máquina es de 32 bits o de 64 bits, solicita al usuario el tipo de paralelismo deseado (como serial, memoria compartida, memoria distribuida o híbrido) y, a continuación, Presenta una selección de posibles opciones de compilador.

Además, a partir de la versión actual de HWRF, el usuario puede optar por ejecutar WRF con datos de entrada reales o idealizados. El caso de datos idealizado requiere establecer

banderas de entorno antes de compilar el código, lo que crea un ejecutable único que sólo se debe ejecutar con los datos idealizados.

2.10.1. COMPILANDO WRF-NMM

Para configurar correctamente WRF-NMM para el sistema HWRF, se deben establecer las siguientes variables de entorno adicionales más allá de lo que normalmente requiere WRF, asumiendo que se utiliza bourne shell:

```
export HWRF=1
export WRF_NMM_CORE=1
export WRF_NMM_NEST=1
export WRFIO_NCD_LARGE_FILE_SUPPORT=1
```

Estos ajustes producen una versión de WRF-NMM compatible con el sistema HWRF. Debemos tener en cuenta que establecer la variable de entorno WRF_NMM_NEST en 1 no impide que se ejecute con un solo dominio.

Para configurar WRF-NMM nos vamos al directorio superior del módulo WRF y ejecutamos

```
./configure
```

Se presentará una lista de opciones de construcción para la computadora. Estas opciones pueden incluir múltiples compiladores y opciones de paralelismo. Para el sistema HWRF, sólo se recomienda la compilación de memoria distribuida (dmpar). Para nuestro caso seleccionamos la opción 3

```
3. (dmpar) PGI (pgf90/gcc)
```

Al completar la configuración del modelo WRF, se crea el archivo configure.wrf con las opciones de compilación y las rutas. Estos valores se pueden editar para una personalización adicional del proceso de compilación.

Para crear el componente WRF-NMM, ejecutamos el comando:

```
./compile nmm_real
```

Generalmente es recomendable guardar la salida y error estándar en un archivo de registro para referencia, para bash utilizamos el comando

```
./compile nmm_real 2>&1 | tee build.log
```

Esto envía la salida estándar y el error estándar tanto al archivo build.log como a la pantalla.

El tiempo de compilación aproximado varía según el sistema que se está utilizando y la agresividad de la optimización. En la mayoría de los sistemas Linux, el modelo WRF normalmente compila en unos 20 minutos.

Es importante tener en cuenta que los comandos `./compile -h` y `./compile` generan una lista de todas las opciones de compilación disponibles, pero sólo la opción `nmm_real` es relevante para el sistema HWRF.

Para quitar todos los archivos objeto (excepto los que están en el directorio `external/`), escribimos:

```
./clean
```

Para llevar a cabo una limpieza completa que elimina todos los archivos incorporados en todos los directorios, así como `configure.wrf`, escriba:

```
./clean -a
```

Se recomienda una limpieza completa si la compilación falla, se ha cambiado el registro o se ha cambiado el archivo de configuración.

Una compilación exitosa produce dos ejecutables enumerados a continuación en el directorio `main/`

- `real_nmm.exe`: Inicialización del WRF
- `wrf.exe`: Integración del modelo WRF

2.10.2. COMPILANDO HWRF-UTILITIES

El directorio hwrp-utilities consta de una colección ecléctica de código fuente y bibliotecas. Las bibliotecas, que se proporcionan en apoyo del POM-TC y del GFDL Vortex Tracker, incluyen las bibliotecas BACIO, BLAS, BUFR, SIGIO, SFCIO, SP y W3. Además de estas bibliotecas, este componente incluye el código fuente para las rutinas de inicialización de vórtices y herramientas de software como el grbindex.

La compilación de utilidades HWRF requiere que las dos variables de ruta de acceso, NETCDF y WRF_DIR, se establezcan en las rutas de acceso adecuadas. La ruta NETCDF de la librería netCDF es necesaria para construir el componente WRF-NMM, y su valor debe establecerse de forma adecuada si dicho componente se compila correctamente. La variable de ruta WRF_DIR debe apuntar al directorio WRF compilado anteriormente. Primero debe crear WRF antes de compilar cualquiera de los otros componentes.

```
export NETCDF={directorio donde se instaló netCDF}
export WRF_DIR={directorio donde se instaló WRF}
```

Es fundamental que el compilador Fortran utilizado para construir las librerías (Intel, PGI, XLF, etc.) sea el mismo que el compilador utilizado para compilar el código fuente. Normalmente, esto es sólo un problema en dos situaciones: en sistemas Linux que tienen Múltiples compiladores instalados; Y en sistemas donde existe la posibilidad de elegir entre construir el código con direccionamiento de 32 bits o de 64 bits.

Para configurar HWRF-Utilities para la compilación, entramos al directorio hwrp-utilities y ejecutamos

```
./configure
```

El script de configuración comprueba el hardware del sistema y si las variables de ruta no están establecidas, pregunta por las rutas correctas a las bibliotecas netCDF y al directorio de compilación WRF. Concluye pidiendo al usuario que elija una configuración soportada por la arquitectura actual de la máquina. Seleccionamos la opción 1

```
1. Linux x86_64, PGI compiler w/LAPACK (dmpar)
```

Si tiene éxito, se generara el archivo llamado configure.hwrf en el directorio hwrf-utilities. Este archivo contiene las opciones de compilación, reglas y rutas específicas de la arquitectura de la máquina actual, y se puede editar para cambiar las opciones de compilación si es necesario.

Para compilar las utilidades HWRF y guardar la salida de generación en un archivo de registro, ejecutamos el comando

```
./compile 2>&1 | tee build.log
```

Para remover todos los archivos objeto ejecutamos

```
./clean
```

Para realizar una limpieza completa que elimine todos los archivos de compilación, incluidos los ejecutables, las librerías y el configure.hwrf, ejecutamos

```
./clean -a
```

Se recomienda una limpieza completa si la compilación falla o si se cambia el archivo de configuración.

Si la compilación es exitosa, se crearán 24 ejecutables en el directorio exec/

- diffwrf_3dvar.exe
- grbindex.exe
- hwrf_anl_4x_step2.exe
- hwrf_anl_bogus_10m.exe
- hwrf_anl_cs_10m.exe
- hwrf_bin_io.exe
- hwrf_create_nest_1x_10m.exe
- hwrf_create_trak_fnl.exe
- hwrf_create_trak_guess.exe
- hwrf_data_flag.exe
- hwrf_inter_2to1.exe
- hwrf_inter_2to2.exe
- hwrf_inter_2to6.exe

- hwrp_inter_4to2.exe
- hwrp_inter_4to6.exe
- hwrp_merge_nest_4x_step12_3n.exe
- hwrp_pert_ct1.exe
- hwrp_prep.exe
- hwrp_readtdrstmid.exe
- hwrp_readtdrtime.exe
- hwrp_split1.exe
- hwrp_swcorner_dynamic.exe
- hwrp_wrfout_newtime.exe
- wgrib.exe

Además, se crearán diez librerías en el directorio `libs/`

- `libbacio.a`
- `libblas.a`
- `libbufr_i4r4.a`
- `libbufr_i4r8.a`
- `libsfcio_i4r4.a`
- `libsigio_i4r4.a`
- `libsp_i4r8.a`
- `libsp_i4r4.a`
- `libw3_i4r8.a`
- `libw3_i4r4.a`

Estas librerías serán utilizadas por el GFDL Vortex Tracker y el modelo de océano POM-TC. El paso de configuración para estos componentes requerirá configurar una variable de ruta para que apunte al directorio `hwrp-utilities/libs/` del directorio de utilidades HWRP.

Las utilidades de HWRP se pueden compilar para producir solamente las bibliotecas escribiendo el comando siguiente

```
./compile library
```

Esto es útil para los usuarios que no tienen la intención de utilizar todo el sistema HWRP, pero sólo necesitan las bibliotecas para construir el rastreador (Vortex Tracker).

2.10.3. COMPILANDO POM-TC

La versión para ciclones tropicales del POM-TC requiere de tres librerías: SFCIO, SP y W3. En las plataformas que carecen de las bibliotecas matemáticas ESSL, normalmente cualquier otra cosa que no sean máquinas IBM AIX, se requiere una cuarta biblioteca (BLAS). Todas estas bibliotecas se encuentran en el directorio `hwrp-utilities/libs/` y deberían estar disponibles si el componente HWRP-Utilities se ha construido correctamente. Primero debe construirlos antes de construir POM-TC. Debemos configurar las librerías de la siguiente manera, considerar que `dir_hwrp` corresponde al directorio base de nuestra instalación

```
export LIB_W3_PATH=${dir_hwrp}/hwrp-utilities/libs/  
export LIB_W3_PATH=${dir_hwrp}/hwrp-utilities/libs/  
export LIB_SP_PATH=${dir_hwrp}/hwrp-utilities/libs/  
export LIB_SFCIO_PATH=${dir_hwrp}/hwrp-utilities/libs/  
export LIB_BLAS_PATH=${dir_hwrp}/hwrp-utilities/libs/
```

La configuración de POM-TC para su compilación, entramos al directorio `pomptc/` y ejecutamos

```
./configure
```

El script de configuración comprueba el hardware del sistema, y si las variables de ruta no están establecidas, pide rutas de software a las bibliotecas BLAS, W3, SP y SFCIO, y para Linux. Concluye pidiendo al usuario que elija una configuración soportada por la arquitectura actual de la máquina. Seleccionamos la opción 1

```
1. Linux x86_64, PGI compiler (dmpar)
```

Después de seleccionar la opción de compilador deseada, el script de configuración crea un archivo denominado `configure.pom`. Este archivo contiene opciones de compilación, reglas y rutas específicas de la arquitectura de la máquina actual, y se puede editar para cambiar las opciones de compilación, si se desea.

Para compilar el POM-TC y guardar la salida de generación en un archivo de registro, ejecutamos

```
./compile 2>&1 | tee ocean.log
```

Para eliminar los archivos objetos ejecutamos

```
./clean
```

Para llevar a cabo una limpieza completa que elimine todos los archivos, objetos, archivos ejecutables y el archivo de configuración configure.pom, ejecutamos

```
./clean -a
```

Se recomienda encarecidamente una limpieza completa si la compilación falla al generar o si se cambia el archivo de configuración.

Si la compilación es exitosa, se generaran 13 archivos ejecutables en el directorio ocean_exec/

- gfdl_date2day.exe
- gfdl_day2date.exe
- gfdl_find_region.exe
- gfdl_getsst.exe
- gfdl_ocean_eastatl.exe
- gfdl_ocean_eastpac.exe
- gfdl_ocean_ext_eastatl.exe
- gfdl_ocean_united.exe
- gfdl_sharp_mcs_rf_l2m_rmy5.exe
- hwrp_ocean_eastatl.exe
- hwrp_ocean_eastatl_ext.exe
- hwrp_ocean_eastpac.exe
- hwrp_ocean_united.exe

Los ejecutables hwrp_ocean_united.exe, hwrp_ocean_eastpac.exe, hwrp_ocean_eastatl.exe y hwrp_ocean_eastatl_ext.exe, son los ejecutables del modelo oceánico utilizados durante la ejecución del modelo acoplado océano- atmósfera. Los ejecutables restantes se utilizan para la inicialización del océano.

2.10.4. COMPILANDO GFDL VORTEX TRACKER

El GFDL Vortex Tracker requiere dos librerías externas, W3 y BACIO. Estas librerías se encuentran en el directorio `hwrp-utility/libs/` y deberían estar disponibles si `HWRP-utilities` se construyen correctamente, por lo que estas utilidades deber crearse antes de Vortex Tracker. Considerando que `dir_hwrp` corresponde al directorio base de nuestra instalación del HWRP, configuramos las variables de ambiente de la siguiente manera

```
export LIB_W3_PATH=${dir_hwrp}/hwrp-utilities/libs/  
export LIB_BACIO_PATH=${dir_hwrp}/hwrp-utilities/libs/
```

La configuración de GFDL Vortex Tracker para su compilación, dentro del directorio `gfdl-vortextracker` y ejecutamos

```
./configure
```

El script de configuración comprueba el hardware del sistema, y si las rutas de las variables no están establecidas, pide las rutas de software a las bibliotecas W3 y BACIO. Concluye pidiendo al usuario que elija una configuración soportada por la arquitectura actual de la máquina. Seleccionamos la opción 1

```
1. Linux x86_64, PGI compiler
```

El script `configure` crea un archivo llamado `configure.trk`. Este archivo contiene opciones de compilación, reglas y rutas específicas para la arquitectura actual de la máquina.

El archivo de configuración se puede editar para cambiar las opciones de compilación en caso que sea necesario.

Para compilar Vortex Tracker y guardar la salida de generación en un archivo de registro, ejecutamos

```
./compile 2>&1 | tee tracker.log
```

Para eliminar los archivos objetos ejecutamos

```
./clean
```

Para llevar a cabo una limpieza completa que elimine todos los archivos, objetos, archivos ejecutables y el archivo de configuración `configure.trk`, ejecutamos

```
./clean -a
```

Se recomienda encarecidamente una limpieza completa si la compilación falla al generar o si se cambia el archivo de configuración.

Si la compilación es exitosa, se generaran 3 archivos ejecutables en el directorio `trk_exec/`

- `hwrif_gettrk.exe`
- `hwrif_tave.exe`
- `hwrif_vint.exe`

2.10.5. COMPILANDO NCEP COUPLER

La configuración de NCEP Coupler para su compilación, entramos al directorio `ncep-coupler` y ejecutamos

```
./configure
```

El script de configuración revisa el hardware del Sistema, pide al usuario que elija una configuración soportada por la arquitectura actual de la máquina, y crea un archivo de configuración denominado `configure.cpl`. Seleccionamos la opción 1

```
1. Linux x86_64, PGI compiler (dmpar)
```

El archivo de configuración `configure.cpl` contiene opciones de compilación, reglas y rutas específicas de la arquitectura de la máquina actual, y se puede editar para cambiar las opciones de compilación si es necesario.

Para compilar el acoplador y guardar la salida de generación en un archivo de registro, ejecutamos

```
./compile 2>&1 | tee coupler.log
```

Para eliminar los archivos objetos ejecutamos

```
./clean
```

Para llevar a cabo una limpieza completa que elimine todos los archivos, objetos, archivos ejecutables y el archivo de configuración `configure.cpl`, ejecutamos

```
./clean -a
```

Se recomienda encarecidamente una limpieza completa si la compilación falla al generar o si se cambia el archivo de configuración.

Si la compilación es exitosa, se generará el ejecutable `hwrf_wm3c.exe` en el directorio `cpl_exec/`

2.10.6. COMPILANDO WPS

El módulo WPS requiere el mismo entorno de compilación que el modelo WRF-NMM, incluyendo las librerías `netCDF` y `MPI`. Dado que el WPS realiza llamadas directas a las librerías de la API de E/S incluidas con el modelo WRF, por tal motivo el modelo WRF-NMM debe construirse antes de crear el WPS.

