****
13278 Thu Jul 28 06:11:58 2011
new/src/cpu/zero/vm/frame_zero.cpp
****

```
   1 /*
   2  * Copyright (c) 2003, 2010, Oracle and/or its affiliates. All rights reserved.
   3  * Copyright 2007, 2008, 2009, 2010, 2011 Red Hat, Inc.
   3  * Copyright 2007, 2008, 2009, 2010 Red Hat, Inc.
   4  * DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS FILE HEADER.
   5  *
   6  * This code is free software; you can redistribute it and/or modify it
   7  * under the terms of the GNU General Public License version 2 only, as
   8  * published by the Free Software Foundation.
   9  *
  10  * This code is distributed in the hope that it will be useful, but WITHOUT
  11  * ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
  12  * FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
  13  * version 2 for more details (a copy is included in the LICENSE file that
  14  * accompanied this code).
  15  *
  16  * You should have received a copy of the GNU General Public License version
  17  * 2 along with this work; if not, write to the Free Software Foundation,
  18  * Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA.
  19  *
  20  * Please contact Oracle, 500 Oracle Parkway, Redwood Shores, CA 94065 USA
  21  * or visit www.oracle.com if you need additional information or have any
  22  * questions.
  23  *
  24  */

  26 #include "precompiled.hpp"
  27 #include "code/scopeDesc.hpp"
  28 #include "interpreter/interpreter.hpp"
  29 #include "interpreter/interpreterRuntime.hpp"
  30 #include "memory/resourceArea.hpp"
  31 #include "oops/markOop.hpp"
  32 #include "oops/methodOop.hpp"
  33 #include "oops/oop.inline.hpp"
  34 #include "runtime/frame.inline.hpp"
  35 #include "runtime/handles.inline.hpp"
  36 #include "runtime/javaCalls.hpp"
  37 #include "runtime/monitorChunk.hpp"
  38 #include "runtime/signature.hpp"
  39 #include "runtime/stubCodeGenerator.hpp"
  40 #include "runtime/stubRoutines.hpp"
  41 #include "vmreg_zero.inline.hpp"
  42 #ifdef COMPILER1
  43 #include "c1/c1_Runtime1.hpp"
  44 #include "runtime/vframeArray.hpp"
  45 #endif

  47 #ifdef ASSERT
  48 void RegisterMap::check_location_valid() {
  49   ShouldNotCallThis();
  50 }
_____unchanged_portion_omitted_

 421 #ifdef ASSERT

 423 void frame::describe_pd(FrameValues& values, int frame_no) {

 425 }

 427 #endif
 428 #endif /* ! codereview */
```

```
*******************************************************
    3130 Thu Jul 28 06:11:59 2011
new/src/cpu/zero/vm/methodHandles_zero.hpp
*******************************************************
```

```
/*
 * Copyright (c) 2011, Oracle and/or its affiliates. All rights reserved.
 * Copyright 2011 Red Hat, Inc.
 * DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS FILE HEADER.
 *
 * This code is free software; you can redistribute it and/or modify it
 * under the terms of the GNU General Public License version 2 only, as
 * published by the Free Software Foundation.
 *
 * This code is distributed in the hope that it will be useful, but WITHOUT
 * ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
 * FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
 * version 2 for more details (a copy is included in the LICENSE file that
 * accompanied this code).
 *
 * You should have received a copy of the GNU General Public License version
 * 2 along with this work; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA.
 *
 * Please contact Oracle, 500 Oracle Parkway, Redwood Shores, CA 94065 USA
 * or visit www.oracle.com if you need additional information or have any
 * questions.
 *
 */


// Adapters
enum /* platform_dependent_constants */ {
  adapter_code_size = 0
};

#define TARGET_ARCH_NYI_6939861 1
// ..#ifdef TARGET_ARCH_NYI_6939861
// ..  // Here are some backward compatible declarations until the 6939861 ports
// ..  #define _adapter_flyby    (_EK_LIMIT + 10)
// ..  #define _adapter_ricochet (_EK_LIMIT + 11)
// ..  #define _adapter_opt_spread_1    _adapter_opt_spread_1_ref
// ..  #define _adapter_opt_spread_more _adapter_opt_spread_ref
// ..  enum {
// ..    _INSERT_NO_MASK   = -1,
// ..    _INSERT_REF_MASK  = 0,
// ..    _INSERT_INT_MASK  = 1,
// ..    _INSERT_LONG_MASK = 3
// ..  };
// ..  static void get_ek_bound_mh_info(EntryKind ek, BasicType& arg_type, int&
// ..    arg_type = ek_bound_mh_arg_type(ek);
// ..    arg_mask = 0;
// ..    arg_slots = type2size[arg_type];;
// ..  }
// ..  static void get_ek_adapter_opt_swap_rot_info(EntryKind ek, int& swap_byte
// ..    int swap_slots = ek_adapter_opt_swap_slots(ek);
// ..    rotate = ek_adapter_opt_swap_mode(ek);
// ..    swap_bytes = swap_slots * Interpreter::stackElementSize;
// ..  }
// ..  static int get_ek_adapter_opt_spread_info(EntryKind ek) {
// ..    return ek_adapter_opt_spread_count(ek);
// ..  }
// ..