Considerando que `dir_hwrf` corresponde al directorio base de nuestra instalación del HWRF, configuramos el entorno del compilador para el WPS estableciendo la variable de ambiente de la siguiente manera

```
export WRF_DIR=${dir_hwrf}/WRFV3/
```

Dentro del directorio WPS, ejecutamos el comando siguiente

```
./configure
```

Seleccionamos la opción apropiada para memoria distribuida “`dmpar`” para nuestra arquitectura. Seleccionamos la opción 7

7. Linux x86_64, PGI compiler (dmpar)

Para compilar WPS y guardar la salida de generación en un archivo de registro, ejecutamos

```
./compile 2>&1 | tee wps.log
```

Para realizar una limpieza completa que elimine todos los archivos incorporados en todos los directorios, así como el configure.wps, ejecutamos

```
./clean -a
```

Se recomienda una limpieza completa si la compilación falla o si se cambia el archivo de configuración.

Después de emitir el comando compile, una compilación exitosa de WPS produce los tres enlaces simbólicos: geogrid.exe, ungrid.exe y metgrid.exe en el directorio main/, y un solo enlace simbólico mod_levs.exe, usado para la configuración idealizada de ciclones tropicales, en el directorio util/. Si alguno de estos enlaces no existe, compruebe el archivo de registro de compilación para determinar lo que salió mal.

2.10.7. COMPILANDO UPP

El post-procesador unificado de NCEP (UPP) fue diseñado para interpolar la salida de WRF de coordenadas y variables nativas a coordenadas y variables más útiles para el análisis. Específicamente, la UPP desacelera la salida HWRF, interpola los datos desde su cuadrícula vertical nativa a niveles estándar y crea variables de diagnóstico adicionales.

UPP requiere los mismos compiladores Fortran y C utilizados para construir el modelo WRF. Además, UPP requiere la librería netCDF y la API de E/S del WRF (esta última se incluye con la compilación WRF).

La compilación UPP requiere varias librerías de soporte (IP, SP, W3), que se proporcionan con el código fuente y se encuentran en el directorio UPP/lib/. Estas librerías son para la compilación UPP solamente. No deben confundirse con las librerías del mismo nombre que se encuentran en el directorio hwrf-utilities/libs/.

Dado que la construcción UPP requiere vincular a las bibliotecas WRF-NMM API de E/S, debe ser capaz de encontrar el directorio WRF. La construcción UPP utiliza la variable de entorno WRF_DIR para definir la ruta de acceso a WRF. Por lo tanto, la variable de ruta WRF_DIR debe establecerse en la ubicación del directorio raíz WRF.

Además de establecer la variable de ruta, la creación de UPP para su uso con HWRF requiere establecer la variable de entorno HWRF. Esta es la misma variable establecida para construir WRF-NMM para HWRF.

```
export HWRF=1
export WRF_DIR=${SCRATCH}/HWRF/WRFV3/
```

UPP utiliza un mecanismo de compilación similar al utilizado por el modelo WRF, ejecutamos

```
./configure
```

Para generar el archivo de configuración UPP. El script de configuración se quejará si no se ha establecido la ruta WRF_DIR. A continuación, se le dará una lista de opciones de configuración adaptadas a su computadora. Seleccionamos la opción 2

```
2. Linux x86_64, PGI compiler (dmpar)
```

El script de configuración generará el archivo de configuración configure.upp. Si es necesario, el archivo configure.upp se puede modificar para cambiar las opciones y rutas de compilación predeterminadas. Para compilar ejecutamos

```
./compile 2>&1 | tee build.log
```

Este comando crea ocho librerías UPP en lib/ (libCRTM.a, libbacio.a, libip.a, libmersenne.a, libsfcio.a, libsigio.a, libsp.a, y libw3.a), y tres ejecutables UPP en Bin/ (unipost.exe, ndate.exe y copygb.exe). Una vez más, estas librerías son para la UPP solamente, y no deben ser utilizadas por los otros componentes. Para eliminar todos los archivos creados, así como el configure.upp, ejecutamos

```
./clean
```

Esto se recomienda si la compilación falló o si se ha cambiado el código fuente.

2.10.8. COMPILANDO GSI

El módulo GSI de la comunidad requiere el mismo entorno de compilación que el modelo WRF-NMM, incluyendo las librerías netCDF, MPI y LAPACK. Además, GSI hace llamadas directas a las librerías de API de E/S WRF incluidas con el modelo WRF. Por lo tanto, el modelo WRF debe ser construido antes de la construcción del GSI.

La construcción de GSI para su uso con HWRF requiere establecer tres variables ambientales. El primero, HWRF indica activar las opciones HWRF en la generación GSI. Este es el mismo indicador establecido cuando se construye WRF-NMM para HWRF. El segundo es una variable de ruta que apunta a la raíz del directorio de compilación de WRF. La tercera es la variable LAPACK_PATH, que indica la ubicación de la biblioteca LAPACK en el sistema.

```
export HWRF=1
export WRF_DIR=${SCRATCH}/HWRF/WRFV3
```

La variable de entorno adicional LAPACK_PATH puede ser necesaria en algunos sistemas. Normalmente, la variable de entorno LAPACK_PATH solo necesita ser configurada en sistemas Linux sin una versión de LAPACK suministrada. Los problemas con la librería LAPACK suministrada por el vendedor son más probables de ocurrir con el compilador de Intel. Mientras que los compiladores Intel normalmente tienen las bibliotecas MKL instaladas, el compilador ifort no carga automáticamente la biblioteca. Por lo tanto, es necesario establecer la variable LAPACK_PATH a la ubicación de las bibliotecas MKL cuando se utiliza el compilador Intel.

Suponiendo que la ruta de la biblioteca MKL se establece en la variable de entorno MKL, entonces el entorno LAPACK para bash es

```
export LAPACK_PATH=${MKL}
```

Para construir GSI para HWRF, entramos al directorio GSI y ejecutamos el comando

```
./configure
```

Elegimos la opción la numero 1

```
1. Linux x86_64, PGI compilers (pgf90 & pgcc) (dmpar)
```

Después de seleccionar la opción adecuada, ejecute la secuencia de comandos de compilación (bash)

```
./compile 2>&1 | tee build.log
```

Para realizar una limpieza completa que elimine todos los archivos incorporados en todos los directorios, así como el `configure.gsi`, ejecutamos

```
./clean -a
```

Se recomienda una limpieza completa si la compilación falla o si se cambia el archivo de configuración.

Siguiendo el comando de compilación, el programa ejecutable de GSI “`gsi.exe`” se puede encontrar en el directorio `run/`. Si no se encuentra el archivo ejecutable, compruebe el archivo de registro de compilación para determinar lo que salió mal. Otras herramientas, como `ssrc.exe`, que se utiliza para convertir de alguno de los formatos endianness a un archivo de datos binario, se pueden encontrar en el directorio `util/ test`.

2.11. CONFIGURACIÓN Y PRUEBAS DEL MODELO HWRF

Antes de configurar y ejecutar el modelo HWRF se debe tener claro la estructura de archivos y directorios del código fuente y programas ejecutables (figura 2.1), datasets (figura 2.2) y de los scripts de ejecución (figura 2.3).

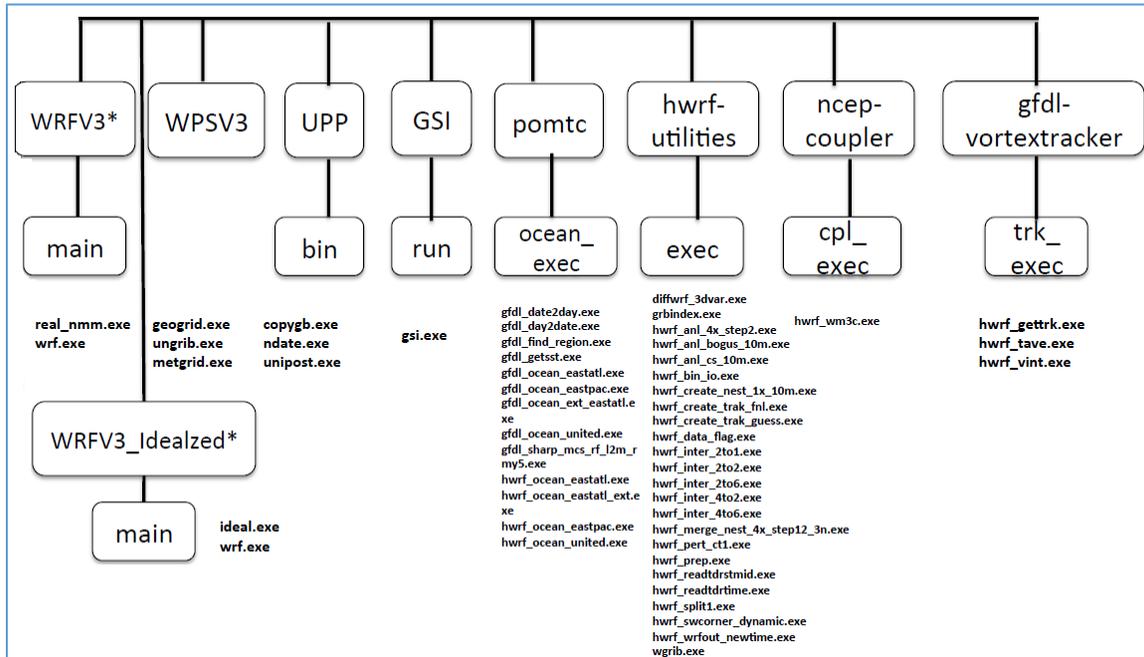


Figura 2.1. Estructura del código fuente y programas ejecutables del sistema HWRF

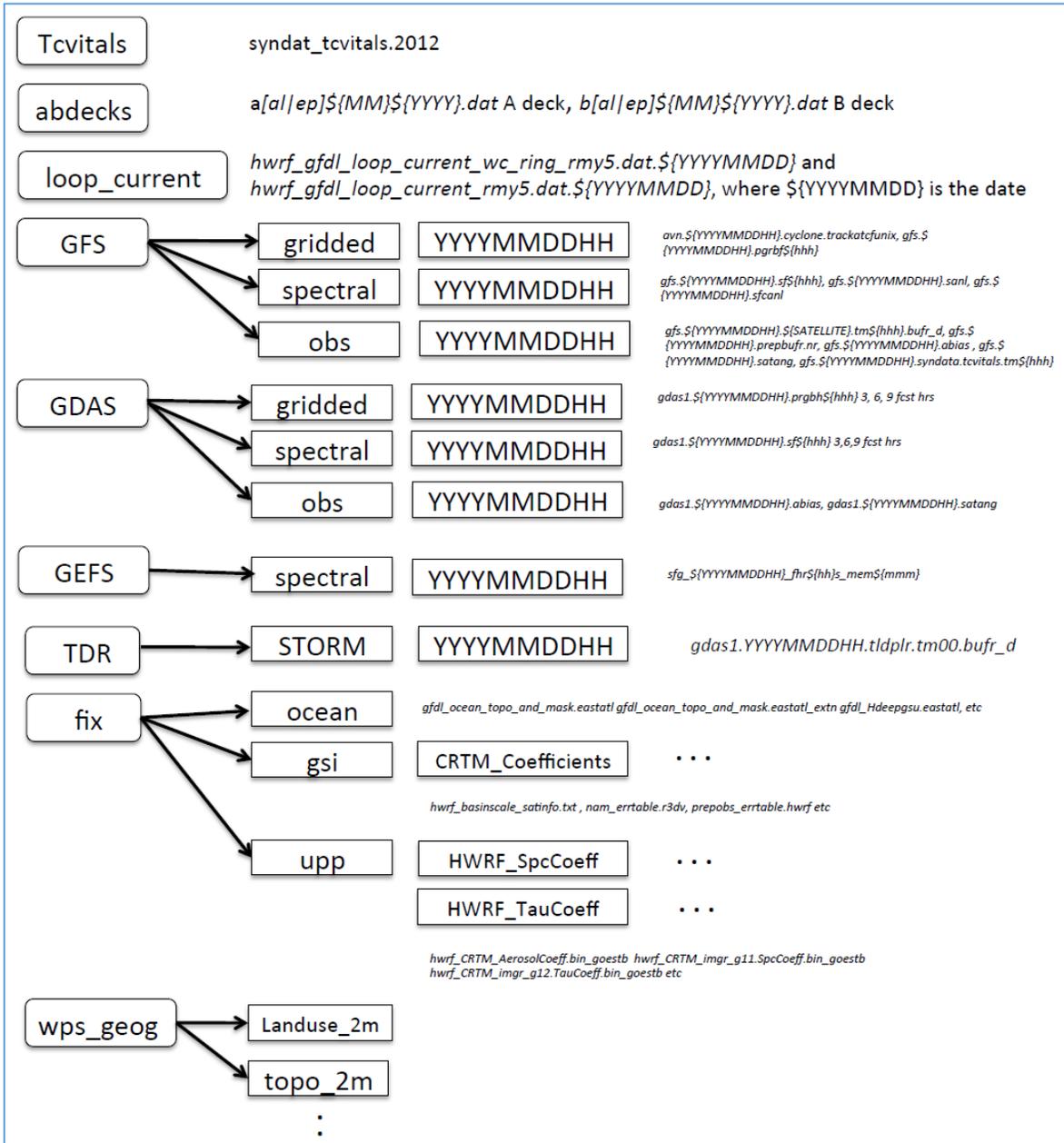


Figura 2.2. Estructura del datasets

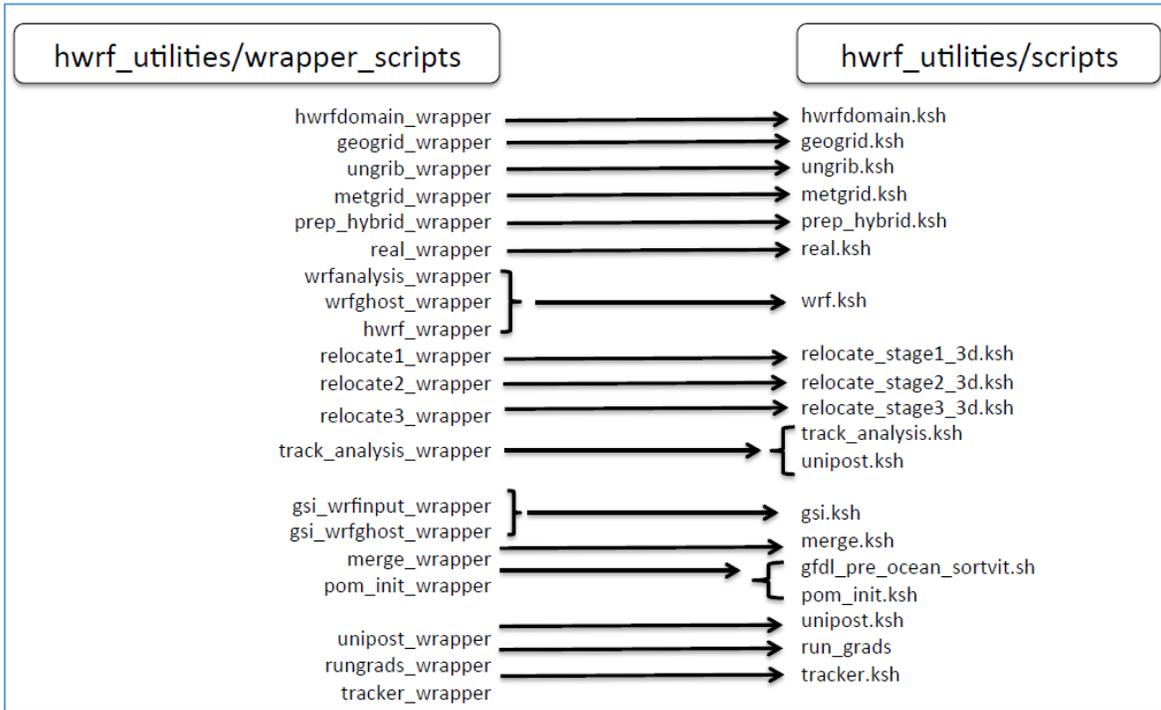


Figura 2.3. Estructura de scripts de ejecución

El primer paso es editar el archivo `global_vars.ksh` que se encuentra en el directorio `hwrf_utilities/wrapper_scripts/global_vars.ksh` y editamos las variables de la tabla 2.1.

Tabla 2.1. Variables a editar del archivo `global_vars.ksh`

Variable	Comentario
<code>export START_TIME=2012102806</code>	Tiempo de inicialización YYYYMMDDHH
<code>export START_TIME_MINUS6=2012102800</code>	Tiempo de inicialización menos 6 horas YYYYMMDDHH
<code>export FCST_LENGTH=12</code>	Longitud del pronóstico en horas
<code>export FCST_INTERVAL=6</code>	Intervalo del pronóstico en horas
<code>export STORM_NAME=SANDY</code>	Nombre de la tormenta
<code>export SID=18L</code>	Identificador de la tormenta (L para el Atlántico, E para el Pacífico Este y W para Pacífico Oeste)
<code>export BASIN=AL</code>	Cuenca (AL para el Atlántico, EP para el Pacífico Este y WP para Pacífico Oeste)

export RUN_PREP_HYB=T	T para correr prep_hybrid y F para no correrlo
export UPP_PROD_SAT=T	T para almacenar variables de salida en UPP y N para no salvar.
export RUN_GSI=T	T para correr GSI y N para no correr GSI
export HWRP_SRC_DIR=\$USER/HWRP3.5b/sorc	Nivel superior del directorio HWRP
export HWRP_DATA_DIR=\$USER/HWRP/datasets	Nivel superior del directorio de datasets
export CYCLE_DATA=\${HWRP_OUTPUT_DIR}/\${SID}/\${START_TIME MINUS6}	Directorio del ciclo
export DOMAIN_DATA=\${HWRP_OUTPUT_DIR}/\${SID}/\${START_TIME}	Nivel superior del directorio de salida
export GFS_DIR=\${HWRP_DATA_DIR}/GFS	Nivel superior del directorio de los datos GFS
export GFS SPECTRAL DIR=\${GFS_DIR}/spectral	
export GFS GRIDDED DIR=\${GFS_DIR}/gridded	
export GFS_OBS_DIR=\${GFS_DIR}/obs	
export GEFS_DIR=\${HWRP_DATA_DIR}/GEFS	Nivel superior del directorio de los datos GEFS
export GEFS ENS FCST DIR=\${GEFS_DIR}/spectral	
export GDAS_DIR=\${HWRP_DATA_DIR}/GDAS	Nivel superior del directorio de los datos GDAS
export GDAS SPECTRAL DIR=\${GDAS_DIR}/spectral	
export GDAS GRIDDED DIR=\${GDAS_DIR}/gridded	
export GDAS_OBS_DIR=\${GDAS_DIR}/obs	
export TCVITALS=\${HWRP_DATA_DIR}/Tcvitals	Directorio TCVitals
export OCEAN FIXED DIR=\${HWRP_DATA_DIR}/ocean	Archivos ocean fix
export LOOP_CURRENT_DIR=\${HWRP_DATA_DIR}/loop_current	Corriente de ciclo oceanico
export CRTM_FIXED_DIR=\${HWRP_DATA_DIR}/upp	Archivo UPP fix
export GEOG_DATA_PATH=\${HWRP_DATA_DIR}/wps_geog	Archivos geog de WPS
export GSI_FIXED_DIR=\${HWRP_DATA_DIR}/gsi	Archivos GSI fix

export GSI_CRTM_FIXED_DIR=\${GSI_FIXED_DIR}/CRTM_Coefficients	Coeficiente CRTM de GSI
export TDR_OBS_DIR=\${HWRP_DATA_DIR}/TDR	Datos TDR
export MPIRUN=mpirun.lsf	Comando MPIRUN
export WRF_ANAL_CORES=12	Núcleos recomendados para cada módulo
export WRF_GHOST_CORES=12	
export HWRP_FCST_CORES=202	
export GEOGRID_CORES=12	
export METGRID_CORES=12	
export REAL_CORES=1	
export GSI_CORES=200	
export UNI_CORES=12	
export PREP_HYB_CORES=12	
export ATCFNAME=HCOM	Identificador ATCF
export GRADS_BIN=/opt/grads/2.0.2/bin	Ruta de GrADS
export GADDIR=/opt/grads/2.0.2/data	

El archivo hwrp-utilities/parm/namelist se modificara en tiempo de ejecución de acuerdo al archivo global_vars.ksh, por lo que no se debe modificar (figura 2.4).

```

num_metgrid_soil_levels = @[GFS_SOURCE==GRIB_1x1_REDUCED?2:4],
/
&physics
num_soil_layers = 4,
mp_physics = 85, 85, 85,
ra_lw_physics = 98, 98, 98,
ra_sw_physics = 98, 98, 98,
sf_sfclay_physics = 88, 88, 88,
sf_surface_physics = 88, 88, 88,
bl_pbl_physics = 3, 3, 3,
cu_physics = 84, 84, 0,
mommix = 1.0, 1.0, 1.0,
var_ric = 1.0,
coef_ric_l = 0.16,
coef_ric_s = 0.25,
h_diff = 1.0, 1.0, 1.0,
gwd_opt = @[GWD==YES?2:0], 0, 0,

```

Figura 2.4. archivo hwrp-utilities/parm/namelist

2.11.1. CONFIGURAR EL DOMINIO HWRF

Ejecutamos el programa hwrp-utilities/wrapper_scripts/hwrpdomain_wrapper el cual realiza lo siguiente:

- Verifica las variables de ambiente
- Creará el directorio de trabajo ($\{\text{HWRP_OUTPUT_DIR}\}/\{\text{SID}\}/\{\text{START_TIME}\}/\text{messages}$)
- Creará el archivo tcvital y tcvitals.as a partir del archive Tcvitals para la tormenta en particular
- Determina el centro de la tormenta
- Determina el centro del dominio
- Crea el archivo de estatus GSI (si RUN_GIS es verdadero genera el archivo go_gsi, si no, generara el archivo no_gsi).

2.11.2. EJECUTAR WPS

Ejecutamos hwrp-utilities/wrapper_scripts/geogrid_wrapper el cual realizara lo siguiente:

- Verifica las variables de ambiente
- Verifica los ejecutables
- Crea el directorio de trabajo ($\{\text{HWRP_OUTPUT_DIR}\}/\{\text{SID}\}/\{\text{START_TIME}\}/\text{geoprd}$)
- Crea el archivo namelist
- Copia el archivo de tablas geogrid
- Ejecuta geogrid.exe generando los archivos de los tres dominios geo_nmm.d01.nc, geo_nmm_nest.l01.nc y geo_nmm_nest.l02.nc

Ejecutamos hwrp-utilities/wrapper_scripts/ungrid_wrapper el cual realizara lo siguiente:

- Verifica las variables de ambiente
- Verifica los ejecutables
- Crea el directorio de trabajo ($\{\text{HWRP_OUTPUT_DIR}\}/\{\text{SID}\}/\{\text{START_TIME}\}/\text{ungribprd}/\{\text{YYYYMMDDHH}\}$)
- Crea el archivo namelist
- Copia el archivo de tablas ungrid

- Liga los archivos de datos grib
- Ejecuta ungrid.exe generando los archivos FILE:YYYY-MM-DD_HH

Ejecutamos hwr-f-utilities/wrapper_scripts/metgrid_wrapper el cual realizara lo siguiente:

- Verifica las variables de ambiente
- Verifica los ejecutables
- Crea el directorio de trabajo
($\${HWR_OUTPUT_DIR}\${SID}\${START_TIME}/metgridprd/\${YYYYMMDDHH}$)
- Crea el archivo namelist
- Copia el archivo de tablas metgrid
- Copia los archivos de salida de geogrid
- Ejecuta metgrid.exe generando los archivos met_nmm.d01.YYYY-MM-DD_HH.nc

2.11.3. EJECUCIÓN DE PREP_HYBRID

Ejecutamos el programa hwr-f-utilities/wrapper_scripts/prep_hybrid_wrapper el cual realizará lo siguiente:

- Verifica las variables de ambiente
- Verifica los ejecutables
- Crea el directorio de trabajo
($\${HWR_OUTPUT_DIR}\${SID}\${START_TIME}/prep_hybrid/\${YYYYMMDDHH}$)
- Copia los archivos de entrada
- Ejecuta prep_hybrid generando los archivos hwrfini00, hwrfbcs00_00 y hwrfbcs00_01.

2.11.4. EJECUCIÓN DE REAL

Ejecutamos el programa hwr-f-utilities/wrapper_scripts/real_wrapper el cual realizará lo siguiente:

- Verifica las variables de ambiente
- Verifica los ejecutables

- Crea el directorio de trabajo ($\${HWR_OUTPUT_DIR}/\${SID}/\${START_TIME}/realprd/\${YYYYMMDDHH}$)
- Liga los archivos de entrada
- Corre hwr_swcorner_dynamic.exe para calcular la localización del dominio
- Genera el archivo namelist
- Se asegura que la pila este configurada mayor a 2 GB
- Ejecuta real.exe generando los archivos wrfinput_d01, wrfbdy_d01 y fort.65

2.11.5. EJECUCIÓN DE GSI: GSI_WRFINPUT

Ejecutamos el programa hwr_utilities/wrapper_scripts/gsi_wrfinput_wrapper el cual realizará lo siguiente:

- Verifica las variables de ambiente
- Verifica los ejecutables
- Crea el directorio de trabajo ($\${HWR_OUTPUT_DIR}/\${SID}/\${START_TIME}/gsiprd/\${YYYYMMDDHH}$)
- Copia los archivos fix y el analisis del campo de fondo
- Señala los datos observados cercanos al centro de la tormenta si estos no han sido asimilados
- Genera el archivo namelist
- Copia los datos ensamblados
- Ejecuta gsi.exe y genera los archivos stdout y wrf_inout

2.11.6. EJECUCIÓN DE WRF ANALISYS

Ejecutamos el programa hwr_utilities/wrapper_scripts/wrfanalysis_wrapper el cual realizará lo siguiente:

- Verifica las variables de ambiente
- Verifica los ejecutables
- Crea el directorio de trabajo ($\${HWR_OUTPUT_DIR}/\${SID}/\${START_TIME}/wrfanalysisprd/\${YYYYMMDDHH}$)
- Liga los archivos fix, las condiciones iniciales y de frontera.

- Ejecuta `hwrw_swcorner_dinamic.exe` para calcular los valores `istart` y `jstart` para el dominio central.
- Genera el archivo `namelist`
- Ejecuta `wrfanalisis` y genera los siguientes archivos `wrfanl_d02`, `wrfanl_d03`

2.11.7. EJECUCIÓN DE WRF GHOST

Ejecutamos el programa `hwrw-utilities/wrapper_scripts/wrfghost_wrapper` el cual realizará lo siguiente:

- Verifica las variables de ambiente
- Verifica los ejecutables
- Crea el directorio de trabajo (`${HWRW_OUTPUT_DIR}/${SID}/${START_TIME}/wrfghostprd/${YYYYMMDDHH}`)
- Liga los archivos `fix`, las condiciones iniciales y de frontera.
- Ejecuta `hwrw_swcorner_dinamic.exe` para calcular los valores `istart` y `jstart` para el dominio central.
- Genera el archivo `namelist`
- Ejecuta `wrfghost` y genera los siguientes archivos `wrfanl_d02`, `wrfanl_d03`

2.11.8. EJECUCIÓN DE TRACKER ANALISYS

Ejecutamos el programa `hwrw-utilities/wrapper_scripts/track_analysis_wrapper` el cual realizará lo siguiente:

- Verifica las variables de ambiente
- Verifica los ejecutables
- Crea el directorio de trabajo (`${HWRW_OUTPUT_DIR}/${SID}/${START_TIME}/trkanalysisprd/${YYYYMMDDHH}`)
- Copia los archivos `fix` UPP y `wrfout_d01` desde la corrida del analisis 90-s
- Ejecuta `hwrw_wrfout_newtime.exe` para cambiar la marca del tiempo en 90-s `wrfout_d01` para el tiempo 0 para GFDL tracker para inicializar desde la hora 0.
- Ejecuta UPP para `wrfout_d01`
- Ejecuta `copygb` para desescalonar la salida a latitud y longitud normales

- Ejecuta GFDL vortex tracker generando el archivo gfs-anl-fix.atcfunix

2.11.9. PRUEBAS DEL MODELO HWRF

Las pruebas de descarga, compilación, ejecución y casos de estudio del modelo HWRF se realizaron en el equipo de cómputo de alto rendimiento tipo clúster denominado Turing, cuyas características generales son las siguientes:

- 1 nodo maestro con 32 cores y 128 GB de RAM
- 5 nodos de procesamiento con 64 cores y 256 GB de RAM
- 1 Almacenamiento tipo NAS de 50 TB
- 1 Switth Gigabit ethernet
- 1 Swicht infiniband de baja latencia a 40 Gbps

En la figura 1 se esquematiza la configuración del clúster.