// ..  static void insert_arg_slots(MacroAssembler* _masm,
// ..                               RegisterOrConstant arg_slots,
// ..                               int arg_mask,
// ..                               Register argslot_reg,
```



```
// ..                               Register temp_reg, Register temp2_reg, Regis
// ..
// ..  static void remove_arg_slots(MacroAssembler* _masm,
// ..                               RegisterOrConstant arg_slots,
// ..                               Register argslot_reg,
// ..                               Register temp_reg, Register temp2_reg, Regis
// ..
// ..  static void trace_method_handle(MacroAssembler* _masm, const char* adapte
// ..#endif //TARGET_ARCH_NYI_6939861

#endif /* ! codereview */
```

```
*************************************************************
    4756 Thu Jul 28 06:12:00 2011
new/src/cpu/zero/vm/sharedRuntime_zero.cpp
*************************************************************
```

```
   1 /*
   2  * Copyright (c) 2003, 2010, Oracle and/or its affiliates. All rights reserved.
   3  * Copyright 2007, 2008, 2009, 2010, 2011 Red Hat, Inc.
   4  * DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS FILE HEADER.
   5  *
   6  * This code is free software; you can redistribute it and/or modify it
   7  * under the terms of the GNU General Public License version 2 only, as
   8  * published by the Free Software Foundation.
   9  *
  10  * This code is distributed in the hope that it will be useful, but WITHOUT
  11  * ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
  12  * FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
  13  * version 2 for more details (a copy is included in the LICENSE file that
  14  * accompanied this code).
  15  *
  16  * You should have received a copy of the GNU General Public License version
  17  * 2 along with this work; if not, write to the Free Software Foundation,
  18  * Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA.
  19  *
  20  * Please contact Oracle, 500 Oracle Parkway, Redwood Shores, CA 94065 USA
  21  * or visit www.oracle.com if you need additional information or have any
  22  * questions.
  23  *
  24  */

  26 #include "precompiled.hpp"
  27 #include "asm/assembler.hpp"
  28 #include "assembler_zero.inline.hpp"
  29 #include "code/debugInfoRec.hpp"
  30 #include "code/icBuffer.hpp"
  31 #include "code/vtableStubs.hpp"
  32 #include "interpreter/interpreter.hpp"
  33 #include "oops/compiledICHolderOop.hpp"
  34 #include "prims/jvmtiRedefineClassesTrace.hpp"
  35 #include "runtime/sharedRuntime.hpp"
  36 #include "runtime/vframeArray.hpp"
  37 #include "vmreg_zero.inline.hpp"
  38 #ifdef COMPILER1
  39 #include "c1/c1_Runtime1.hpp"
  40 #endif
  41 #ifdef COMPILER2
  42 #include "opto/runtime.hpp"
  43 #endif
  44 #ifdef SHARK
  45 #include "compiler/compileBroker.hpp"
  46 #include "shark/sharkCompiler.hpp"
  47 #endif

  49 DeoptimizationBlob *SharedRuntime::_deopt_blob;
  50 SafepointBlob      *SharedRuntime::_polling_page_safepoint_handler_blob;
  51 SafepointBlob      *SharedRuntime::_polling_page_return_handler_blob;
  52 RuntimeStub        *SharedRuntime::_wrong_method_blob;
  53 RuntimeStub        *SharedRuntime::_ic_miss_blob;
  54 RuntimeStub        *SharedRuntime::_resolve_opt_virtual_call_blob;
  55 RuntimeStub        *SharedRuntime::_resolve_virtual_call_blob;
  56 RuntimeStub        *SharedRuntime::_resolve_static_call_blob;