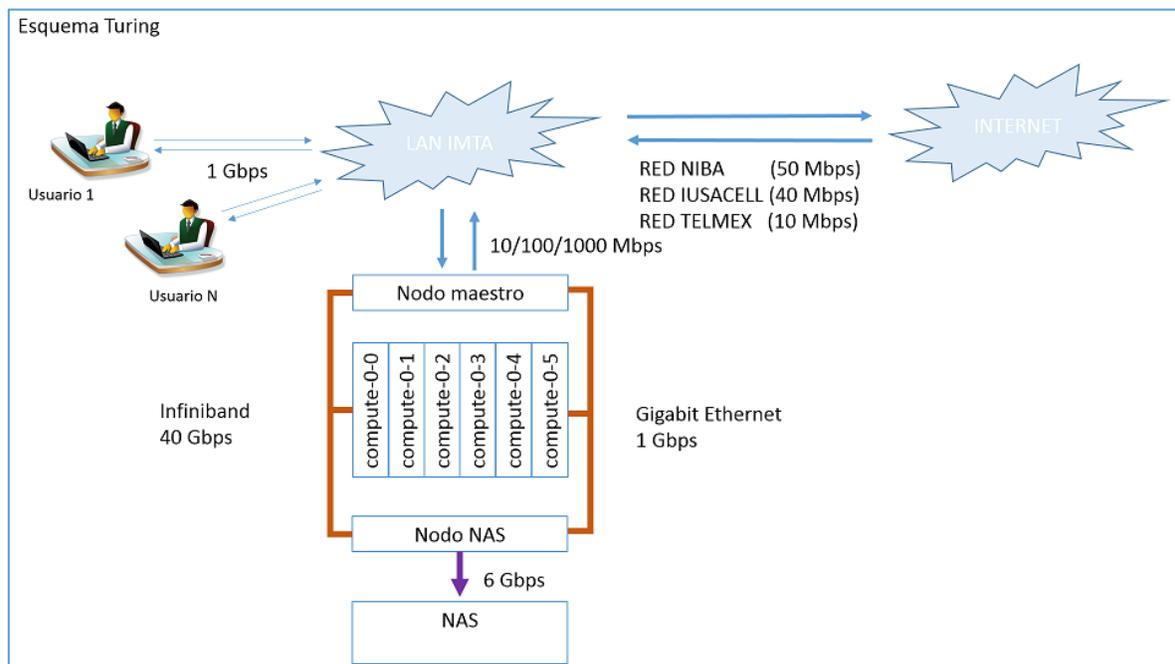


Figura 2.5. Esquema del clúster Turing

El sistema operativo del clúster es Rocks Linux basado en CentOS 6.2, además, cuenta con la licencia de compiladores CDK (Cluster Development Kit, por sus siglas en inglés) que incluye fortran, C y C++ para computo en paralelo.

Si bien este equipo de cómputo tiene altas prestaciones, también es cierto que tiene diferentes usuarios que demandan recursos para la ejecución de modelos numéricos, análisis de datos, entre otros procesos. Por lo que el primer limitante en las pruebas del modelo HWRF, durante la vigencia del proyecto, fue que se disponía de un solo nodo para trabajar (compute-0-1).

Se creó la cuenta de usuario "hwrf" cuyo directorio de trabajo por defecto es /storage/array1/hwrf.

Para realizar las pruebas se descargaron los datos del caso de estudio del huracán Sandy que inicializo el 28 de octubre de 2012. Los datos se descargaron de la página web <http://www.dtcenter.org/HurrWRF/users/downloads/observations/>.

Se descargaron y compilaron las versiones 3.5b y 3.7a, entre las diferencias principales entre versiones se encuentra la resolución de los dominios, pasando de 27/9/3 a 18/6/2 km de resolución para el dominio padre, anidado intermedio y anidado pequeño respectivamente.

Se ejecutaron satisfactoriamente los módulos WPS y REAL, pero los módulos subsecuentes (wrfanalysis, wrfghost, etc.) presentaron inconsistencia truncando las corridas, y en otras pruebas el proceso se fue al infinito, es decir, después de 5 días de ejecución el proceso continuaba en operación, por lo que se dedujo que el proceso es demasiado pesado para un solo nodo de procesamiento.

Para concluir las pruebas se requiere disponer de todos los nodos de procesamiento del clúster para realizar las pruebas, que al cierre de este proyecto no fue posible dado que existen otros usuarios con diversos procesos en operación.

3. DEFINICIÓN DE PRODUCTOS GRAFICOS

Para definir los productos gráficos que se tienen que generar a partir de la ejecución del sistema HWRF, sin lugar a equivocarnos, la referencia obligada de productos gráficos la encontramos en la página web oficial de la versión operativa del modelo HWRF de NCEP (http://www.emc.ncep.noaa.gov/gc_wmb/vxt/HWRF/). Al entrar en la página se presenta un mapa mundial sobre el cual se despliegan las trayectorias de las Tormentas Tropicales presentes al momento de la consulta, o en su defecto, de acuerdo a la fecha de consulta (figura 3.1).



Figura 3.1. Mapa base para desplegar las trayectorias de las Tormentas Tropicales
(Fuente <http://www.emc.ncep.noaa.gov/HWRF/>)

Del lado izquierdo se presenta un menú para seleccionar el periodo de consulta de las Tormentas Tropicales por año, mes, día y hora (figura 3.2).

Current Active TCs

Active TCs on
Year | Month | Day
at Hour UTC

▾
 ▾
 ▾

 ▾

Figura 3.2. Menú de consulta de Tormentas Tropicales
(Fuente <http://www.emc.ncep.noaa.gov/HWRF/>)

Seleccionamos la fecha siguiente 2012/10/23_12 y observamos las tres tormentas tropicales presentes en esta fecha: Invest92W, Sandy y Tony (figura 3.3). Al hacer clic sobre la tormenta, se mostrarán las gráficas correspondientes al evento. Con base a estos productos definimos como productos específicos de Tormentas Tropicales los siguientes:

- Trayectoria de la Tormenta Tropical (Figura 3.4)
- Viento máximo a 10 metros de altura de la superficie (Figura 3.5)
- Presión mínima en hPa (Figura 3.6)

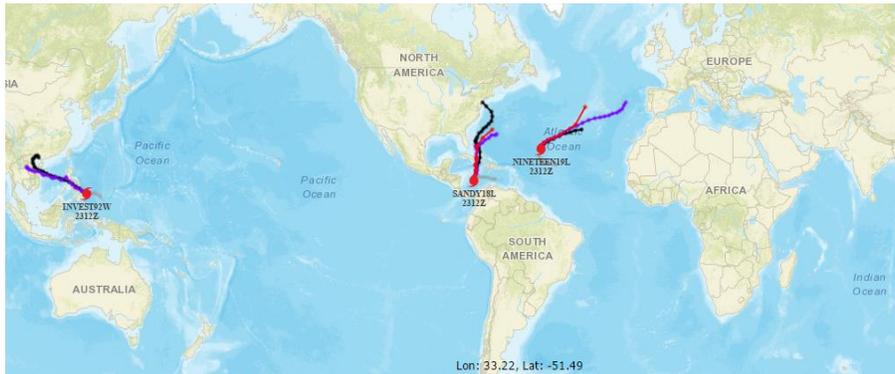


Figura 3.3. Menú de consulta de Tormentas Tropicales
(Fuente <http://www.emc.ncep.noaa.gov/HWRF/>)

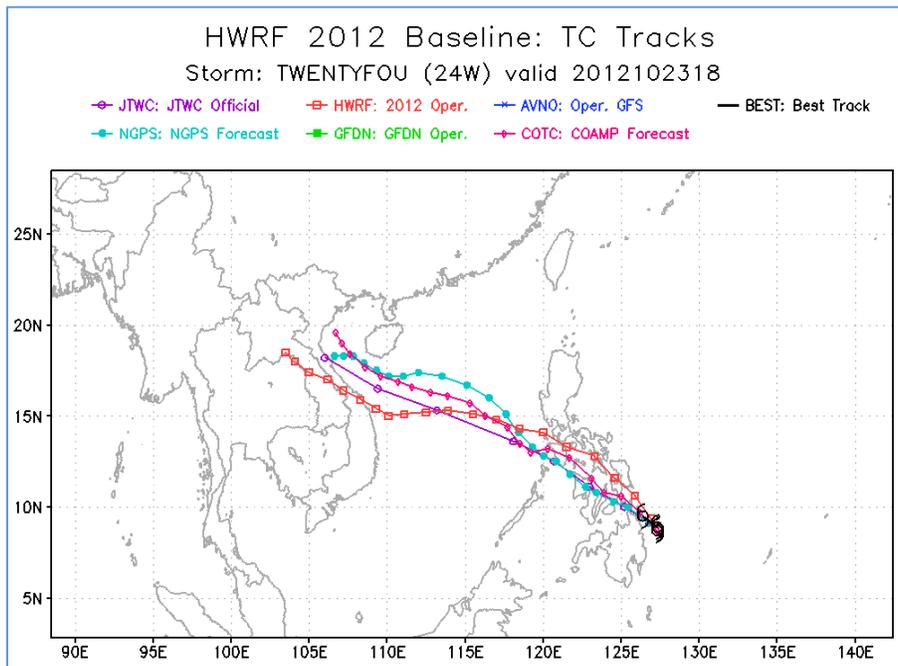


Figura 3.4. Ejemplo de producto grafico de pronóstico de trayectoria de tormentas tropicales
(Fuente http://www.emc.ncep.noaa.gov/HWRF/WestPac/RT_WPAC_FY12/TWENTYFOU24W/)

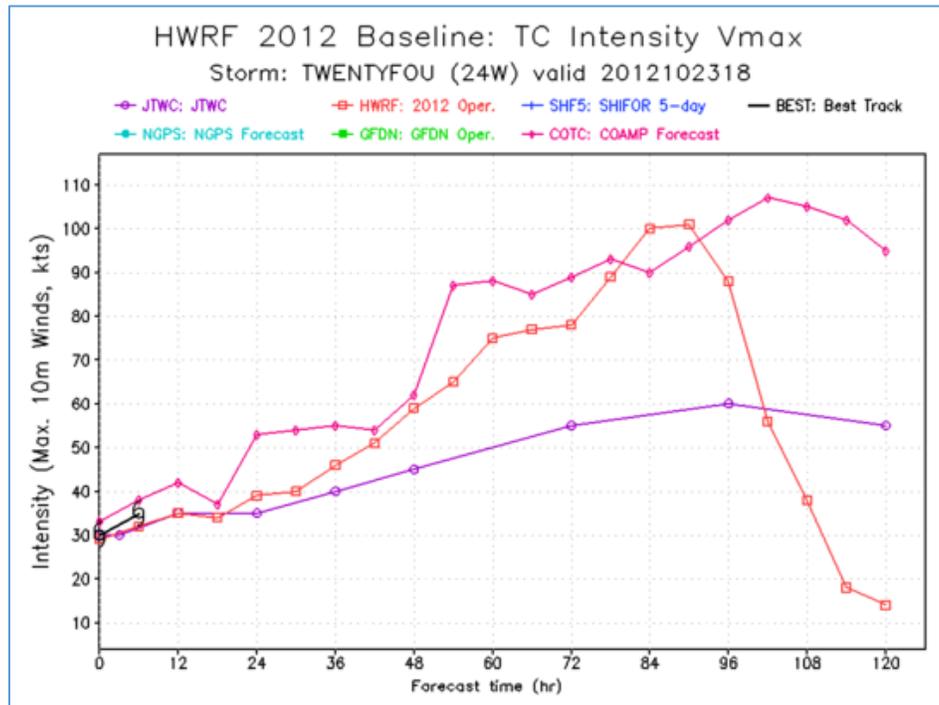


Figura 3.5. Ejemplo de producto grafico de pronóstico de viento máximo de tormentas tropicales (Fuente http://www.emc.ncep.noaa.gov/HWRf/WestPacific/RT_WPAC_FY12/TWENTYFOU24W/)

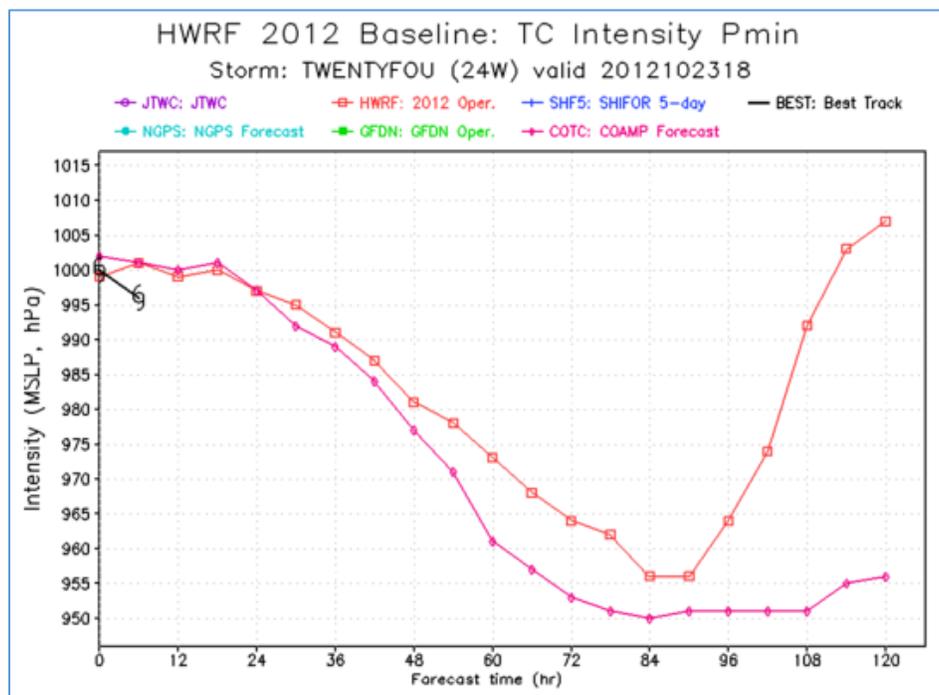


Figura 3.6. Ejemplo de producto grafico de pronóstico de presión mínima de tormentas tropicales (Fuente http://www.emc.ncep.noaa.gov/HWRf/WestPacific/RT_WPAC_FY12/TWENTYFOU24W/)

4. CASO DE ESTUDIO: HURACAN PATRICIA (2015)

Como sabemos, un modelo numérico es una representación matemática de un sistema real, en este caso las tormentas tropicales en sus diferentes facetas (depresión tropical, tormenta tropical o huracán). Estos modelos son una abstracción de la realidad y por ello son un instrumento valioso para predecir con anticipación su formación y evolución. No obstante, están susceptibles a errores y baja precisión, por ello, como parte de este proyecto se llevará a cabo un caso de estudio y su análisis correspondiente.

El evento seleccionado corresponde al huracán Patricia, el cual es el ciclón tropical más intenso jamás observado en el hemisferio occidental en términos de presión atmosférica, y el más fuerte a nivel global en términos de viento máximo sostenido. Este huracán se originó como una perturbación tropical al sur del golfo de Tehuantepec a mediados de octubre de 2015, el huracán Patricia fue clasificado como depresión tropical el 20 de octubre. Se fortaleció lentamente, sin embargo, el huracán Patricia comenzó a forzar profundización temprana el 22 de octubre, y horas más tarde la tormenta se intensificó hasta convertirse en el decimoquinto huracán de la temporada. En un principio se consideró que sería tan grave como los huracanes Kenna y Odile; pero los reportes de la madrugada del 23 de octubre, a las 3:30 a.m. Patricia se convirtió en un huracán de categoría 5 superando con ello al huracán Linda como el “más intenso” del Pacífico. En un principio fue considerado el “más peligroso” del que se tuviera registro en México; posteriormente fue catalogado como “el más peligroso del mundo” en la historia, afortunadamente, una vez tocada tierra en la costa de México el huracán fue perdiendo rápidamente fuerza, convirtiéndose en baja tropical.

Se analizaron las trayectorias pronosticadas por el modelo HWRF en comparación con la trayectoria definida por los modelos operativos de GFDL, NOGAPS y GFS, así como el pronóstico oficial del NHC. Los resultados en la mayoría de las fuentes mantienen una similitud en la trayectoria, sobre todo existe correspondencia entre el HWRF y el GFS, los cuales representaron de la mejor manera la trayectoria real del huracán.

A continuación se presentan los resultados gráficos derivados de las corridas del modelo HWRF que se ejecutaron de forma operativa por NCEP del 19 al 23 de octubre con inicialización de las 00 y 12 UTC.

4.1. PRONÓSTICO DE HWRF INICIALIZADO EL 19 DE OCTUBRE A LAS 00 UTC

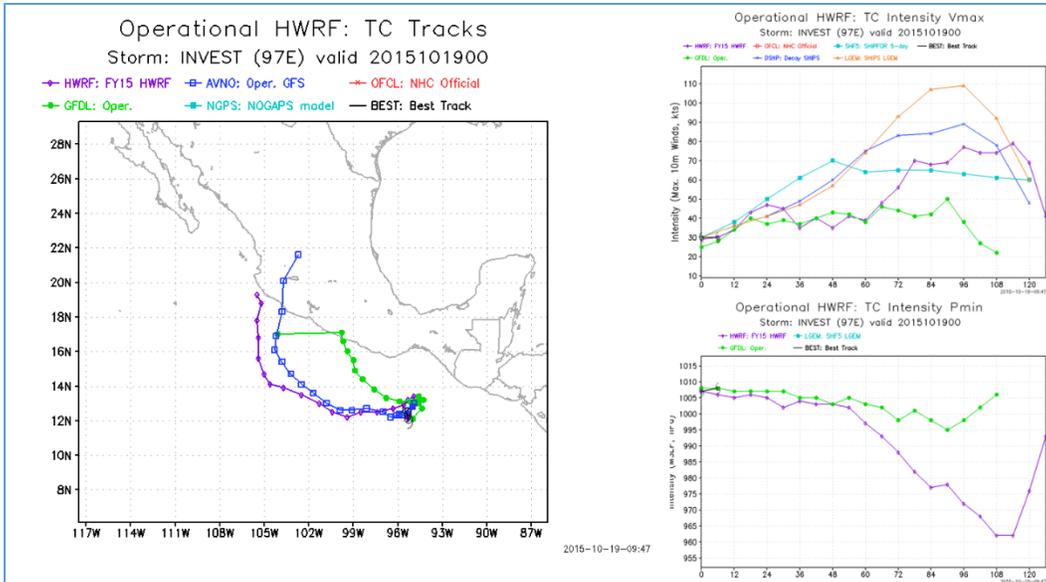


Figura 4.1. Pronóstico de HWRf inicializado el 19 de octubre a las 00 UTC

(Fuente: http://www.emc.ncep.noaa.gov/gc_wmb/vxt/RT_EASTPAC/PATRICIA20E/INVEST97E.2015101900/INVEST97E.2015101900.fsct.png)

4.2. PRONÓSTICO DE HWRf INICIALIZADO EL 19 DE OCTUBRE A LAS 12 UTC

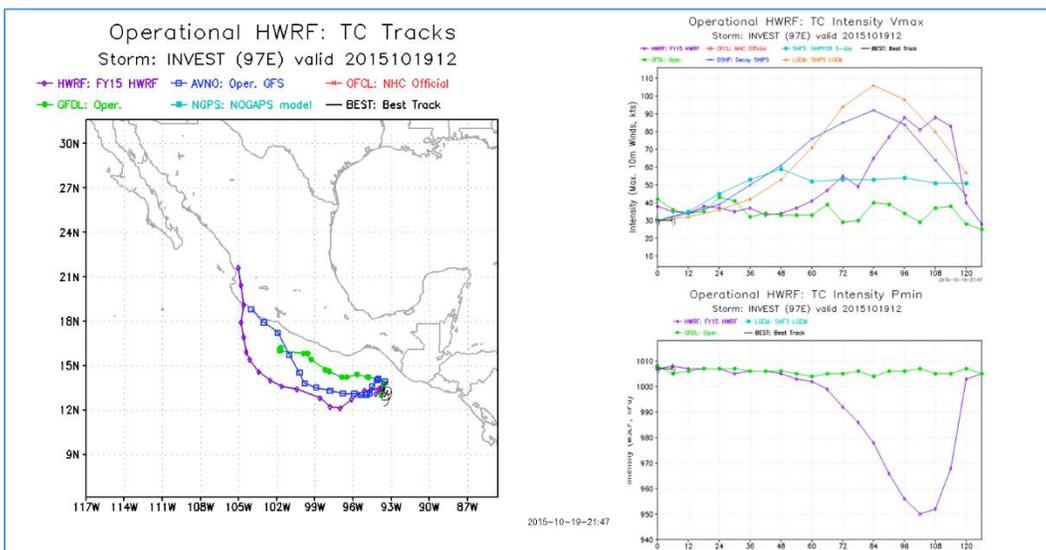


Figura 4.2. Pronóstico de HWRf inicializado el 19 de octubre a las 12 UTC

(Fuente: http://www.emc.ncep.noaa.gov/gc_wmb/vxt/RT_EASTPAC/PATRICIA20E/INVEST97E.2015101912/INVEST97E.2015101912.fsct.png)

4.3. PRONÓSTICO DE HWRP INICIALIZADO EL 20 DE OCTUBRE A LAS 00 UTC

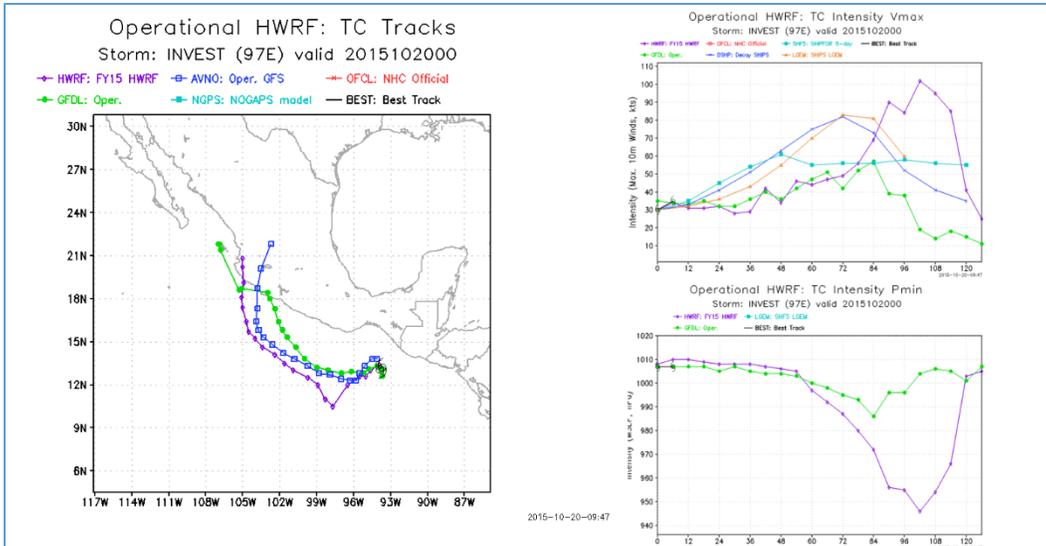


Figura 4.3. Pronóstico de HWRP inicializado el 20 de octubre a las 00 UTC

(Fuente: http://www.emc.ncep.noaa.gov/gc_wmb/vxt/RT_EASTPAC/INVEST97E/INVEST97E.2015102000/INVEST97E.2015102000.fcst.png)

4.4. PRONÓSTICO DE HWRP INICIALIZADO EL 20 DE OCTUBRE A LAS 12 UTC

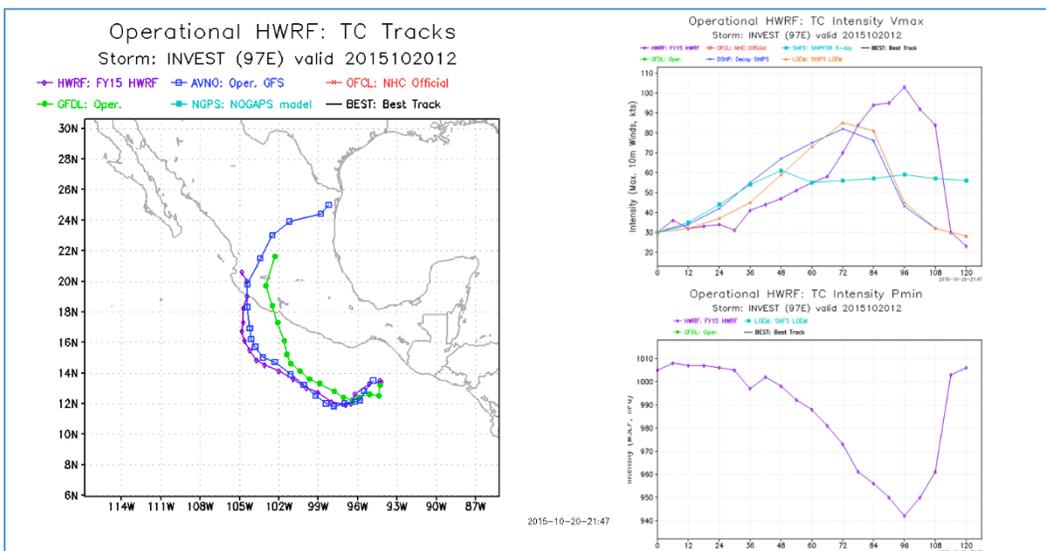


Figura 4.4. Pronóstico de HWRP inicializado el 20 de octubre a las 12 UTC

(Fuente: http://www.emc.ncep.noaa.gov/gc_wmb/vxt/RT_EASTPAC/INVEST97E/INVEST97E.2015102012/INVEST97E.2015102012.fcst.png)

4.5. PRONÓSTICO DE HWRF INICIALIZADO EL 21 DE OCTUBRE A LAS 00 UTC

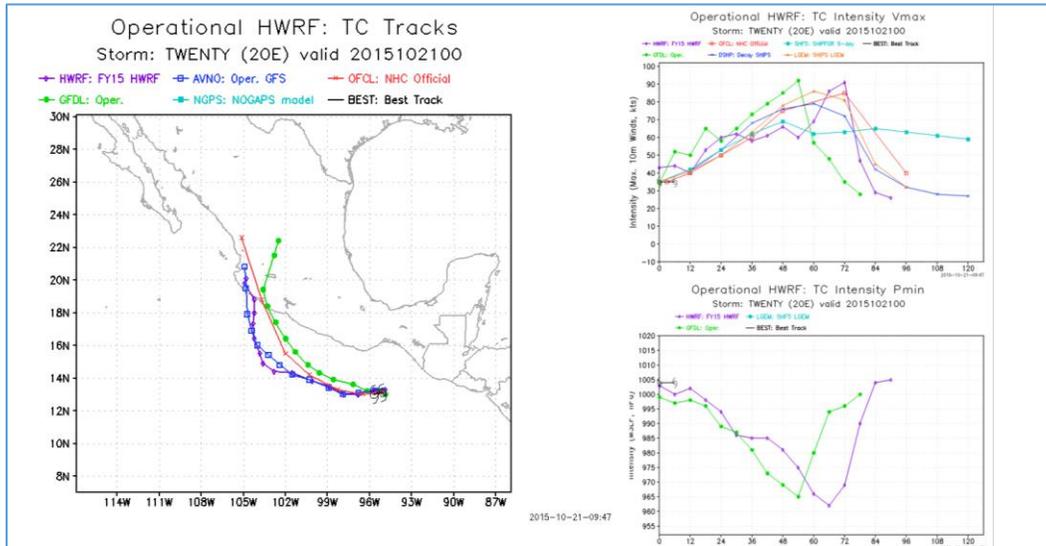


Figura 4.5. Pronóstico de HWRP inicializado el 21 de octubre a las 00 UTC

(Fuente: http://www.emc.ncep.noaa.gov/gc_wmb/vxt/RT_EASTPAC/PATRICIA20E/TWENTY20E.2015102100/TWENTY20E.2015102100.fsct.png)

4.6. PRONÓSTICO DE HWRF INICIALIZADO EL 21 DE OCTUBRE A LAS 12 UTC

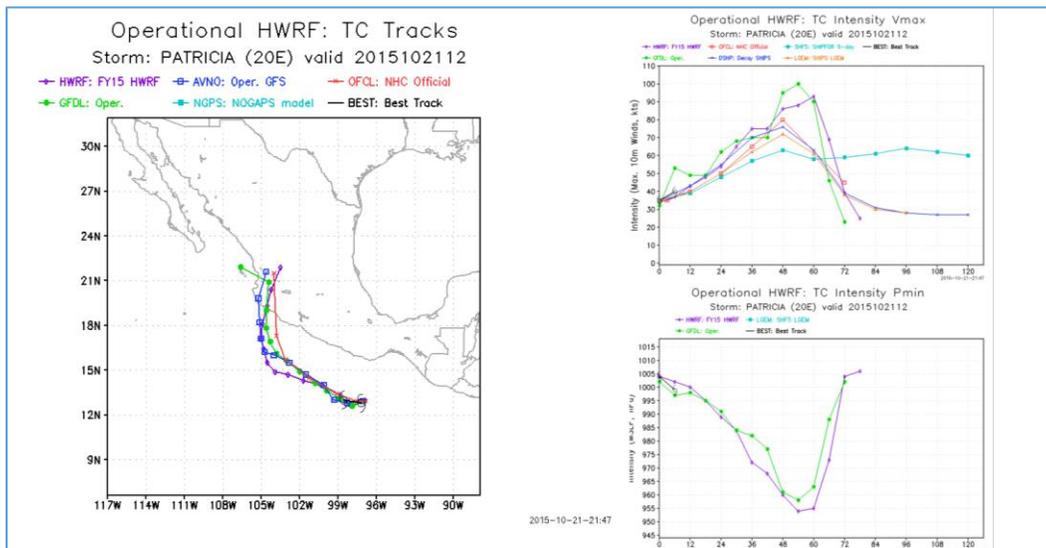


Figura 4.6. Pronóstico de HWRP inicializado el 21 de octubre a las 12 UTC

(Fuente: http://www.emc.ncep.noaa.gov/gc_wmb/vxt/RT_EASTPAC/PATRICIA20E/PATRICIA20E.2015102112/PATRICIA20E.2015102112.fsct.png)