  50 int SharedRuntime::java_calling_convention(const BasicType *sig_bt,
  51                                            VMRegPair *regs,
  52                                            int total_args_passed,
  53                                            int is_outgoing) {
  54   return 0;
```



```
  55 }
______unchanged_portion_omitted_

 109 static DeoptimizationBlob* generate_empty_deopt_blob() {
 110   CodeBuffer buffer("handler_blob", 0, 0);
 111   return DeoptimizationBlob::create(&buffer, NULL, 0, 0, 0, 0);
 112 }


 115 void SharedRuntime::generate_deopt_blob() {
 116   _deopt_blob = generate_empty_deopt_blob();
 117 }

 119 SafepointBlob* SharedRuntime::generate_handler_blob(address call_ptr, bool cause
 120   return generate_empty_safepoint_blob();
 121 }

 123 RuntimeStub* SharedRuntime::generate_resolve_blob(address destination, const cha
 124   return generate_empty_runtime_stub("resolve_blob");
 117 void SharedRuntime::generate_stubs() {
 118   _wrong_method_blob =
 119     generate_empty_runtime_stub("wrong_method_stub");
 120   _ic_miss_blob =
 121     generate_empty_runtime_stub("ic_miss_stub");
 122   _resolve_opt_virtual_call_blob =
 123     generate_empty_runtime_stub("resolve_opt_virtual_call");
 124   _resolve_virtual_call_blob =
 125     generate_empty_runtime_stub("resolve_virtual_call");
 126   _resolve_static_call_blob =
 127     generate_empty_runtime_stub("resolve_static_call");

 129   _polling_page_safepoint_handler_blob =
 130     generate_empty_safepoint_blob();
 131   _polling_page_return_handler_blob =
 132     generate_empty_safepoint_blob();
 125 }
______unchanged_portion_omitted_
```

```
**********************************************************
    5743 Thu Jul 28 06:12:01 2011
new/src/share/vm/shark/sharkContext.hpp
**********************************************************
```

```
   1 /*
   2  * Copyright (c) 1999, 2010, Oracle and/or its affiliates. All rights reserved.
   3  * Copyright 2009, 2010 Red Hat, Inc.
   4  * DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS FILE HEADER.
   5  *
   6  * This code is free software; you can redistribute it and/or modify it
   7  * under the terms of the GNU General Public License version 2 only, as
   8  * published by the Free Software Foundation.
   9  *
  10  * This code is distributed in the hope that it will be useful, but WITHOUT
  11  * ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
  12  * FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
  13  * version 2 for more details (a copy is included in the LICENSE file that
  14  * accompanied this code).
  15  *
  16  * You should have received a copy of the GNU General Public License version
  17  * 2 along with this work; if not, write to the Free Software Foundation,
  18  * Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA.
  19  *
  20  * Please contact Oracle, 500 Oracle Parkway, Redwood Shores, CA 94065 USA
  21  * or visit www.oracle.com if you need additional information or have any
  22  * questions.
  23  *
  24  */

  26 #ifndef SHARE_VM_SHARK_SHARKCONTEXT_HPP
  27 #define SHARE_VM_SHARK_SHARKCONTEXT_HPP

  29 #include "shark/llvmHeaders.hpp"
  30 #include "shark/sharkCompiler.hpp"

  32 // The LLVMContext class allows multiple instances of LLVM to operate
  33 // independently of each other in a multithreaded context.  We extend
  34 // this here to store things in Shark that are LLVMContext-specific.