4.7. PRONÓSTICO DE HWRP INICIALIZADO EL 22 DE OCTUBRE A LAS 00 UTC

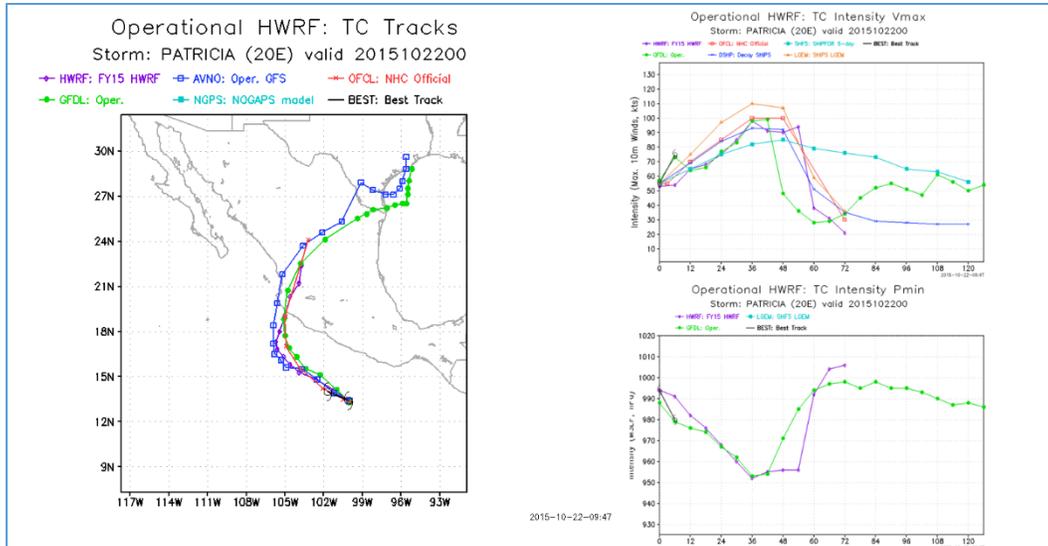


Figura 4.7. Pronóstico de HWRP inicializado el 22 de octubre a las 00 UTC

(Fuente: http://www.emc.ncep.noaa.gov/gc_wmb/vxt/RT_EASTPAC/PATRICIA20E/PATRICIA20E.2015102200/PATRICIA20E.2015102200.fsct.png)

4.8. PRONÓSTICO DE HWRP INICIALIZADO EL 22 DE OCTUBRE A LAS 12 UTC

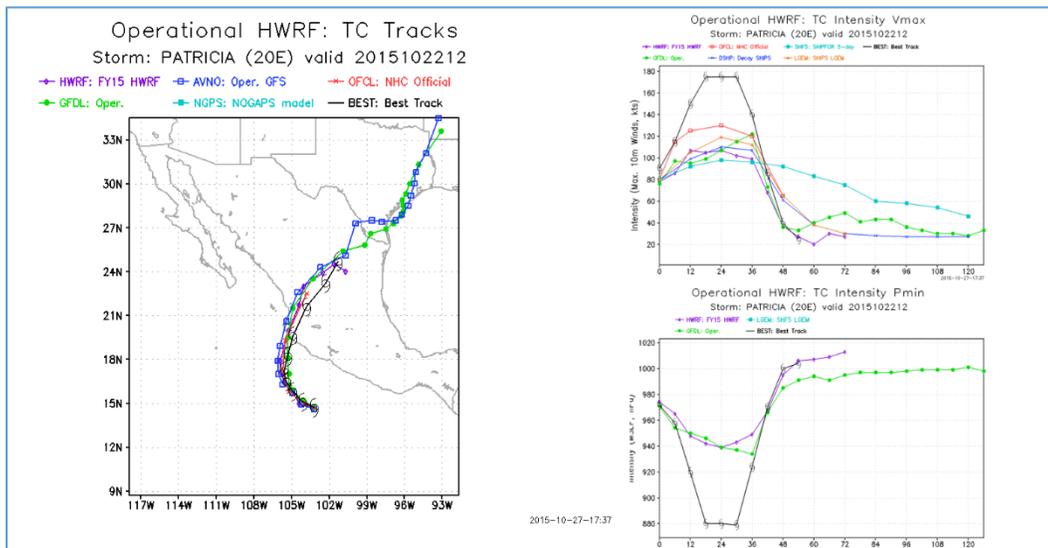


Figura 4.8. Pronóstico de HWRP inicializado el 22 de octubre a las 12 UTC

(Fuente: http://www.emc.ncep.noaa.gov/gc_wmb/vxt/RT_EASTPAC/PATRICIA20E/PATRICIA20E.2015102212/PATRICIA20E.2015102212.fsct.png)

4.9. PRONÓSTICO DE HWRP INICIALIZADO EL 23 DE OCTUBRE A LAS 00 UTC

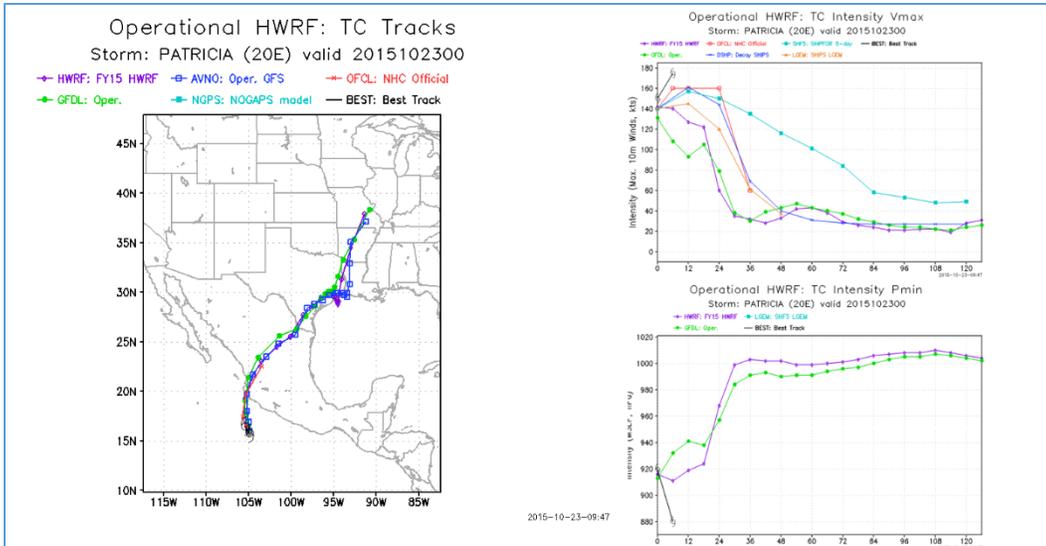


Figura 4.9. Pronóstico de HWRP inicializado el 23 de octubre a las 00 UTC

(Fuente: http://www.emc.ncep.noaa.gov/gc_wmb/vxt/RT_EASTPAC/PATRICIA20E/PATRICIA20E.2015102300/PATRICIA20E.2015102300.fsct.png)

4.10. PRONÓSTICO DE HWRP INICIALIZADO EL 23 DE OCTUBRE A LAS 12 UTC

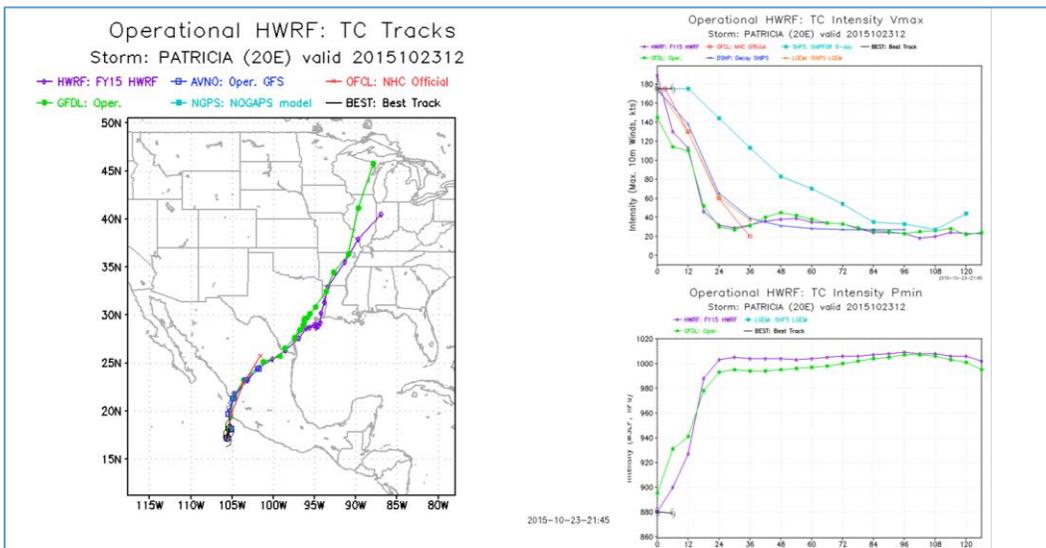


Figura 4.10. Pronóstico de HWRP inicializado el 23 de octubre a las 12 UTC

(Fuente: http://www.emc.ncep.noaa.gov/gc_wmb/vxt/RT_EASTPAC/PATRICIA20E/PATRICIA20E.2015102312/PATRICIA20E.2015102312.fsct.png)

CONCLUSIONES

El modelo HWRF está basado en el núcleo del modelo WRF que se ejecutan de forma operativa en NCEP. Se actualiza anualmente para proporcionar la mejor orientación numérica para el National Hurricane Center. Estas actualizaciones se benefician enormemente de las contribuciones de la comunidad fomentadas por el programa WRF, DTC y HFIP.

Como líneas de mejora del modelo HWRF incluye la adopción de física avanzada para aplicaciones tropicales de alta resolución. Como ejemplo, las pruebas que utilizan la microfísica de Thompson, la radiación RRTMG y el modelo de superficie terrestre de Noah están en curso.

La inicialización de HWRF es otra área en desarrollo activo. Se está probando el uso de radiancia de satélites, así como datos de reconocimiento de aeronaves más allá de TDR (dropsondes, vientos de imágenes de microondas de frecuencia escalonada, etc.).

Se está discutiendo el marco modelo para la HWRF. Varios modelos de NOAA, como el modelo de Mesoescala de América del Norte (NAM) Ahora se ejecutan usando NMM-B sin el marco WRF. En su lugar, emplean el Sistema de Modelación Ambiental de la NOAA (NEMS). No se ha decidido si HWRF hará la transición a NMM-B bajo NEMS, o permanecen en el marco WRF. Independientemente de la elección, se reconoce la importancia de la participación de la comunidad y juega un papel decisión.

Varios pasos importantes han sido tomados por DTC y colaboradores para hacer HWRF un código que se puede utilizar tanto para la investigación y como para pronóstico operativo. Se espera que la inversión de tiempo y esfuerzos dé dividendos al hacer posible que el código recién desarrollado esté fácilmente disponible para pruebas y consideración de posibles implementación. El apoyo a la comunidad de todos los componentes del HWRF hace posible que los investigadores de instituciones laboratorios gubernamentales a desarrollar en el mismo código utilizado por las operaciones, y la suite funcionalmente similar desarrollada y DTC soporta la integridad del código y permite una prueba eficiente de nuevas capacidades.

Como parte del esfuerzo de DTC para unir las comunidades de investigación y operación del NWP, HWRF es ahora un código apoyado por la comunidad en EMC. Los lanzamientos anuales del código estable probado se proporcionan a través del Web site de DTC, junto con un servicio de ayuda, amplia documentación, casos de prueba y conjuntos de datos. Además,

el DTC ha establecido un nuevo protocolo de gestión de códigos para HWRF, que permite Expertos para acceder, compartir y aportar código experimental.

El DTC lleva a cabo actividades de prueba y evaluación para HWRF. Estas incluyen pruebas previas a la liberación, pruebas para verificar que las nuevas capacidades han sido correctamente migradas a HWRF, ensayando para evaluar nuevos desarrollos para la implementación operacional potencial, y análisis de Conjuntos de datos aportados por grupos externos.

El objetivo de estos esfuerzos es colaboración entre la investigación y las comunidades operativas, con el fin de acelerar la mejora de HWRF y la predicción operacional de huracanes.

Si bien en México existen casos de éxito de la implementación del modelo HWRF, específicamente por investigadores del CICESE. En el caso particular de este proyecto se tuvieron limitaciones técnicas al no disponer de un clúster con suficientes recursos disponibles para este proyecto, lo cual represento una de las más fuertes limitantes para concluir satisfactoriamente el proyecto. Además, queda en claro que el HWRF presenta una complejidad exponencial con respecto a la operación del WRF normal, no solo por la diversidad de módulos que lo componen, si no por el cambio de paradigma en la operación de los scripts de ejecución, utilizando ahora una combinación de programas en ksh y Python para su ejecución.

Como trabajos futuros se continuara investigando por el grupo de hidrometeorología del sobre la operación del modelo HWRF, así como la descarga y pruebas del modelo en su versión 3.8a que fue liberado recientemente, el 23 de noviembre, el cual resuelve algunos errores en tiempo de compilación, como es el caso del error `“./WRFV3/external/io_netcdf/./wrf_io.f:1578: undefined reference to `nf_open_”` que es señalado durante la compilación de algunos módulos, cuya solución establece el definir la variable de ambiente `NETCDF4=1`, además, esta nueva versión incluye la corrección de otros errores de versiones anteriores.

BIBLIOGRAFIA

- Ligia Bernardet, V. Tallapragada, S. Bao, Y. Kwon, S. Trahan, M. Tong, Q. Liu, X. Zhang, S. Gopalakrishnan, R. Yablonsky, T. Marchok; (2013). Hurricane WRF: 2013 operational implementation and community support.
- Bao, S., R. Yablonsky, D. Stark, and L. Bernardet; (2012). HWRF Users' Guide V3.4a. Developmental Testbed Center, 123 pp.
- Wolff , J., L. Nance, L. Bernardet and B. Brown; (2010). Implementation of WRF Reference Configurations. 11th WRF Users' Workshop, Boulder, CO.
- L. Bernardet, S. Bao, T. Brown, D. Stark, M. Biswas, L. Carson, and C. Harrop; (2012). Hurricane WRF: Testing activities and community support at the DTC.
- Diario la prensa; (10 de octubre de 2014). Los modelos para pronosticar huracanes, buenos pero perfectibles: AMC. Página web consultada el 15 de agosto de 2016 en <http://www.oem.com.mx/laprensa/notas/n3566765.htm#sthash.yfqkXn7k.dpuf>
- Erick Falcón; (2014). Modelación numérica permite estudiar el comportamiento de ciclones tropicales en el Pacífico. Pagina Web consultada el 12 de octubre del 2016.
- L. Bernardet, S. Bao, R. Yablonsky, D. Stark, T. Brown; (2013). COMMUNITY HWRF USERS GUIDE V3.5A. THE DEVELOPMENT TESTBED CENTER.

APENDICE A. MANUAL DE INSTALACIÓN Y OPERACIÓN DEL MODELO HWRF

A continuación se describe de forma resumida el procedimiento para instalar el modelo HWRF así como los pasos para su operación.

A.1. DESCARGA DE LOS MÓDULOS DEL HWRF

```
wget -c http://www.dtcenter.org/HurrWRF/users/downloads/HWRF_releases/  
V3.5b/hwrf_v3.5b_hwrf-utilities.tar.gz
```

```
wget -c http://www.dtcenter.org/HurrWRF/users/downloads/HWRF_releases/V3.5b/  
hwrf_v3.5b_pomtc.tar.gz
```

```
wget -c http://www.dtcenter.org/HurrWRF/users/downloads/HWRF_releases/V3.5b/  
hwrf_v3.5b_ncep-coupler.tar.gz
```

```
wget -c http://www.dtcenter.org/HurrWRF/users/downloads/HWRF_releases/V3.5b/  
hwrf_v3.5b_gfdl-vortextracker.tar.gz
```

```
wget -c http://www.dtcenter.org/HurrWRF/users/downloads/HWRF_releases/V3.5b/  
hwrf_v3.5b_GSI.tar.gz
```

```
wget -c http://www.dtcenter.org/HurrWRF/users/downloads/HWRF_releases/V3.5b/  
hwrf_v3.5b_WRFV3.tar.gz
```

```
wget -c http://www.dtcenter.org/HurrWRF/users/downloads/HWRF_releases/V3.5b/  
hwrf_v3.5b_WPSV3.tar.gz
```

```
wget -c http://www.dtcenter.org/HurrWRF/users/downloads/HWRF_releases/V3.5b/  
hwrf_v3.5b_UPP.tar.gz
```

A.2. DESEMPAQUETADO DE LOS MÓDULOS

Nos ubicamos en el directorio de instalación y ejecutamos

```
gunzip * .tar.gz  
tar -xvf hwrfv3.5a_hwrf-utilities.tar.gz  
tar -xvf hwrfv3.5a_pomtc.tar.gz  
tar -xvf hwrfv3.5a_gfdl-vortextracker.tar.gz  
tar -xvf hwrfv3.5a_ncep-coupler.tar.gz  
tar -xvf hwrfv3.5a_WRFV3.tar.gz  
tar -xvf hwrfv3.5a_WPSV3.tar.gz  
tar -xvf hwrfv3.5a_UPP.tar.gz  
tar -xvf hwrfv3.5a_GSI.tar.gz
```

A.3. COMPILACIÓN DE LOS MÓDULOS

Para configurar correctamente WRF-NMM para el sistema HWRF, se deben establecer las siguientes variables de entorno adicionales más allá de lo que normalmente requiere WRF, asumiendo que se utiliza bourne shell:

```
export HWRF=1
export WRF_NMM_CORE=1
export WRF_NMM_NEST=1
export WRFIO_NCD_LARGE_FILE_SUPPORT=1
```

Para configurar WRF-NMM nos vamos al directorio superior del módulo WRF y ejecutamos

```
./configure
```

Se presentará una lista de opciones de construcción para la computadora. Estas opciones pueden incluir múltiples compiladores y opciones de paralelismo. Para el sistema HWRF, sólo se recomienda la compilación de memoria distribuida (dmpar).

Al completar la configuración del modelo WRF, se crea el archivo `configure.wrf` con las opciones de compilación y las rutas. Estos valores se pueden editar para una personalización adicional del proceso de compilación.

Para crear el componente WRF-NMM, ejecutamos el comando:

```
./compile nmm_real
```

Para guardar la salida y error estándar en un archivo de registro para referencia, para bash utilizamos el comando

```
./compile nmm_real 2>&1 | tee build.log
```

Una compilación exitosa produce dos ejecutables enumerados a continuación en el directorio `main/`

`real_nmm.exe`: Inicialización del WRF

`wrf.exe`: Integración del modelo WRF

La compilación de las utilidades HWRF (HWRF-Utilities) requiere que las dos variables de ruta de acceso, `NETCDF` y `WRF_DIR`, se establezcan en las rutas de acceso adecuadas. La ruta

NETCDF de la librería netCDF es necesaria para construir el componente WRF-NMM, y su valor debe establecerse de forma adecuada si dicho componente se compila correctamente. La variable de ruta WRF_DIR debe apuntar al directorio WRF compilado anteriormente. Primero debe crear WRF antes de compilar cualquiera de los otros componentes.

```
export NETCDF={directorio donde se instaló netCDF}  
export WRF_DIR={directorio donde se instaló WRF}
```

Es fundamental que el compilador Fortran utilizado para construir las librerías (Intel, PGI, XLF, etc.) sea el mismo que el compilador utilizado para compilar el código fuente. Normalmente, esto es sólo un problema en dos situaciones: en sistemas Linux que tienen Múltiples compiladores instalados; Y en sistemas donde existe la posibilidad de elegir entre construir el código con direccionamiento de 32 bits o de 64 bits.

Para configurar HWRF-Utilities para la compilación, entramos al directorio hwrf-utilities y ejecutamos

```
./configure
```

El script de configuración comprueba el hardware del sistema y si las variables de ruta no están establecidas, pregunta por las rutas correctas a las bibliotecas netCDF y al directorio de compilación WRF. Concluye pidiendo al usuario que elija una configuración soportada por la arquitectura actual de la máquina.

Si tiene éxito, se generara el archivo llamado configure.hwrf en el directorio hwrf-utilities. Este archivo contiene las opciones de compilación, reglas y rutas específicas de la arquitectura de la máquina actual, y se puede editar para cambiar las opciones de compilación si es necesario.

Para guardar la salida del proceso de compilado se ejecuta

```
./compile 2>&1 | tee build.log
```

Si la compilación es exitosa, se crearán 24 ejecutables en el directorio exec/

- diffwrf_3dvar.exe
- grbindex.exe
- hwrf_anl_4x_step2.exe
- hwrf_anl_bogus_10m.exe

- hwrp_anl_cs_10m.exe
- hwrp_bin_io.exe
- hwrp_create_nest_1x_10m.exe
- hwrp_create_trak_fnl.exe
- hwrp_create_trak_guess.exe
- hwrp_data_flag.exe
- hwrp_inter_2to1.exe
- hwrp_inter_2to2.exe
- hwrp_inter_2to6.exe
- hwrp_inter_4to2.exe
- hwrp_inter_4to6.exe
- hwrp_merge_nest_4x_step12_3n.exe
- hwrp_pert_ct1.exe
- hwrp_prep.exe
- hwrp_readtdrstmid.exe
- hwrp_readtdrtime.exe
- hwrp_split1.exe
- hwrp_swcorner_dynamic.exe
- hwrp_wrfout_newtime.exe
- wgrib.exe

Además, se crearán diez librerías en el directorio `libs/`

- libbacio.a
- libblas.a
- libbufr_i4r4.a
- libbufr_i4r8.a
- libsfcio_i4r4.a
- libsigio_i4r4.a
- libsp_i4r8.a
- libsp_i4r4.a
- libw3_i4r8.a
- libw3_i4r4.a

Estas librerías serán utilizadas por el GFDL Vortex Tracker y el modelo de océano POM-TC. El paso de configuración para estos componentes requerirá configurar una variable de ruta para que apunte al directorio `hwrp-utilities/libs/` del directorio de utilidades HWRP. Las

utilidades de HWRF se pueden compilar para producir solamente las bibliotecas escribiendo el comando siguiente

```
./compile library
```

La versión para ciclones tropicales del POM-TC requiere de tres librerías: SFCIO, SP y W3. En las plataformas que carecen de las bibliotecas matemáticas ESSL, normalmente cualquier otra cosa que no sean máquinas IBM AIX, se requiere una cuarta biblioteca (BLAS). Todas estas bibliotecas se encuentran en el directorio `hwrp-utilities/libs/` y deberían estar disponibles si el componente HWRF-Utilities se ha construido correctamente. Primero debe construirlos antes de construir POM-TC. Debemos configurar las librerías de la siguiente manera, considerar que `dir_hwrp` corresponde al directorio base de nuestra instalación del HWRF

```
export LIB_W3_PATH=${dir_hwrp}/hwrp-utilities/libs/  
export LIB_W3_PATH=${dir_hwrp}/hwrp-utilities/libs/  
export LIB_SP_PATH=${dir_hwrp}/hwrp-utilities/libs/  
export LIB_SFCIO_PATH=${dir_hwrp}/hwrp-utilities/libs/  
export LIB_BLAS_PATH=${dir_hwrp}/hwrp-utilities/libs/
```

La configuración de POM-TC para su compilación, entramos al directorio `pomptc/` y ejecutamos

```
./configure
```

El script de configuración comprueba el hardware del sistema, y si las variables de ruta no están establecidas, pide rutas de software a las bibliotecas BLAS, W3, SP y SFCIO, y para Linux. Concluye pidiendo al usuario que elija una configuración soportada por la arquitectura actual de la máquina.

Después de seleccionar la opción de compilador deseada, el script de configuración crea un archivo denominado `configure.pom`. Este archivo contiene opciones de compilación, reglas y rutas específicas de la arquitectura de la máquina actual, y se puede editar para cambiar las opciones de compilación, si se desea.

Para compilar el POM-TC y guardar la salida de generación en un archivo de registro, ejecutamos

```
./compile 2>&1 | tee ocean.log
```

Si la compilación es exitosa, se generaran 13 archivos ejecutables en el directorio ocean_exec/

- gfdl_date2day.exe
- gfdl_day2date.exe
- gfdl_find_region.exe
- gfdl_getsst.exe
- gfdl_ocean_eastatl.exe
- gfdl_ocean_eastpac.exe
- gfdl_ocean_ext_eastatl.exe
- gfdl_ocean_united.exe
- gfdl_sharp_mcs_rf_l2m_rmy5.exe
- hwrp_ocean_eastatl.exe
- hwrp_ocean_eastatl_ext.exe
- hwrp_ocean_eastpac.exe
- hwrp_ocean_united.exe

Los ejecutables hwrp_ocean_united.exe, hwrp_ocean_eastpac.exe, hwrp_ocean_eastatl.exe y hwrp_ocean_eastatl_ext.exe, son los ejecutables del modelo oceánico utilizados durante la ejecución del modelo acoplado océano- atmósfera. Los ejecutables restantes se utilizan para la inicialización del océano.

El GFDL Vortex Tracker requiere dos librerías externas, W3 y BACIO. Estas librerías se encuentran en el directorio hwrp-utility/libs/ y deberían estar disponibles si HWRF-utilities se construyen correctamente, por lo que estas utilidades deber crearse antes de Vortex Tracker. Considerando que $\{\text{dir_hwrp}\}$ corresponde al directorio base de nuestra instalación del HWRF, configuramos las variables de ambiente de la siguiente manera

```
export LIB_W3_PATH $\{\text{dir\_hwrp}\}$  /hwrp-utilities/libs/  
export LIB_BACIO_PATH $\{\text{dir\_hwrp}\}$  /hwrp-utilities/libs/
```

La configuración de GFDL Vortex Tracker para su compilación, dentro del directorio gfdl-vortextracker y ejecutamos

```
./configure
```

El script de configuración comprueba el hardware del sistema, y si las rutas de las variables no están establecidas, pide las rutas de software a las bibliotecas W3 y BACIO. Concluye

pidiendo al usuario que elija una configuración soportada por la arquitectura actual de la máquina.

El script configure crea un archivo llamado configure.trk. Este archivo contiene opciones de compilación, reglas y rutas específicas para la arquitectura actual de la máquina.

El archivo de configuración se puede editar para cambiar las opciones de compilación en caso que sea necesario.

Para compilar Vortex Tracker y guardar la salida de generación en un archivo de registro, ejecutamos

```
./compile 2>&1 | tee tracker.log
```

Si la compilación es exitosa, se generaran 3 archivos ejecutables en el directorio trk_exec/

- hwrf_gettrk.exe
- hwrf_tave.exe
- hwrf_vint.exe

La configuración de NCEP Coupler para su compilación, entramos al directorio ncep-coupler y ejecutamos

```
./configure
```

El script de configuración revisa el hardware del Sistema, pide al usuario que elija una configuración soportada por la arquitectura actual de la máquina, y crea un archivo de configuración denominado configure.cpl.

El archivo de configuración configure.cpl contiene opciones de compilación, reglas y rutas específicas de la arquitectura de la máquina actual, y se puede editar para cambiar las opciones de compilación si es necesario.

Para compilar el acoplador y guardar la salida de generación en un archivo de registro, ejecutamos

```
./compile 2>&1 | tee coupler.log
```

El módulo WPS requiere el mismo entorno de compilación que el modelo WRF-NMM, incluyendo las librerías netCDF y MPI. Dado que el WPS realiza llamadas directas a las librerías de la API de E/S incluidas con el modelo WRF, por tal motivo el modelo WRF-NMM debe construirse antes de crear el WPS.

Considerando que `dir_hwrf` corresponde al directorio base de nuestra instalación del HWRF, configuramos el entorno del compilador para el WPS estableciendo la variable de ambiente de la siguiente manera

```
export WRF_DIR=${dir_hwrf}/WRFV3/
```

Dentro del directorio WPS, ejecutamos el comando siguiente

```
./configure
```

Seleccionamos la opción apropiada para memoria distribuida “dmpar” para nuestra arquitectura.

Para compilar WPS y guardar la salida de generación en un archivo de registro, ejecutamos

```
./compile 2>&1 | tee wps.log
```

El post-procesador unificado de NCEP (UPP) fue diseñado para interpolar la salida de WRF de coordenadas y variables nativas a coordenadas y variables más útiles para el análisis. Específicamente, la UPP desacelera la salida HWRF, interpola los datos desde su cuadrícula vertical nativa a niveles estándar y crea variables de diagnóstico adicionales. UPP requiere los mismos compiladores Fortran y C utilizados para construir el modelo WRF. Además, UPP requiere la librería netCDF y la API de E/S del WRF (esta última se incluye con la compilación WRF).

La compilación UPP requiere varias librerías de soporte (IP, SP, W3), que se proporcionan con el código fuente y se encuentran en el directorio UPP/lib/. Estas librerías son para la compilación UPP solamente. No deben confundirse con las librerías del mismo nombre que se encuentran en el directorio hwrf-utilities/libs/.

Dado que la construcción UPP requiere vincular a las bibliotecas WRF-NMM API de E/S, debe ser capaz de encontrar el directorio WRF. La construcción UPP utiliza la variable de entorno WRF_DIR para definir la ruta de acceso a WRF. Por lo tanto, la variable de ruta WRF_DIR debe establecerse en la ubicación del directorio raíz WRF.

Además de establecer la variable de ruta, la creación de UPP para su uso con HWRF requiere establecer la variable de entorno HWRF. Esta es la misma variable establecida para construir WRF-NMM para HWRF.

```
export HWRF=1
export WRF_DIR=${SCRATCH}/HWRF/WRFV3/
```

UPP utiliza un mecanismo de compilación similar al utilizado por el modelo WRF, ejecutamos

```
./configure
```

Para generar el archivo de configuración UPP. El script de configuración se quejará si no se ha establecido la ruta WRF_DIR. A continuación, se le dará una lista de opciones de configuración adaptadas a su computadora. El script de configuración generará el archivo de configuración configure.upp. Si es necesario, el archivo configure.upp se puede modificar para cambiar las opciones y rutas de compilación predeterminadas. Para compilar ejecutamos

```
./compile 2>&1 | tee build.log
```

El módulo GSI de la comunidad requiere el mismo entorno de compilación que el modelo WRF-NMM, incluyendo las librerías netCDF, MPI y LAPACK. Además, GSI hace llamadas directas a las librerías de API de E/S WRF incluidas con el modelo WRF. Por lo tanto, el modelo WRF debe ser construido antes de la construcción del GSI.