  36 class SharkFreeQueueItem;

  38 class SharkContext : public llvm::LLVMContext {
  39  public:
  40   SharkContext(const char* name);

  42  private:
  43   llvm::Module* _module;

  45 #if SHARK_LLVM_VERSION >= 27
  46  public:
  47 #else
  48  private:
  49 #endif
  50   llvm::Module* module() const {
  51     return _module;
  52   }

  54   // Get this thread's SharkContext
  55  public:
  56   static SharkContext& current() {
  57     return *SharkCompiler::compiler()->context();
  58   }

  60   // Module accessors
  61  public:
  62 #if SHARK_LLVM_VERSION < 27
```



```
  63   llvm::ModuleProvider* module_provider() const {
  64     return new llvm::ExistingModuleProvider(module());
  65   }
  66 #endif
  67   void add_function(llvm::Function* function) const {
  68     module()->getFunctionList().push_back(function);
  69   }
  70   llvm::Constant* get_external(const char*              name,
  71                                const llvm::FunctionType* sig) {
  72     return module()->getOrInsertFunction(name, sig);
  73   }

  75   // Basic types
  76  private:
  77   const llvm::Type*        _void_type;
  78   const llvm::IntegerType* _bit_type;
  79   const llvm::IntegerType* _jbyte_type;
  80   const llvm::IntegerType* _jshort_type;
  81   const llvm::IntegerType* _jint_type;
  82   const llvm::IntegerType* _jlong_type;
  83   const llvm::Type*        _jfloat_type;
  84   const llvm::Type*        _jdouble_type;

  86  public:
  87   const llvm::Type* void_type() const {
  88     return _void_type;
  89   }
  90   const llvm::IntegerType* bit_type() const {
  91     return _bit_type;
  92   }
  93   const llvm::IntegerType* jbyte_type() const {
  94     return _jbyte_type;
  95   }
  96   const llvm::IntegerType* jshort_type() const {
  97     return _jshort_type;
  98   }
  99   const llvm::IntegerType* jint_type() const {
 100     return _jint_type;
 101   }
 102   const llvm::IntegerType* jlong_type() const {
 103     return _jlong_type;
 104   }
 105   const llvm::Type* jfloat_type() const {
 106     return _jfloat_type;
 107   }
 108   const llvm::Type* jdouble_type() const {
 109     return _jdouble_type;
 110   }
 111   const llvm::IntegerType* intptr_type() const {
 112     return LP64_ONLY(jlong_type()) NOT_LP64(jint_type());
 113   }

 115   // Compound types
 116  private:
 117   const llvm::PointerType*  _itableOffsetEntry_type;
 118   const llvm::PointerType*  _jniEnv_type;
 119   const llvm::PointerType*  _jniHandleBlock_type;
 120   const llvm::PointerType*  _klass_type;
 121   const llvm::PointerType*  _methodOop_type;
 122   const llvm::ArrayType*    _monitor_type;
 123   const llvm::PointerType*  _oop_type;
 124   const llvm::PointerType*  _thread_type;
 125   const llvm::PointerType*  _zeroStack_type;
 126   const llvm::FunctionType* _entry_point_type;
 127   const llvm::FunctionType* _osr_entry_point_type;
```

```
 129  public:
 130   const llvm::PointerType* itableOffsetEntry_type() const {
 131     return _itableOffsetEntry_type;
 132   }
 133   const llvm::PointerType* jniEnv_type() const {
 134     return _jniEnv_type;
 135   }
 136   const llvm::PointerType* jniHandleBlock_type() const {
 137     return _jniHandleBlock_type;
 138   }
 139   const llvm::PointerType* klass_type() const {
 140     return _klass_type;
 141   }
 142   const llvm::PointerType* methodOop_type() const {
 143     return _methodOop_type;
 144   }
 145   const llvm::ArrayType* monitor_type() const {
 146     return _monitor_type;
 147   }
 148   const llvm::PointerType* oop_type() const {
 149     return _oop_type;
 150   }
 151   const llvm::PointerType* thread_type() const {
 152     return _thread_type;
 153   }
 154   const llvm::PointerType* zeroStack_type() const {
 155     return _zeroStack_type;
 156   }
 157   const llvm::FunctionType* entry_point_type() const {
 158     return _entry_point_type;
 159   }
 160   const llvm::FunctionType* osr_entry_point_type() const {
 161     return _osr_entry_point_type;
 162   }

 164   // Mappings
 165  private:
 166   const llvm::Type* _to_stackType[T_CONFLICT];
 167   const llvm::Type* _to_arrayType[T_CONFLICT];

 169  private:
 170   const llvm::Type* map_type(const llvm::Type* const* table,
 171                              BasicType                type) const {
 172     assert(type >= 0 && type < T_CONFLICT, "unhandled type");
 173     const llvm::Type* result = table[type];
 174     assert(result != NULL, "unhandled type");
 174     assert(type != NULL, "unhandled type");
 175     return result;
 176   }

 178  public:
 179   const llvm::Type* to_stackType(BasicType type) const {
 180     return map_type(_to_stackType, type);
 181   }
 182   const llvm::Type* to_arrayType(BasicType type) const {
 183     return map_type(_to_arrayType, type);
 184   }

 186   // Functions queued for freeing
 187  private:
 188   SharkFreeQueueItem* _free_queue;

 190  public:
 191   void push_to_free_queue(llvm::Function* function);
 192   llvm::Function* pop_from_free_queue();
 193 };
```

*unchanged_portion_omitted_*