La construcción de GSI para su uso con HWRF requiere establecer tres variables ambientales. El primero, HWRF indica activar las opciones HWRF en la generación GSI. Este es el mismo indicador establecido cuando se construye WRF-NMM para HWRF. El segundo es una variable de ruta que apunta a la raíz del directorio de compilación de WRF. La tercera es la variable LAPACK_PATH, que indica la ubicación de la biblioteca LAPACK en el sistema.

```
export HWRF=1
export WRF_DIR=${SCRATCH}/HWRF/WRFV3
```

La variable de entorno adicional LAPACK_PATH puede ser necesaria en algunos sistemas. Normalmente, la variable de entorno LAPACK_PATH solo necesita ser configurada en sistemas Linux sin una versión de LAPACK suministrada. Los problemas con la librería LAPACK suministrada por el vendedor son más probables de ocurrir con el compilador de Intel.

Mientras que los compiladores Intel normalmente tienen las bibliotecas MKL instaladas, el compilador ifort no carga automáticamente la biblioteca. Por lo tanto, es necesario establecer la variable LAPACK_PATH a la ubicación de las bibliotecas MKL cuando se utiliza el compilador Intel.

Suponiendo que la ruta de la biblioteca MKL se establece en la variable de entorno MKL, entonces el entorno LAPACK para bash es

```
export LAPACK_PATH=$MKL
```

Para construir GSI para HWRP, entramos al directorio GSI y ejecutamos el comando

```
./configure
```

Se presentaran las diferentes opciones de compilación, debemos elegir la que corresponda con la plataforma y compiladores instalados en el sistema.

Después de seleccionar la opción adecuada, ejecute la secuencia de comandos de compilación (bash)

```
./compile 2>&1 | tee build.log
```

A.4. OPERACIÓN DEL MODELO HWRP

El primer paso es editar el archivo global_vars.ksh que se encuentra en el directorio hwrp-utilities/wrapper_scripts/global_vars.ksh y editamos las variables de la tabla A.1. Posterior a ello se debe crear un directorio para depositar los resultados, el cual debe coincidir con el definido en el archivo globa_vars.ksh.

Tabla A.1. Variables a editar del archivo global_vars.ksh

Variable	Comentario
export START_TIME=2012102806	Tiempo de inicialización YYYYMMDDHH
export START_TIME_MINUS6=2012102800	Tiempo de inicialización menos 6 horas YYYYMMDDHH
export FCST_LENGTH=12	Longitud del pronóstico en horas
export FCST_INTERVAL=6	Intervalo del pronóstico en horas
export STORM_NAME=SANDY	Nombre de la tormenta

export SID=18L	Identificador de la tormenta (L para el Atlántico, E para el Pacífico Este y W para Pacífico Oeste)
export BASIN=AL	Cuenca (AL para el Atlántico, EP para el Pacífico Este y WP para Pacífico Oeste)
export RUN_PREP_HYB=T	T para correr prep_hybrid y F para no correrlo
export UPP_PROD_SAT=T	T para almacenar variables de salida en UPP y N para no salvar.
export RUN_GSI=T	T para correr GSI y N para no correr GSI
export HWRP_SRC_DIR=\$USER/HWRP3.5b/sorc	Nivel superior del directorio HWRP
export HWRP_DATA_DIR=\$USER/HWRP/datasets	Nivel superior del directorio de datasets
export CYCLE_DATA=\${HWRP_OUTPUT_DIR}/\${SID}/\${START_TIME MINUS6}	Directorio del ciclo
export DOMAIN_DATA=\${HWRP_OUTPUT_DIR}/\${SID}/\${START_TIME}	Nivel superior del directorio de salida
export GFS_DIR=\${HWRP_DATA_DIR}/GFS	Nivel superior del directorio de los datos GFS
export GFS_SPECTRAL_DIR=\${GFS_DIR}/spectral	
export GFS_GRIDDED_DIR=\${GFS_DIR}/gridded	
export GFS_OBS_DIR=\${GFS_DIR}/obs	
export GEFS_DIR=\${HWRP_DATA_DIR}/GEFS	Nivel superior del directorio de los datos GEFS
export GEFS_ENS_FCST_DIR=\${GEFS_DIR}/spectral	
export GDAS_DIR=\${HWRP_DATA_DIR}/GDAS	Nivel superior del directorio de los datos GDAS
export GDAS_SPECTRAL_DIR=\${GDAS_DIR}/spectral	
export GDAS_GRIDDED_DIR=\${GDAS_DIR}/gridded	
export GDAS_OBS_DIR=\${GDAS_DIR}/obs	
export TCVITALS=\${HWRP_DATA_DIR}/Tcvitals	Directorio TCVitals
export OCEAN_FIXED_DIR=\${HWRP_DATA_DIR}/ocean	Archivos ocean fix
export LOOP_CURRENT_DIR=\${HWRP_DATA_DIR}/loop current	Corriente de ciclo oceanico
export CRTM_FIXED_DIR=\${HWRP_DATA_DIR}/upp	Archivo UPP fix

export GEOG_DATA_PATH=\${HWRP_DATA_DIR}/wps_geog	Archivos geog de WPS
export GSI_FIXED_DIR=\${HWRP_DATA_DIR}/gsi	Archivos GSI fix
export GSI_CRTM_FIXED_DIR=\${GSI_FIXED_DIR}/CRTM_Coefficients	Coeficiente CRTM de GSI
export TDR_OBS_DIR=\${HWRP_DATA_DIR}/TDR	Datos TDR
export MPIRUN=mpirun.lsf	Comando MPIRUN
export WRF_ANAL_CORES=12	Núcleos recomendados para cada módulo
export WRF_GHOST_CORES=12	
export HWRP_FCST_CORES=202	
export GEOGRID_CORES=12	
export METGRID_CORES=12	
export REAL_CORES=1	
export GSI_CORES=200	
export UNI_CORES=12	
export PREP_HYB_CORES=12	
export ATCFNAME=HCOM	Identificador ATCF
export GRADS_BIN=/opt/grads/2.0.2/bin	Ruta de GrADS
export GADDIR=/opt/grads/2.0.2/data	

El archivo hwrp-utilities/parm/namelist se modificara en tiempo de ejecución de acuerdo al archivo global_vars.ksh, por lo que no se debe modificar (figura A.1).

```

num_metgrid_soil_levels = @[GFS_SOURCE==GRIB_1x1_REDUCED?2:4],
/
&physics
num_soil_layers = 4,
mp_physics = 85, 85, 85,
ra_lw_physics = 98, 98, 98,
ra_sw_physics = 98, 98, 98,
sf_sfclay_physics = 88, 88, 88,
sf_surface_physics = 88, 88, 88,
bl_pbl_physics = 3, 3, 3,
cu_physics = 84, 84, 0,
mommix = 1.0, 1.0, 1.0,
var_ric = 1.0,
coef_ric_l = 0.16,
coef_ric_s = 0.25,
h_diff = 1.0, 1.0, 1.0,
gwd_opt = @[GWD==YES?2:0], 0, 0,

```

Figura A.1. archivo hwrp-utilities/parm/namelist

Ejecutamos el programa hwrp-utilities/wrapper_scripts/hwrpdomain_wrapper el cual realiza lo siguiente:

- ✓ Verifica las variables de ambiente
- ✓ Creará el directorio de trabajo ($\${HWR_OUTPUT_DIR}\backslash\${SID}\backslash\${START_TIME}\backslash\text{messages}$)
- ✓ Creará el archivo `tcvital` y `tcvitals.as` a partir del archivo `Tcvitals` para la tormenta en particular
- ✓ Determina el centro de la tormenta
- ✓ Determina el centro del dominio
- ✓ Crea el archivo de estatus GSI (si `RUN_GIS` es verdadero genera el archivo `go_gsi`, si no, generara el archivo `no_gsi`)

Para la ejecución del módulo WPS, corremos el programa `hwr-utilities/wrapper_scripts/geogrid_wrapper` el cual realizara lo siguiente:

- ✓ Verifica las variables de ambiente
- ✓ Verifica los ejecutables
- ✓ Crea el directorio de trabajo ($\${HWR_OUTPUT_DIR}\backslash\${SID}\backslash\${START_TIME}\backslash\text{geoprd}$)
- ✓ Crea el archivo `namelist`
- ✓ Copia el archivo de tablas `geogrid`
- ✓ Ejecuta `geogrid.exe` generando los archivos de los tres dominios `geo_nmm.d01.nc`, `geo_nmm_nest.l01.nc` y `geo_nmm_nest.l02.nc`

Ejecutamos `hwr-utilities/wrapper_scripts/ungrid_wrapper` el cual realizara lo siguiente:

- ✓ Verifica las variables de ambiente
- ✓ Verifica los ejecutables
- ✓ Crea el directorio de trabajo ($\${HWR_OUTPUT_DIR}\backslash\${SID}\backslash\${START_TIME}\backslash\text{ungrbprd}\backslash\${YYYYMMDDHH}$)
- ✓ Crea el archivo `namelist`
- ✓ Copia el archivo de tablas `ungrid`
- ✓ Liga los archivos de datos `grib`
- ✓ Ejecuta `ungrid.exe` generando los archivos `FILE:YYYY-MM-DD_HH`

Ejecutamos `hwr-utilities/wrapper_scripts/metgrid_wrapper` el cual realizara lo siguiente:

- ✓ Verifica las variables de ambiente
- ✓ Verifica los ejecutables

- ✓ Crea el directorio de trabajo
($\${HWR_OUTPUT_DIR}\${SID}\${START_TIME}\metgridprd\${YYYYMMDDHH}$)
- ✓ Crea el archivo namelist
- ✓ Copia el archivo de tablas metgrid
- ✓ Copia los archivos de salida de geogrid
- ✓ Ejecuta metgrid.exe generando los archivos met_nmm.d01.YYYY-MM-DD_HH.nc

Para la ejecución del módulo Prep_hybrid corremos el programa hwr-utilities/wrapper_scripts/prep_hybrid_wrapper el cual realizará lo siguiente:

- ✓ Verifica las variables de ambiente
- ✓ Verifica los ejecutables
- ✓ Crea el directorio de trabajo
($\${HWR_OUTPUT_DIR}\${SID}\${START_TIME}\prep_hybrid\${YYYYMMDDHH}$)
- ✓ Copia los archivos de entrada
- ✓ Ejecuta prep_hybrid generando los archivos hwrfininit_00, hwrfbcs00_00 y hwrfbcs00_01

Para la ejecución del módulo Real se corre el programa hwr-utilities/wrapper_scripts/real_wrapper el cual realizará lo siguiente:

- ✓ Verifica las variables de ambiente
- ✓ Verifica los ejecutables
- ✓ Crea el directorio de trabajo ($\${HWR_OUTPUT_DIR}\${SID}\${START_TIME}\realprd\${YYYYMMDDHH}$)
- ✓ Liga los archivos de entrada
- ✓ Corre hwr_swcorner_dynamic.exe para calcular la localización del dominio
- ✓ Genera el archivo namelist
- ✓ Se asegura que la pila este configurada mayor a 2 GB
- ✓ Ejecuta real.exe generando los archivos wrfinput_d01, wrfbdy_d01 y fort.65

Para la ejecución del módulo GSI se corre el programa hwr-utilities/wrapper_scripts/gsi_wrfinput_wrapper el cual realizará lo siguiente:

- ✓ Verifica las variables de ambiente
- ✓ Verifica los ejecutables

- ✓ Crea el directorio de trabajo ($\${HWR_OUTPUT_DIR}/\${SID}/\${START_TIME}/$ gsiprd/ $\${YYYYMMDDHH}$)
- ✓ Copia los archivos fix y el analisis del campo de fondo
- ✓ Señala los datos observados cercanos al centro de la tormenta si estos no han sido asimilados
- ✓ Genera el archivo namelist
- ✓ Copia los datos ensamblados
- ✓ Ejecuta gsi.exe y genera los archivos stdout y wrf_inout

La ejecución del módulo WRF Analisis se lleva a cabo mediante el programa hwrutilities/wrapper_scripts/wrfanalysis_wrapper el cual realizará lo siguiente:

- ✓ Verifica las variables de ambiente
- ✓ Verifica los ejecutables
- ✓ Crea el directorio de trabajo ($\${HWR_OUTPUT_DIR}/\${SID}/\${START_TIME}/$ wrfanalysisprd/ $\${YYYYMMDDHH}$)
- ✓ Liga los archivos fix, las condiciones iniciales y de frontera.
- ✓ Ejecuta hwr_swcorner_dinamic.exe para calcular los valores istart y jstart para el dominio central.
- ✓ Genera el archivo namelist
- ✓ Ejecuta wrfanalysis y genera los siguientes archivos wrfanl_d02, wrfanl_d03

La ejecución del módulo WRF Ghost se lleva a cabo mediante el programa hwrutilities/wrapper_scripts/wrfghost_wrapper el cual realizará lo siguiente:

- ✓ Verifica las variables de ambiente
- ✓ Verifica los ejecutables
- ✓ Crea el directorio de trabajo ($\${HWR_OUTPUT_DIR}/\${SID}/\${START_TIME}/$ wrfghostprd/ $\${YYYYMMDDHH}$)
- ✓ Liga los archivos fix, las condiciones iniciales y de frontera.
- ✓ Ejecuta hwr_swcorner_dinamic.exe para calcular los valores istart y jstart para el dominio central.
- ✓ Genera el archivo namelist
- ✓ Ejecuta wrfghost y genera los siguientes archivos wrfanl_d02, wrfanl_d03

La ejecución del módulo Tracker Analisis se lleva a cabo mediante el programa hwrutilities/wrapper_scripts/track_analysis_wrapper el cual realizará lo siguiente:

- ✓ Verifica las variables de ambiente
- ✓ Verifica los ejecutables
- ✓ Crea el directorio de trabajo ($\${HWR_OUTPUT_DIR}/\${SID}/\${START_TIME}/$
 $trkanalysisprd/\${YYYYMMDDHH}$)
- ✓ Copia los archivos fix UPP y wrfout_d01 desde la corrida del analisis 90-s
- ✓ Ejecuta hwrf_wrfout_newtime.exe para cambiar la marca del tiempo en 90-s
wrfout_d01 para el tiempo 0 para GFDL tracker para inicizar desde la hora 0.
- ✓ Ejecuta UPP para wrfout_d01
- ✓ Ejecuta copygb para desescalonar la salida a latitud y longitud normales
- ✓ Ejecuta GFDL vortex tracker generando el archivo gfs-anl-fix.atcfunix

APENDICE B. CONTENIDO DEL DVD

Se acompaña el presente informe con un DVD cuyo contenido se lista a continuación:

- InformeFinal.pdf
- HWRFV3.5b
 - ✓ hwrp_v3.5b_hwrp-utilities.tar.gz
 - ✓ hwrp_v3.5b_pomtc.tar.gz
 - ✓ hwrp_v3.5b_ncep-coupler.tar.gz
 - ✓ hwrp_v3.5b_gfdl-vortextracker.tar.gz
 - ✓ hwrp_v3.5b_GSI.tar.gz
 - ✓ hwrp_v3.5b_WRFV3.tar.gz
 - ✓ hwrp_v3.5b_WPSV3.tar.gz
 - ✓ hwrp_v3.5b_UPP.tar.gz